JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

IMPROVING THE PACKET TRANSMISSION PERFORMANCE IN MANET BY DETECTING AND PREVENTING JELLY FISH ATTACK

BY **GELETU HUSSEIN WAKO**

Advisor: Mr. Esmael Kedir (PhD Candidate)

Co-Advisor: Mr. Negasa Berahanu (MSc.)

THESIS SUBMITTED TO FACULTY OF COMPUTING AND INFORATICS OF JIMMA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER NETWORKING

AUG, 2021

JIMMA, ETHIOPIA

Abstraction

The MANET would soon replace existing wireless technology, due to its easy to deploy and bare minimum infrastructure requirement and dynamic topology. Emergence of faster high-speed hand-held mobile device, mail delivery drones, drone camera footage of live cricket match etc. have reserved MANET in the reach of common man. There for it is imperative to address ubiquitous issue existing in MANET. These are safeguard from attacks by intruder nodes, consideration of tradeoffs between speed and efficient communication and selection of reliable and secure network paradigm. Analysis of the mobile ad hoc networks system from security stand point is crucial in order to construct a robust and counteractive system.

MANETs are surrounded by various attacks, each with different behavior and aftermaths. One of the serious attacks that affect the normal work of MANET is DoS attack. One of DoS is jellyfish attack, which is quite hard because of its foraging behavior and exploit the behavior of closed loop k2protocol and disturb the communication process without disobeying any protocol rules, thus the detection process becomes challenging. Consequently, traffic is disrupted leading to degradation in network throughput which degrades over all network performance. The jellyfish attack is regarded as one of the most difficult attack to detect and degrade the overall network performance. In order to mitigate jellyfish attack in MANET this paper propose a novel technique called holding period foundation and accurately detecting and preventing jellyfish attack node in the path. Support vector machine is utilized for learning packet forwarding behavior. The proposed technique chooses the node in the network for performing routing of packet on the bases of hierarchical trust evaluation property of node. The technique is tested using NS2 simulator algorithms using various parameters; throughput, packet deliver ratio, end to end delay. The result proves that finding holding period for detecting and preventing jellyfish attack node is highly efficient in jellyfish attack detection and also perform well as compared to another algorithm.

Keyword: jellyfish attack, holding period, denial of service, MANET, AODV, delay, network simulation.

JIMMA UNIVERSITY

Jimma Institute of Technology

Faculty of Computing and Informatics

APPROVAL SHEET

This independent research entitled as improving the packet transmission performance in MANET by detecting and preventing jelly fish attack have been read and approved as meeting preliminary research requirement of the Faculty of Computing in partial fulfillment for the award of the degree of masters in computer networking.

Signed by the examining committee

Name	**Signature**	**Date**
Advisor: Mr. ESMAEL KEDIR (PhD Candidate)		1/8/2021
Co. Advisor: Mr. NEGASA BIRAHANU (MSc)		8/8/2021
External Examiner: Mr. HENOCK MULUGETA (PhD)		August 29, 2021
Internal Examiner: Mr. BERHANU MEGARSA (MSc.)		

Acknowledgement

My first and foremost heartfelt gratitude goes almighty Allah for keeping me in sustaining any hindrances and succeeded my ambition. Secondly, I want to thanks my adviser Mr.Esmael Kedir and my co-advisor Mr. Negasa Berahanu. In addition, I want to thanks computer networking department chairman Mr. Kebabew Abebu and my best friend Mohammed Ahsanu for him great and strong support.

Dedication

This Research is lovingly dedicated to our respective parents, Hussein Wako, Shamsi Haji
and to my lovely brother and Sister Abdi, Siraj, Ahmad Zakir, Caltu, Badru and
Radia, next to almighty Allah. In addition, I dedicate this thesis for
my best and most knowledgeable wife. They Have given us
the drive and discipline to tackle task with enthusiast
and determination. Without Their love and
support this study Would not have
been Made possible.

Thankyou.

Ok!

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AODV	Ad Hock on Demand Distance Vector
MANET	Mobile Ad hock Network
QoS	Quality of Service
PDR	Packet Delivery Ratio
DPR	Dropped Packet Ratio
NS2	Network Simulator two
JF	Jellyfish
ACK	Acknowledgement
VANET	Vehicular Ad Hock Network
DSR	Dynamic Source Routing
TORA	Temporally Ordered Routing Algorithm
IoT	Internet of Things
RREQ	Routing Request
RREP	Routing Replay

Terminology

Routing……………………….. The process of selecting the best path and forward the packet

Link……………………………………… Connection between two or more node in the network

Path……………………………………………Route between two nodes is a collection of links

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the Study

In the previous few years, significant advancement was observed towards availability of wireless network in a number of handheld devices like portable computers, smart phones, IoT based wearable technology. The most common example of wireless communication is the availability of Wi- Fi access points in bus stops, railway stations, hotels, cafes even small shops in which people use these points to surf the internet. Wireless device connects to gateways to access the internet via infrastructure based wireless network without any sort of relaying called ad hoc network. MANET is regarded as a systematic organization of communication device willing to communicate with each other for sharing information without any fixed infrastructure [1][2].

MANET nodes are highly responsible for dynamic discovery of neighbor node to form dynamic network for transferring packets from source to destination nodes. In MANET all the mobile nodes operate in self-organization and self-managing manner connected in wireless manner and making random topology in dynamic manner[3]. All nodes in such system have freedom to move randomly in the network; organize with other nodes in an arbitrary manner and in turn the topology of MANET network changes in random fashion and highly unpredictable manner .ad hoc nodes in MANET should have the ability to detect other node presences to allow break-free communication and sharing of information[4]. The aim of ad hoc network is that its discoveries the path which start at the source and route to sink node, traversing the node that are connecting them. Packet delivery ratio, throughput, end to end delay, packet rate and routing overhead are used as attribute which perform the analysis of the network and they provide the information about which routing protocol gives better results[5]. The connectivity among the nodes in the network is done by implementing protocol. which have different types of routing discovery process from the starting point to sink node. Hence, when we talk about such networks, attacks such as jelly fish, black hole, worm hole, Sybil attack and flooding attacks makes the mobile node vulnerable to them. The types of Jellyfish attack is types of attack which maintains compliance with both the control and data protocol to make its detection and prevention difficult[6]. The jellyfish attack delays the packet before forwarding it, since we use TCP protocol, so the TCP sends each packet again as it does not receive the ACK in time [7]. This gives the congestion in network and affects the throughput of the networks. Due to no functional distinction among mobile node in MANET, an intermediate node can introduce a critical vulnerability for TCP congestion control mechanism[8]. Such a compromised /malicious node alters its forwarding behavior. In this jelly fish attack node introduces the extra delay to the coming packet before sending the packet to the next node in the path, to detect and prevent such attack node is so complex, in this study we propose the novel method to detect the jelly fish attack node Improving the packet transmission performance in MANET by Detecting and preventing the Jelly Fish attack node in AODV.

Mobile ad hoc networks are vulnerable to different types of attack due to inherently in-secure wireless communication medium and multi hop routing communication process[9].in this study we analyze behavior and impact of jellyfish attack over TCP based mobile ad hoc networks. We have implemented and evaluated delay variant Jelly fish attack through simulation process[8]. These types of attackers exploit the behavior of closed loop protocols such as TCP and disturb the communication process without disobey any protocol rules, thus the detection process is very difficult. Consequently traffic is disrupted leading to degradation in network throughput[10]. Through extensive simulation results that are obtained using an industry standard scalable network simulator called NS2, impact of these assault in terms of network throughput, overheard incurred and end to end delay is analyzed and used for devising detection and countermeasure[8]. We have k2proposed a delay calculation-based detection and prevention algorithm which detect and remove jelly fish attack node from active communication route. In our proposed algorithm, Frist end to end delay value is calculated and checked if the extra delay is added or introduced. If there is extra delay added to normal delay is detected we called these is holding period (newly introduced delay of packet) and directly calculating each delay on each node and each path that are included in the current active communication route to identify which node is detected as jellyfish attack. This is mostly by calculating transmission delay and propagation delay. Where transmission delay is the time taken to put the coming packet to the linked path and it's calculated as the ratio of packet size to the bandwidth of the link whereas propagation delay is the time taken to pass the packet from node to node and it's calculated as the ratio of distance between two nodes to the speed of the moving packet after put on the link. The aim of this paper is to description the jellyfish attack behavior and analyzes the existing solution that proposed to detect and eliminate jellyfish attack and proposing the new detecting and preventing algorithm. This is mainly about finding the holding period while transmission of packet from source node to destination node.

In MANET there is no central control of the network to control or manage. There for any malicious node can easily enter in to the network and disturb the functionality of the network. In other hand AODV is a routing protocol that create a route from source to destination on demand, but the source node does not have direct connection to destination node, because there is a number of nodes in between source node and destination node, so that the path passes all the node which is in between the source and destination node. Trusting all intermediate nodes is the behavior of MANET[11], but from the intermediate node probably there is a node that has an attacking behavior, such node attacks the packet and makes the performance of the network lower[5]. Due to this reason, we propose the network delay calculation value-based algorithm to detect and prevent jellyfish attacker node.

1.2 Statement of the Problem

MANET is an infrastructure-less network technology that have shared medium radio communication, dynamic topology and limited energy and computing resources. The application of MANET includes military service, rescue during natural disaster or tragedies and home networking. This type of networks technology is vul

nerable to different types of attack due to inherently in-secure wireless communication medium and multi hop routing communication process[11]. A mobile ad hoc network is surrounded by ton of different attacks, each with different behavior and aftermaths. One of the serious attacks that affect the normal work of MANET is DoS attack. A most of DoS attack is jellyfish attack, which is quite hard because of its foraging behavior. Jellyfish attack is regarded as one of the most challenging and worst types of attack because it is difficult to detect and hence difficult to remove from the network. This attack type desynchronizes the TCP connection by delaying the packet and hence reduces and degrades the overall network performance. The jellyfish attack can be easily carried out in MANET, because the MANET are infrastructure less and there is no centralized administration on nodes present in the networks. A malicious node can center the networks easily and create the attack[12]. This attack affects the functionality of TCP. Because of that the TCP is reliable and wait for the acknowledgements before sending more packets[9].

The basic operation of MANET lacks efficient security features in which all intermediate nodes from source to destinations are assumed as trustworthy at different layer for packet transmission. The most critical issue faced by MANET is trust all intermediary node when operating in dynamic topology[13]. It is highly easy for an attacker to eavesdrop the network, especially in wireless communication scenario and perform packet capturing and even break in the network and compromise trustworthy nodes. Without strict and serious security methodologies, all the layer especially network layer and transport layers are prone to serious threats which affect the overall MANET operational scenarios. UDP is used by most of the application in MANET as the transport layer protocol which is the prime reason of error and unreliable communication process because of interference and dynamic changing topology[14]. Various applications like FTP; HTTP requires end to end reliable communication and mostly relies on TCP protocol to reliable end to end packet delivery. In MANET, TCP does not perform well and performance decreases gradually when network mobility increases[15]. The reason is that TCP has no detection mechanism to detect whether any packet dropped or late during transmission between source and destination it may due to network property or congestion.

1.3 Objective

The overall objective of this study is to improve the performance of packet transmission in mobile ad hoc networks by detecting and preventing the jellyfish attack node.

1.4 Specific Objective

In order to achieve the general objective stated above, the following specific task will be taken into account during this study:

- To identify the jellyfish attack node in the path during packet transmission process in between source node and destination node
- To improve technique and protocol to detect node that is considered as jellyfish attack node during packet transmission from source to destination
- To investigate the characteristics of jellyfish attack node on selected area
- To improve detection protocol that detect jellyfish attack node

- To improve detection protocol prototype for the purpose of examining the effectiveness of designed scheme on jellyfish attack node detection
- Evaluate and report the experimental result founds
- And elicit the conclusion from squared experiment result and recommended further research direction works in the area.

1.5 Methodology

The Emphasized of this proposed approach is designing and developing security solution to detect and mitigate the jellyfish attack node in MANET. To accomplish the aforementioned objective of this study, several approaches considered as methodologies.

1.5.1 Literature Review

To overcome the objective of this study, several materials used like articles, journal papers, magazine papers, conference papers, some survey papers, books and literature where reviewed. Material and resources used in this study is specifically related to the area of research work on several types of attack on MANET, where also reviewed to explore the best security solution to detect jellyfish attack node in mobile ad hoc networks.

1.5.2 Evaluation of Proposed Scheme

The main contribution of this research work is designing and developing security mechanism for jellyfish node attack and evaluates the effectiveness of the scheme in terms of communication overhead, end to end packet delay, specific node and links delay, networks throughput, detection and mitigation probability and latency measure. In this paper we will use network simulation tool called NS2 to simulate and evaluate the effectiveness of the scheme.

1.5.3 Designing and Implementation

The model and algorithm of the proposed scheme will be designed in the design phase. Due to the convenient issue the simulation of the proposed scheme will be conducted by network simulator two in order to make sure that the effectiveness of the scheme for detection of jellyfish attack nodes and evaluate its performance based on the evaluation metrics mentioned above.

1.6 Scope of the study

The scope of this study is limited to designing and simulating the security mechanism to detect the jellyfish attack node in MANET. The proposed scheme consists two phases. Balancing the overall networks delay (i.e., delay from source to destination during the transmission of packet) and also balancing transmission delay (i.e., Transmission delay. In the second phase, identifying the jellyfish node that introduces the extra delay to the packet while transmission. Furthermore, this study will aim to detect the jellyfish attack node from the path which is selected for transmission of packet and mitigate the detected node from the path. Especially this is more focused on transmission delay because the most probable place to generate extra delay to packet (holding period) is focused. Computing network delay and finding holding period formation is the main course of action in this scheme. In general, the proposed scheme is limited to the following tasks.

- Identify the node that introduces holding period to the forwarded packet
- Detecting and preventing the jellyfish attack node from path candidates
- Finally formulating the jellyfish attack node behavior among the trustworthy mobile ad hoc networks' node.

1.7 Contribution

The main contribution of this thesis is:

1) A novel jellyfish attack node detection scheme is proposed and extremely evaluated theoretically and in simulation. A jellyfish attack node detection is designed based on the delay generated during the transmission of packet from node to node (i.e., from source node to destination node) in the case of mobile ad hoc networks. It shows promising performance compared to other competing detection scheme in terms of communication cost, detection probability, prolonging network delay and latency consumption per packet.
2) The result where promising and the scheme exhibited the desired properties: path information, node information, optimal jellyfish attack detection, optimal packet delay calculation, optimal routing solution for minimizing packet delay time and significantly prolog packet delay time in the network. In summary the scheme of detecting jellyfish attack node during the transmission of packet designed in thesis not only support detection

of attack node, but also able to decrease network congestion and increase network performance.

1.8 Organization of the study

This thesis work comprises of six chapters. Chapter two covers the literature review on MANET such as different type of networks, Characteristics of mobile ad hoc networks design challenges, application of MANET, routing protocol in MANET. Chapter three discuss about related research work from different source. In chapter four we design the proposed system. in chapter five we present an experimental setup, simulation result and performance evaluation, finally conclusion, recommendation and feature works are included in chapter six.

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Overview

The computing device is involved the different evolution of paths to becoming the inseparable part our everyday life. We envision that this can happen that by reducing the physical appearance of the computing devices and provide them to access computing capability over broadband network using lightweight deices. This evolution introduces that the movement of the computer from insulated and sealed rooms to body computing devices. With the introduction of computer networks and advance in micro-electro mechanical system, a new technology paradigm has emerged since the last decades so called ubiquitous computing. Particularly the aim of this paradigm is to make many computers to be available throughout the physical environment but making them effectively hidden from the user. Recent advances in wireless communication and digital electronics have enabled the production of low cost, low power and smart device such as smart phones, Personal digital assistant, radio frequency identification systems, wireless sensor networks, mobile ad hoc networks and many other computing technologies.

2. 2 Overview of MANET

The mobile ad hoc networks are autonomously self-organized networks without fixed topology. In such networks each node acts as both router and host at the same time. All networks' nodes are equivalents to each other and can move out or join in the networks freely. The mobile nodes that are in the radio range of each other can directly communicate and transfer the necessary information. All networks node has a wireless interface to communicate with other node in the range. These kinds of networks are fully distributed and can work at any place without the help of any fixed infrastructure as access points or base station.

Mobile ad hoc networks comprise of wireless node that communicate each other by exchanging the information. The path chosen for transferring the information from one node to other node is called routing and the protocol used is called routing protocol [18]. The requirement of routing protocol is to send and receive information among the node with best suited path with the most minimum delay. Correct and efficient route establishment between a pair of nodes is the primary goal of routing protocol. Many routing protocols for MANET have been proposed earlier performance analysis of routing protocol is a significant challenge in the research area.

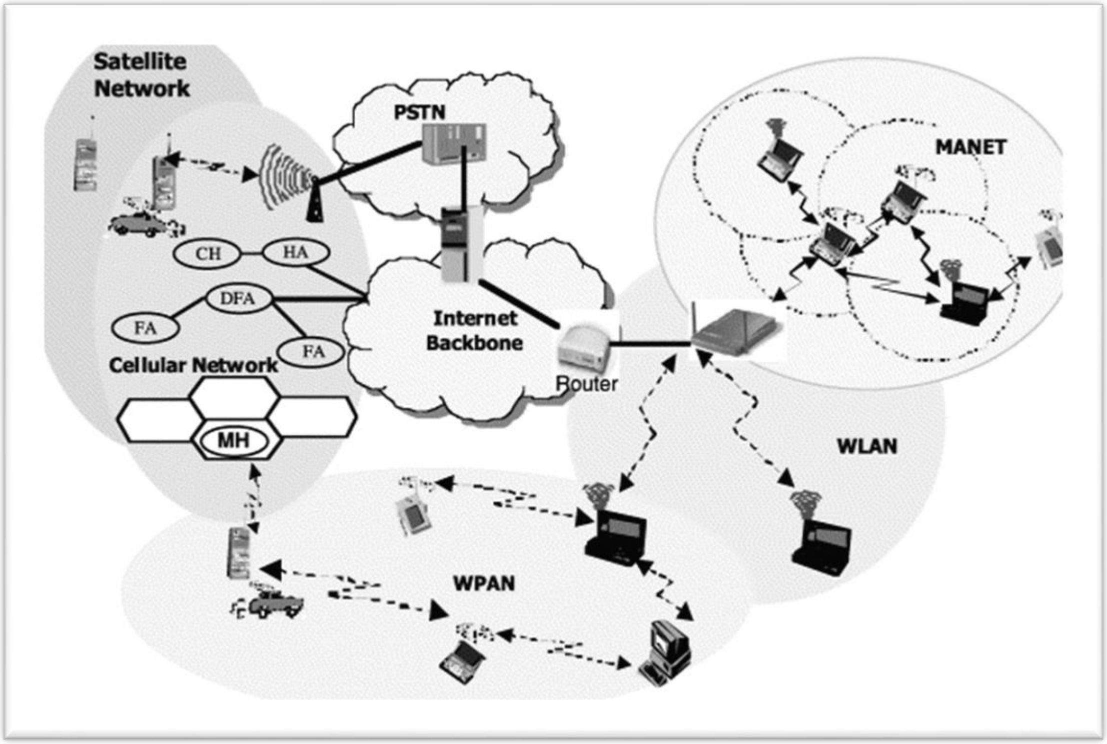


Fig 1: routing protocol

2.2.1 Characteristics of MANET

Mobile ad hoc network is characterized as:

Energy constraint: - Energy efficient is becomes the major design issue as nodes in MANET rely on the pattern power.

Multi-hop routing: - In mobile ad hoc network there is no dedicated central authority (Every device act as router and routing the packet to the neighbor nodes).

Dynamic topology: - due to arbitrary movement of nodes at different speed and external factor such as noise, the topology of the network changes randomly and unpredictable manner.

Mobility of the node: - mobility in ad hoc network causes topology change and lead dynamic, unpredicted and frequent network topology which are requiring an efficient routing protocol.

Autonomous and infrastructure less: - in mobile ad hoc networks each node acts as both router and host.

Distributed operation: - the node in MANET is connected in wireless link and the communication among nodes are wireless. The control of network is distributed among each node.

Light-weight terminals: - mostly the nodes in MANET have less CPU capability and have a little storage and memory size.

Link stability: - in order to provide the reliability of mobile ad hoc networks link stability have a major role and as MANET is mobile link stability must be considered.

Security threats: - in mobile ad hoc networks packet is transferred from source node to destination node, it passes different nodes which is in between destination and source node, by default all nodes in between source node and destination node is trusted, but there is node that attacks the packet in between source and destination node.

***Design issue*: -** while designing the network topology in MANET the following parameter has always been challenging the performance of ad hoc networks.

a) Data rate enhancement: - one of the key parameters which is used to determine the performance of any routing protocol is throughput. With the increased use of wireless device, providing method to enhance the throughput has thus far been a challenge.
b) Providing reliability: - since ad hoc networks are formed on the go, in ad hoc networks the link stability for reliable communication is all ways a challenge.
c) Power demands: - because of the nodes in ad hoc networks device have limited battery power, long lasting energy of these device has always been a challenge. So that how to efficiently use the residual energy of node in ad hoc networks is the critical issue.

d) Security: - the data transmitted and available resource in ad hoc network is greatly exploited due to the existence of malicious nodes. The security issue in ad hoc network is fustily growing and its currently the wide area of the research.

2.2.2 Application of MANET

Mobile ad hoc network has multiple application, some of the typical applications of mobile ad hoc networks are:

personal area networks; device like laptop, PDAs and mobile phone create a temporary network of short range to share data among each other, such networks were called personal area networks.

Civilian environments: MANET finds its use in many civilian activities like taxi cab networks, in meeting rooms, etc.

Collaborative work: the need for collaborative computing more is very important for some business environments where people do need to have information sharing on a given project and outside meeting to cooperate.

Military battel field: in order to exchange data or information between solders, military information head quarter, MANET allows the military to take merit of common place wireless network technology. Conferencing: one of the most known applications of mobile ad hoc network is mobile conferencing. For mobile user where they need to collaborate in a project outside the typical office environment designing an ad hoc network is essential.

Emergency service: the other most important role of application in ad hoc networks domain is responding to emergency situations such as disaster recovery.

Home networking: devices such as computers, laptops, smart mobile phone can create an ad hoc network at home where each device can communicate with the other without taking their original point of attachment in to consideration.

2.3 Routing protocol in MANET

Routing is a process of selecting the route in network for transmitting packet from one node to the other node (i.e., from source node to the destination node). Routing is the way of transmitting the packet from source node to destination node through one or more intermediate nodes. It basically involves two processes like finding an optimal routing path and transfers the packets in the internetworks. Routing information of the node is maintained in routing table. The routing table maintains the information of neighbor nodes and about possible destination. However, the potential problem of this technique is that some of the destination node is unreachable. The following are the main activities involved in the data transmission through the network:

- Determination of optimal routes between source and destination pair
- Delivery of packets to the correct destination node.

Routing protocols are used for finding and selecting the path for data transmission. The routing protocol in mobile ad hoc network also specifies how mobile device in the network exchange the information with their neighbor's node and report changes which happened. Optimal path selection must be needs consideration since ad hoc network have lower available resources when compared with infrastructure-based networks. To select optimum rout for forwarding the packet across the network, routing network uses metrics such as bandwidth, delay, energy consumption, hop count, etc. routing algorithm maintain routing table in order to ensure route selection process, which include the total route information for the packets. Based on their algorithm, this information of the path varies from one routing protocol to the other.

2.3.1 Property of MANET Routing protocol

Ad hoc routing protocol consist many properties and some of the common property of mobile ad hoc routing protocol are:

i) Distributed operation: in mobile ad hoc networks, routing protocol should not be depending on central administration which means that routing protocol should be distributed.

ii) Loop free: in order to avoid the wastage of resource such as energy and bandwidth ad hoc routing protocol should establish the loop free path which improve the overall performance of the network s.

iii) Unidirectional link support: for mobile ad hoc network routing protocols unidirectional path is important that can handle a situation where two unidirectional links form the only bidirectional connection between the nodes.

iv) On demand-based operation: in order to minimize the control overhead in the network and to provide the better utilization of the networks the ad hoc routing protocol network should be reactive.

v) Sleep period operation: since ad hoc network node may have energy constraints, nodes may want to stop transmitting and receiving the packet for arbitrary time period.

2.3.2 Type of Routing Protocol in AODV

The Traditional routing algorithms such as link state and distance vector not have scale in large mobile ad hoc networks. This is because periodic or frequent route updates in large networks may have consumed significant part of the available bandwidth, increase channel contusion and it may require each node to frequently recharge their power supply. To overcome the problem associated with the link state and distance vector algorithms a number of routing protocol have been proposed for mobile ad hoc networks. Mobile ad hoc routing protocol networks are classified as proactive (table driven), reactive (on demand) and hybrid (the collection of proactive and reactive) protocol.

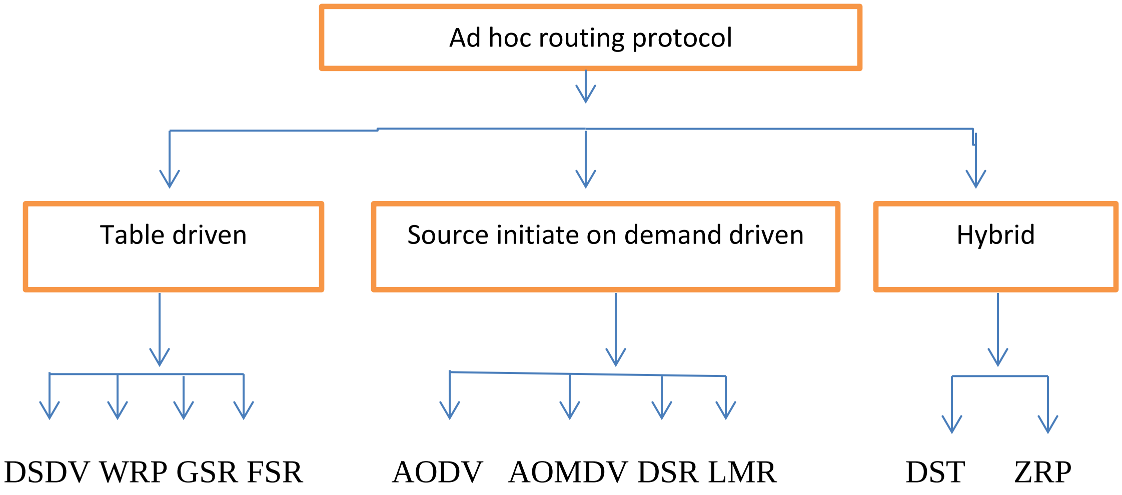


Fig 2: ad hoc routing protocol

In proactive routing protocol the paths to all of the destinations are determined at the start up and path can be maintained. This can be achieved using periodic route update process. But in reactive routing protocol, the routes are initiated by the source node only when they are required. This can be achieved by using path discovery process. The feature of both proactive and reactive protocol is integrated into one hybrid routing protocol. Each group of routing protocol has a number of routing protocol techniques, for finding optimal path between source and destination nodes and also for routing packet.

2.3.3 Proactive versus Reactive Approach

According to their routing mechanism, routing protocol in ad hoc network can be classified as reactive and proactive routing protocol [27]. In proactive (driven table) routing protocols, mobile device in ad hoc network should keep the list of paths to all possible destination nodes. In these types of protocol when the information needs to be transmitted the path is already known and the packet is forwarded immediately toward the intended destination node and any changes which happen in the topology are propagated through the network. Some examples of proactive routing protocol include: destination sequenced distance vector routing, wireless routing protocol, global state routing, fisheye state routing, etc.

In reactive (on demand) routing protocol, the path between source and destination pair is established when needed (on demand). The path is maintained by route maintenance process and if the route is failed after the path is established. Some examples of reactive routing protocol include: ad hoc on demand distance vector routing, ad hoc on demand multipath distance vector routing, dynamic source routing, and cluster-based routing protocol, etc.

The advantage of using proactive routing protocol is that new communication with arbitrary destinations experience minimal delays. But it has also its own disadvantage. The main problem of these routing protocols is that to update routing information at all nodes it is suffered from additional control overhead. To address, on demand protocol adepts the inverse approach by finding the path of destination node only when needed. When compared with proactive routing protocols, reactive (on demand) routing protocol consumes much less bandwidth. But for discovering the path to destination prior to the actual communication they will experience the long delay. Reactive routing protocol may also generate excessive traffic in comparable with proactive protocol. Because they need to broadcast rout request if the route discovery procedure is required frequently. But on demand routing protocol can decrease routing overhead and minimize, energy, storage and bandwidth requirement when compared with proactive routing protocol.

2.3.4 Clustering and Hierarchical Routing protocol

The most important issue in mobile ad hoc network is scalability. In ad hoc network scalability is the ability of the network to guarantee an acceptable level of service to packets, even in the network with large number of devices. For proactive routing protocol, when the number of devices in the network increases, the number control packets in the topology increase. This may increase the routing overhead and the node may consume large amount of remaining energy and available bandwidth. In on demand routing protocol large number of control message (route request and reply) to the entire network may eventually become packet broadcast storms. Typically, when the network size increases beyond certain thresholds, the computation and storage requirement become infeasible. The frequency of routing information update may be typically increased when mobility is considered, thus worsening scalability issues. In order to solve this problem and to produce efficient and scalable solution hierarchical routing is used. The idea of hierarchical routing is combining the nodes in a certain group and assigning nodes with different functionality outside and inside group. Both update packet size and the routing table size are reduced by maintaining them only in the part of the network. In reactive routing protocols, in the way by limiting the range of broadcasting route request also help to increase efficiency [16]. The most popular mechanism of building hierarchy is to combine group nodes close to each other into clusters. Each cluster has a cluster head which leads node to enable communication between nodes on behalf of this clusters. Some examples of hierarchical ad hoc routing protocols are: zone routing protocol, zone based hierarchical link state routing protocol, distributed spanning trees-based routing protocol, etc.

2.3.5 Review of Ad Hoc Reactive Routing Protocol

Reactive routing protocol is other types of ad hoc routing protocol, unlike to proactive routing protocols, in reactive routing protocol all update routes are not maintained at every node. Instead of that the paths are created as and only when required. When a source node needs to transmit packet to destination node, it initiates the route discovery mechanism to find the route to the destination node. In these types of routing protocol several typical reactive routing protocols are generated.

2.3.6 Ad hoc on demand distance vector routing (AODV)

Ad hoc on demand distance vector adopts the concept of destination sequence number adopted from destination sequenced distance vector routing**.** This protocol also adopts concept DSR which is modified on demand broadcast route discovery approach. The source node initiates a path discovery process and broadcast a route request message to its neighbors when a source node needs to send the packets to some destination and does not have valid route to that destination[24]. Then after until the RREQ control packet reaches the destination node, the neighbor node forwards the route request to its neighbor.

Each node in AODV maintains its broadcast ID and its own sequence number. Each RREQ message contains the sequence numbers of the source and destination nodes and is uniquely identified by the source nodes address and broadcast ID[22]. In order to ensure loop free route and use of updated route information. AODV protocol utilizes destination sequence numbers. If intermediate node has the path to the destination node whose destination sequence number is greater or equal to that contained in the RREQ message, they can replay to the RREQ control packets. To setup the reverse path, each intermediate device keeps the address of the neighbor node from which it received the first copy of the RREQ control packets and duplicated copies of the RREQ control packets are dropped[25]. After the route request control packet reaches the destination, the destination node responds a route replay packet back to the neighbor node from which it first received the RREQ control packet. The intermediate nodes along this path set up forward path entire in their routing table when k2the RREP message is routed back along the reverse path. When nodes detect a path failure or change in neighborhood, a route maintenances produce is invoked. If a source node moves from one position to other, it can reinitiate the path discovery process to find a new path to the destination node.

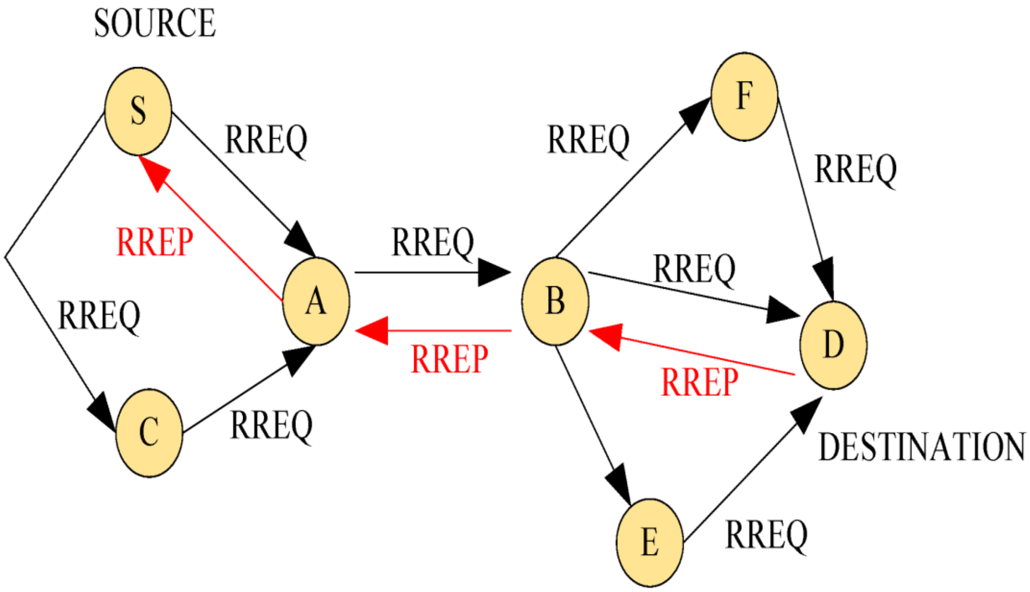


Fig 3: AODV routing

2.3.7 Ad hoc on demand distance vector multipath routing

AODVM is an extension to AODV for finding multiple node disjoint paths. Instead of discarding the duplicate RREQ packets, intermediate nodes are required to record the information contained in this packet in the RREQ table. For each received copy of an RREQ message, the receiving intermediate node record the source that generated the RREQ, the destination for which the RREQ is intended, the neighbor that transmitted the RREQ and some additional information in the RREQ table. In this protocol intermediate reply nodes are precluded from sending back the RREP control message directly to the source. Destination node updates its sequence number and generates the RREP packet when it receives the first RREQ packet from one of its neighbor nodes. The RREP packet used in this protocol contains an additional field called "last hop ID" which announce the neighbor from which the particular copy of RREQ control packet was received. This RREP control packet is sent ac to the source node via the route traversed by the RREQ message. The destination node updates its sequence number when it receives duplicate copy of the RREQ control packet from other neighbor and generates RREP packets for each of them. This RREP control packet also contains the ID of its respective last hop nodes like the first RREP packet. When an intermediate node receives the RREP packet from one of its neighbors, it deletes the entry corresponding to this neighbor from its RREQ table and adds a routing entry to its routing table to indicate the discovered route to the originator of the RREP packet (the destination). Then the node distinguishes the neighbor node in the RREQ table through which the path to the source is the shortest and transmit RREP control message to that neighbor node. In order to ensure that a node does not participate in multiple paths, the entry corresponding to this neighbor is then deleted from the RREQ table. The node deletes the entry corresponding to the transmitting node from its RREQ tables when they overhear any node broad casting an RREP message. The destination node is an unaware as to how many of these RREP control packet that it generated actually made it back to the source node and the intermediate nodes make discussion on where to transmit the RREP control packets. Thus, for the source to confirm each received RREP control message, the route conformation message is necessary. The route conformation message can be added to the first data packet to be sent on the corresponding route. It will also contain information with regards to the hop count of the path and the first and last hop relays on that path.

CHAPTER THREE

3. RELATED WORK

Deepika and sharad saxena. (2018), presents performance evaluation of AODV with self-cooperative trust scheme using jellyfish delay variance attack. In these study two basic step approaches is used as self-detection and neighbor detection approach. Under self-detection each node detects itself and broadcast the information to its neighbors[22]. This is followed by the cooperative detection. In cooperative detection node will send the hello message to the neighboring node. Therefore, each node on receiving the hello message detects itself and its neighbors. Step 1: node x sends its hello message to its neighbor. Step 2: on receiving the request packet neighbors y checks for the history. If the neighbor history has the number of requesting node x, it will replay to the x, and increase the trust value of x. step 3: on receiving the route replay the node x checks for the replied node and if the number is found then it will increase the trust value for y. step 4: this cooperative trust-based scheme will be followed at each occasion before the actual transmission will be taken place[29]. Step 5: end.

In order to assure packets, reach the destination, the network has primary responsibility to provide a secure mechanism between all nodes (destination, sender as well as intermediate nodes)[30]. In MANET, if any one malicious node enters the operating network, it can lead to in correct network performance and network will show the following outcomes.

- Tremendous increase in the number of the junk packets, in turn, preventing the trustworthy node to transmit data packets in the network.
- Generation of fake control packet carrying incorrect topology information and impacting routing table.
- Delay in packet transmission and impacting overall throughput in the networks.

Sudeep Tan war et al. (2017) study and presents accurate prevention and detection of jelly fish attack in MANET. In their study, in order to defend the MANET networks against jellyfish attack, a novel methodology called accurate prevention and detection of jellyfish attack detection is proposed[6]. Node property based hierarchical trust evaluation is carried out in their proposed technique. As a result, jellyfish attacks are prevented by choosing only trusted nodes for route path construction. In their proposed technique, support vector machine is utilized for packet forwarding behavior learning[16]. This technique guaranties the detection of jellyfish attack with high precision.

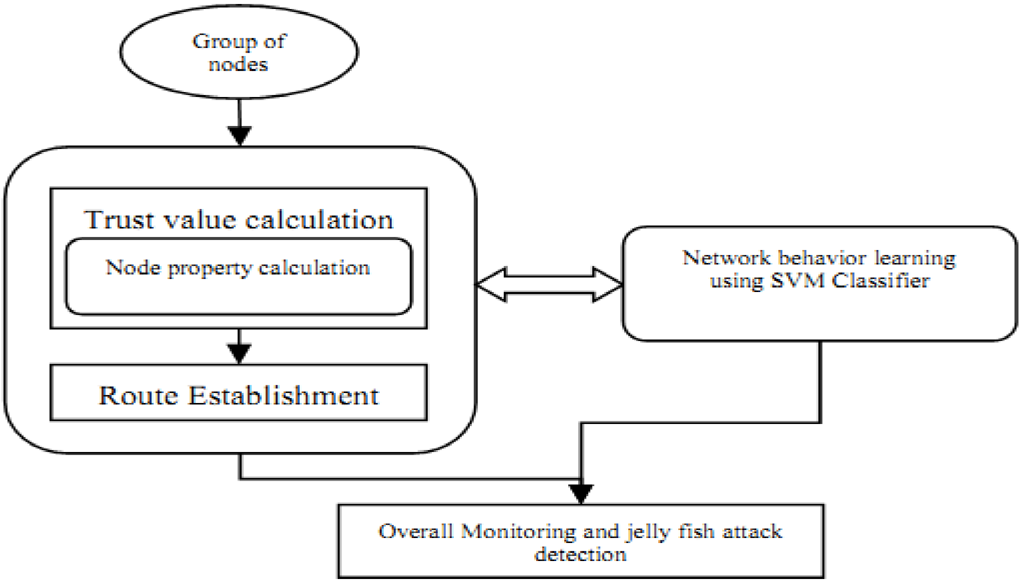

Group of nodes
Trust value calculation
Node property calculation
Route Establishment
Network behavior learning using SVM Classifier
Overall Monitoring and jelly fish attack detection

Fig 4: Mechanism of APD-JFAD for combating jellyfish attack in MANET

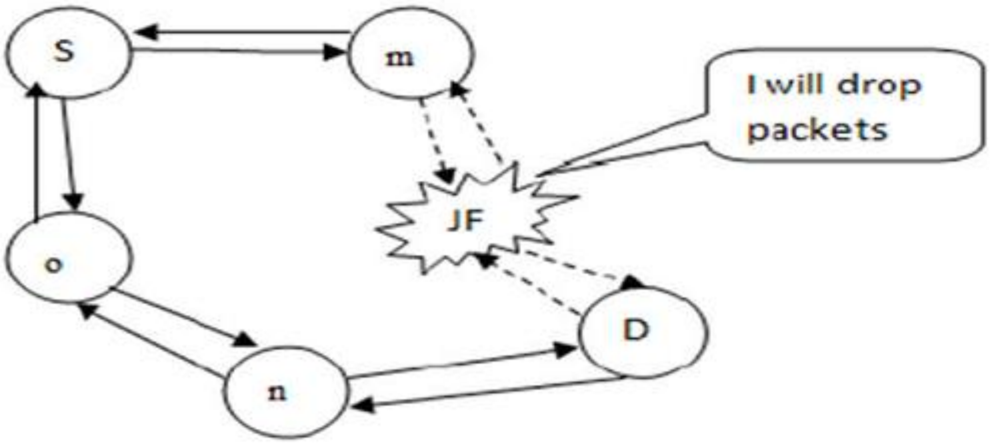


Fig 5: Overview of Jellyfish attack in MANETs

Lakshmana Narayana et al (2020) proposes the method that is efficiently identifies the malicious nodes in the MANETs. When MANET is established and routing process is completed, a node is selected from the MANET called network block monitoring node hat is used for monitoring and analyzing the blocks generated after every transaction done by a node[26]. The transaction done by every node is blocked and a block is created that link with other blocks forming block chain.

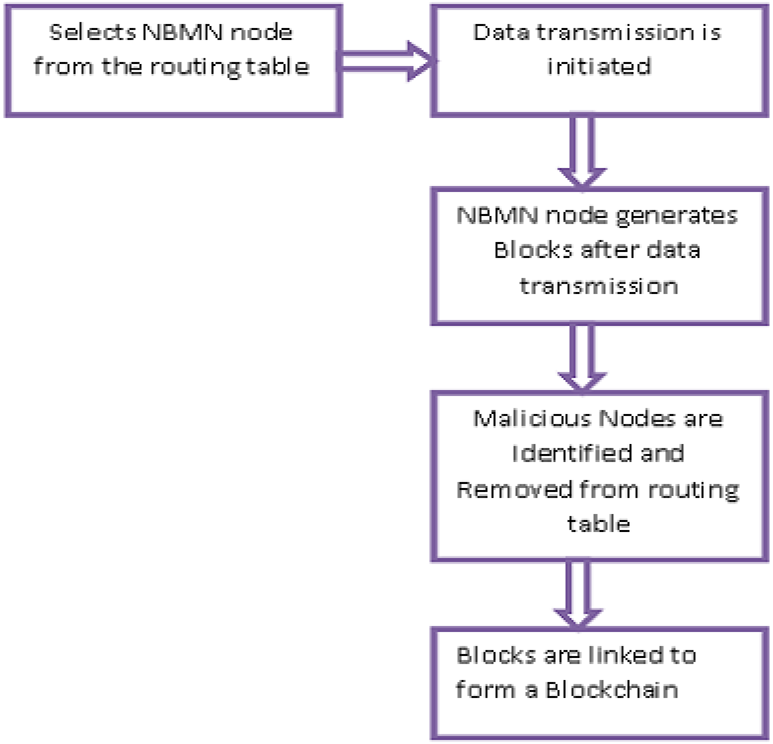

Selects NBMN node from the routing table
Data transmission is initiated
NBMN node generates Blocks after data transmission
Malicious Nodes are Identified and Removed from routing table
Blocks are linked to form a Blockchain

Fig 6: Time interval block chain network block monitoring framework [18]

Patel Pooja B. et al (2017) proposed approach how to detect and prevent jellyfish attack by considering sending time and receiving time of packet threshold time of packet and load of networks. They proposed approach first check the receive time and send time difference of packet with threshold value of time packet. If result is less than continues process of packet forwarding otherwise delay occurs[29]. For that check load of networks. It is greater than predefined load threshold of network than delay occurs due to congestion and result less that means jelly fish attack node present. In this way they use to detect jelly fish attack node[31]. They present that threshold time is the average time of sending and receiving data packet. Load of network depends on capacity of the networks for resource access, attribute behavior, etc.[32]. They conclude that, if delay occurred due to congestion, then disable retransmission of same packet and enable selective ACK. Delay occur due to jellyfish attack node, discarding the jellyfish node and that replay. And also choose multipoint relay node and forwarding packets[33]. MPR node is not malicious node this way attack is prevented.

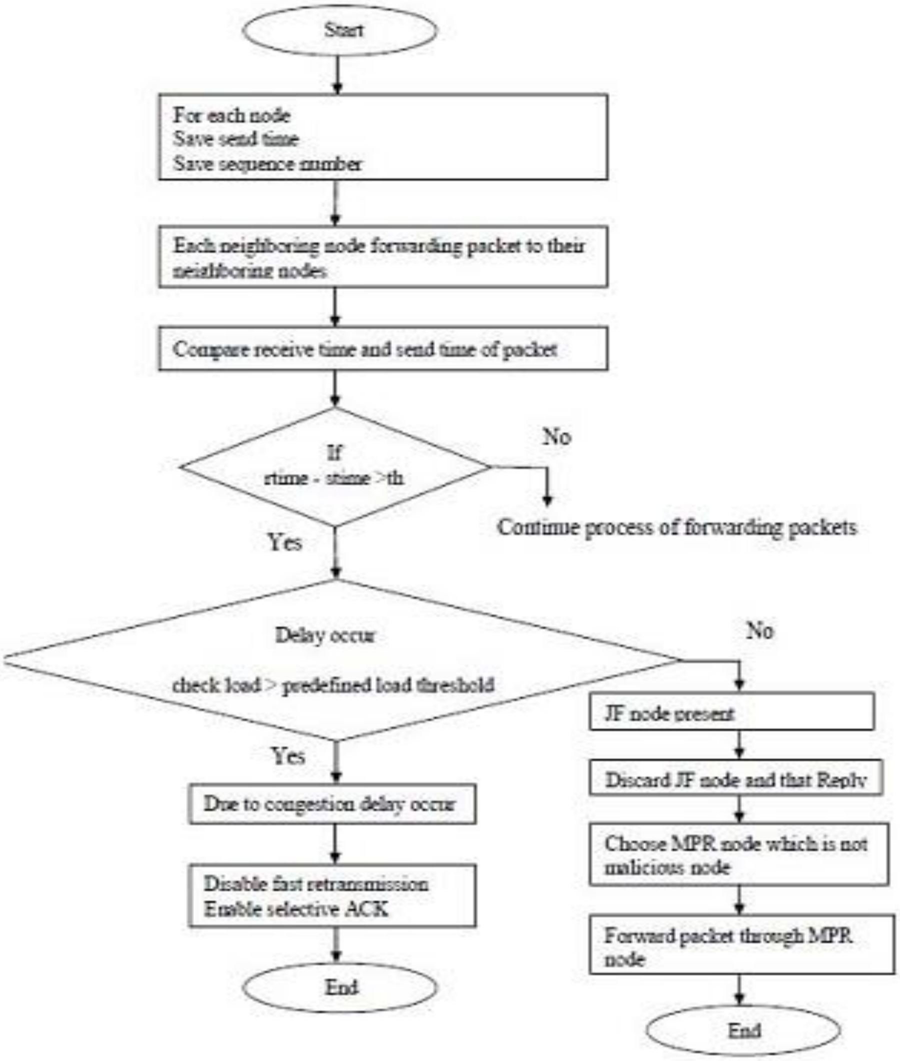

Start
For each node
Save send time
Save sequence number
Each neighboring node forwarding packet to their neighboring nodes
Compare receive time and send time of packet
If
rtime - stime >th
No
Continue process of forwarding packets
Yes
Delay occur
check load > predefined load threshold
No
JF node present
Discard JF node and that Reply
Choose MPR node which is not malicious node
Forward packet through MPR node
End
Yes
Due to congestion delay occur
Disable fast retransmission
Enable selective ACK
End

Fig 7: Proposed work flow [13]

Manikant sharma and Praveen tripathi (2015) in this study the performance of routing protocol like ad hock on demand vector routing and dynamic source routing. And they have studied the various performance metrics packet delivery ratio routing overhead, end to end delay throughput and path optimality. In this paper they have studied the routing protocol AODV and DSR over various numbers of nodes and various speeds. Here they study five performances metrics like packet delivery ration, routing overhead end to end delay, throughput and path optimality. And they said their study show that the behavior of routing protocol varies as the number of nodes, speed of nodes (node mobility models) is changed. The performance of routing protocols varies with the above models.

Simranpreet kaur et al (2015) in this paper the types of attacker which is called as jellyfish attack is discussed. They proposed that jellyfish attack is one of denial-of-service attack and also, they report that these types of attack are occurred due to vulnerability of TCP. And also, they proposed that AODV is a routing protocol which is affected by jellyfish attack and it is one of passive attack which is difficult to detect because the attacker don't disobey any of the protocol rule. They wrote that cutting down the good put of the traffic to minimize or zero either by dropping the data packet or by changing the order of the data packets is the main objective of this attack. It is similar to black hole attack; the only means by which it is different from black hole attack is that, in black hole attack the attacker node drops the data packet, but in jellyfish attack packets are delayed before transmission of packets and after reception of packets in the network. Jellyfish attack targets closed loop flows because such flow reacts to the network condition like loss of packet and packet delay. The first step to be taken by jellyfish attacker is to gain access to the routing mesh and intrude in to the forwarding groups. Jellyfish attack is classified as three types.

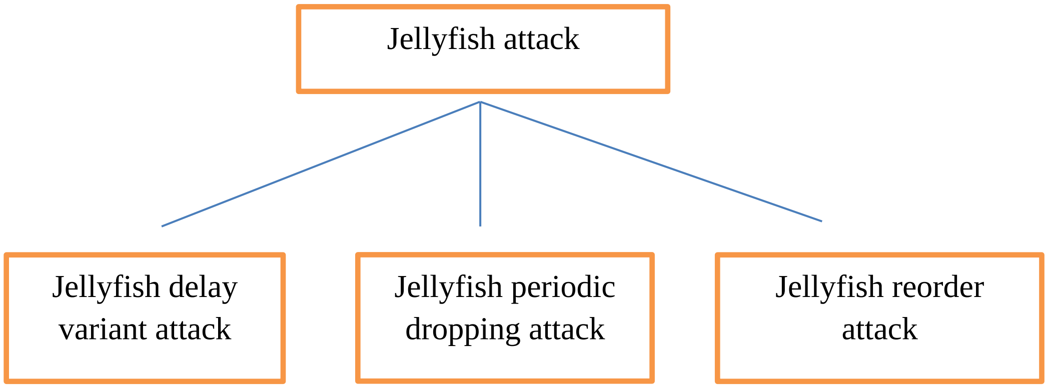

Jellyfish attack
Jellyfish delay variant attack
Jellyfish periodic dropping attack
Jellyfish reorder attack

Fig 8: Types of jellyfish attack

3.2 Jellyfish Attack

Jellyfish attacker keeps up acknowledgement with the two situations like control and information convention. Because it acts compliant to both data and control protocol which make it difficult to detect and prevent[7]. Therefore, jellyfish attacker is difficult to detect until after the sting. The jellyfish attacker right off the bat actualizes the hurrying attacker to access the directing cross section. On the off chance that ends up successful, at that point it defers ever one of the packets by an irregular time frame [10]. As there is no utilitarian refinement among portable nodes in MANET, a transitional node can present a basic weakness for TCP blockage control system. There are different variations of the jellyfish kinds of attacker. Those are jellyfish reorder attack, jellyfish periodic dropping attack, and jellyfish delay variant attack.

3.2.1 Jellyfish Reorder Attack

In these types of attack, malicious node forwards the packet in arbitrary order from the queue, resulting in zero good put, instead of forwarding them in FIFO order. Packet can be placed in a random buffer in place of FIFO buffer[8]. Buffer is reordered by JF attack node and packets are sent from the buffer. At the destination side when packet do not arrive in actual order duplicate acknowledgement is sent to the sender, if three duplicated acknowledgements is received at the sender side, retransmission of packet is started without waiting for transmission timeout. Even if the packets are reached the destination side, sender still believes that packet is lost and may keep retransmitting the packet again.

3.2.2 Jellyfish Periodic Dropping Attack

This attack is usually possible at relay nodes. Due to congestion anode is forced to drop packets and if node drops packets periodically than TCP throughput will reduce to zero.

3.2.3 Jellyfish Delay Variance Attack

In this attack order of packet are not altered but packets are delayed in an arbitrary manner. Malicious node has to acquire access to the routing path and after that the packet it receives are delayed before forwarding. TCP being a reliable protocol makes sure that sender receives acknowledgement for every packet sent, but if jellyfish attacker node is present then it delays the acknowledgement. When timeout period for acknowledgement is finished than senders assume that packet is lost and it retransmit the packet which results in congestion.

Table 1: Comparison the types of jellyfish attack

Attack	Purpose	Cause	Effect
Jellyfish reorder attack	Reordering of the packet is done	Due to vulnerability of TCP	Result in degraded throughput and retransmission
Jellyfish periodic dropping attack	Jellyfish node drops the packet in periodic manner	Due to malicious period chosen by attacker node	Attacker try to maintain synchronicity with transmission window and can cause near zero throughput
Jellyfish delay variant attack	Packets are delayed in arbitrary order	Can be exploited through TCP	Increase delay variance and lead to congestion inference

Summary

We summarized the work that are directly or indirectly related with our study in the following table

Table 2: Summary of related work

Author	Title	QoS metrics considered
Vijay laxmi and chhagan lal	Jellyfish attack analyze, detection and counter measure in TCP based MANET	Throughput, overhead incurred and end to end delay
Skudeep Tanwar	Accurate detection and prevention of jellyfish attack in MANET	Packet delivery ratio (PDR), throughput, dropped packet ratio (DPR) and end to end delay
Deepika and Sharad Saxena	Performance evaluation of AODV with self-cooperative trust scheme using jellyfish delay variant attack	For their performance measurement they use Average end to end delay and protocol throughput
Manikant Sharma and Praveen Tripathi	International journal of research on study of throughput and delay comparison of AODV and DSR routing protocols	Packet delivery ratio, routing overhead, end – end delay, end to end throughput and path optimality
Dr. V. Lakshman Narayana and Dr. Divya Mindhuncha Karavarthi	A time interval-based block chain model for detection of malicious nodes in MANET using network block monitoring node.	Packet dropping rate in the network
Simranpreet Kaur, Rupinder Deep Kaur and A.K Verma	Jellyfish attack in MANETs	Throughput and delay of the networks
Shruti Thapar, Sudhir Kumar Sharma	Detection and prevention policy of jellyfish attack in MANET	End to end delay, and throughput

<table>
<tr><td>Anjani Gargi, Sunil Kumar, Kamlesh Dutta</td><td>An analytical survey of state-of-the-art jellyfish attack detection and prevention techniques</td><td>Variation of jellyfish attack, throughput and end 2 to end delay of the network</td></tr>
<tr><td>Madhup Shrivastava, Monika Sahu</td><td>An improved AODV routing protocol for MANET</td><td>Packet delivery ratio, end to end delay, throughput</td></tr>
<tr><td>Danista Khan, Mah Zaib Jamil</td><td>Study of detection and overcoming blackhole attacks in MANET</td><td>RREQ, RREP, Data</td></tr>
<tr><td colspan="3">Research gap
Depending on the above study we summarized, we present the research gap of this study as follows:
✓ Detecting specific jellyfish attack node is not presented
✓ Preventing specific jellyfish attack node is not presented
✓ Holding period which is generated to the coming packet is not specified briefly
✓ Only checking end to end delay consideration is not enough to present specific jellyfish attack node in the path, presenting specific transmission delay on each node in the path is very important to detect jelly fish attack</td></tr>
<tr><td>Geletu H.</td><td>Improving packet transmission performance by detecting and preventing delay variant JFA</td><td>Throughput, End to end delay, transmission delay and packet delivery ratio
“find out holding period” based approach to detect JFA in MANET</td></tr>
</table>

CHAPTER FOUR

4. PROPOSED SOLUTION

4.1 Overview

The proposed solution focuses mainly on packet delay and finding out specific node that generate some delay. We call the extra added delay to packet as holding period. In this discussion, we identify the node that delays the coming packet, because these is the cause of degrade of the network throughput which gives degrade of overall network performance. These is mainly due to the attacker node called jellyfish attack. Jellyfish attack node introduces some delay to the coming packet and send. These causes ACK to delay. If ACK is delayed the sender node resend the same packet and the path is Bussey with specific packet. Wich affects the network performance. In our investigation we propose the method to detect and prevent node with these behaviors. This is by finding the holding period. In our case holding period is the extra added time to transfer packet.

4.2 Architecture of proposed Solution

In proposed solution we discussed more on network delay calculation and check whether there is attack node or note on the path in the network. As we discussed above in chapter three, jellyfish attacks node has the behavior of introducing new delay to normal networks delay and minimizes the throughput, as a result network overcrowding is occurred due to new delay added to the normal one. In this study we calculate the values of all types of delay in the network and compare with the outcomes during transmission of packets from source to destination. In this case delay type is discussed in the following. As a packet travel from one (host or router) to the subsequent node (host or router) along this path the packet suffers from several types of delay at each node along the path[8]. The following diagram shows the activity to detect and prevent jellyfish attack that generate extra delay to the coming packet, before transmitted to the next neighbor node link in the path[37]. These is done by calculating the total or end to end delay value and check if there is extra unknown delay generated or note. If there is extra delay, delay in each node which is served as intermediate node must be calculated and checked until the attacker is detected[25].

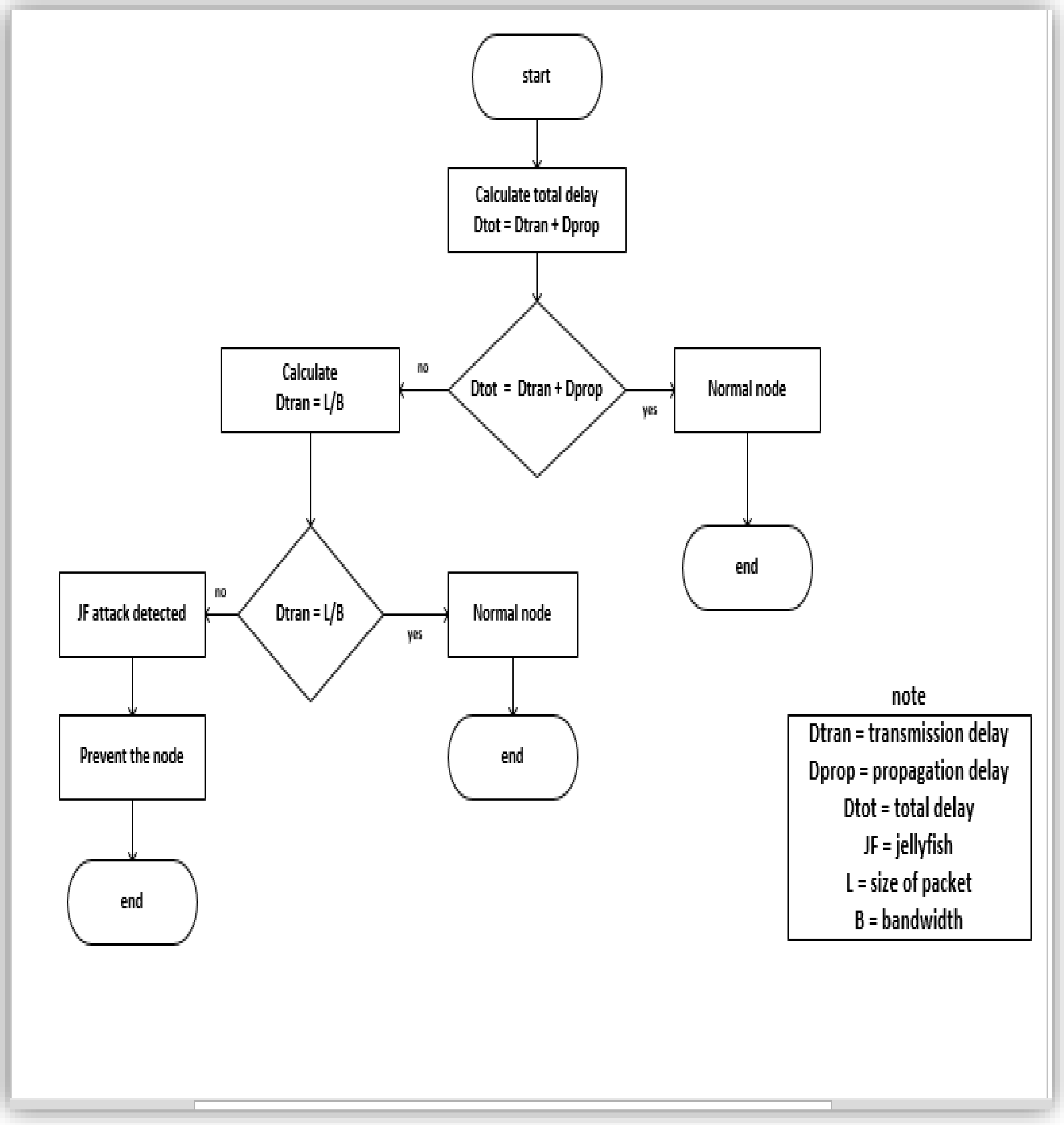


Fig 9: The flow of activity in proposed work

The following diagram shows the structure of the statement to show where the system is vulnerable to attack. Jellyfish attack is targeted to the moving packet, especially when packet is putted on the link from the node to move to the next node on medium. The diagram shows as when and how the packet is putted on the linked bandwidth and the formula used to compute the time taken by node to put the coming packet to the linked bandwidth.

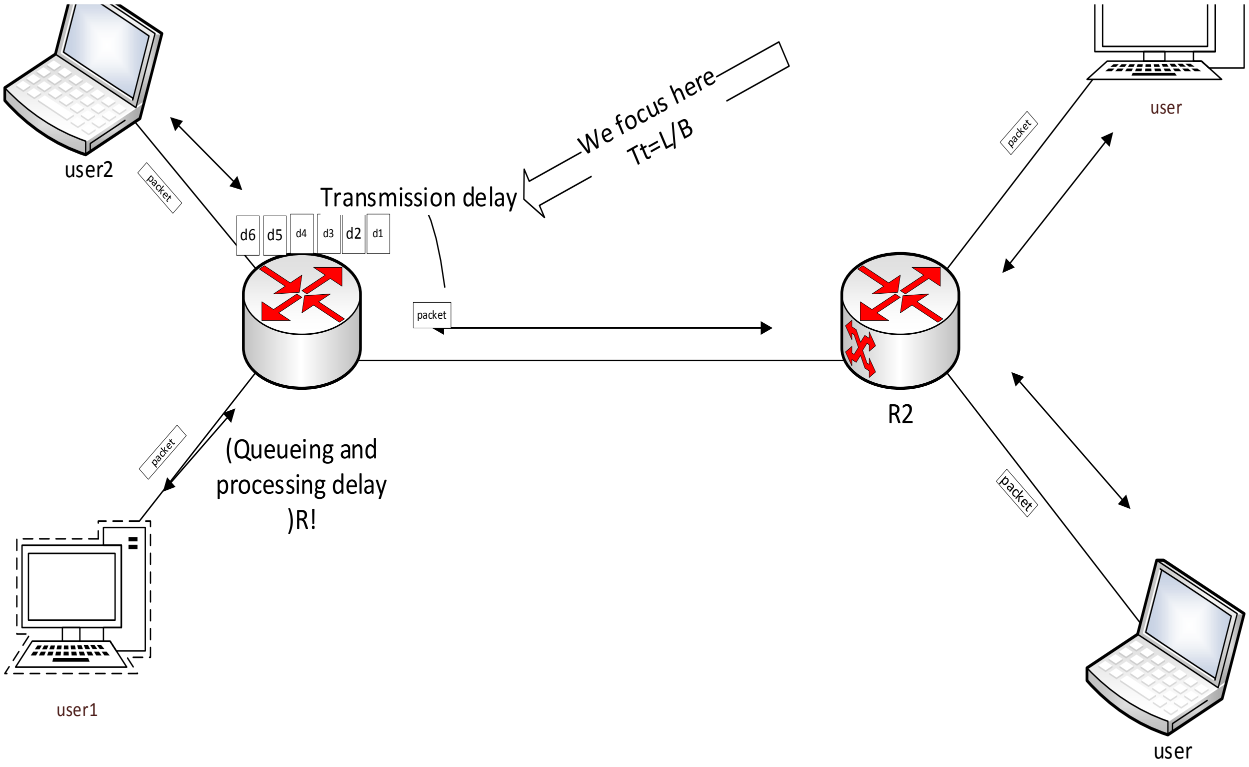


Fig 10: Shows network delay

Jellyfish delay variance attack node is the one which takes after all convention rules and subsequently hard to recognize and detection [40]. Jellyfish attack is a passive attack as the attacker disrupts the network from within. Jellyfish attacker becomes the part of routing mesh and introduces some amount of delay (holding period) before forwarding the packets[5]. When acknowledgment is delayed then the sender will not receive the acknowledgement within specified amount of time. Source node will assume that packet is lost and start retransmitting the packets. It leads to increase congestion and reduce throughput. Jellyfish attack targets closed loop flows because of which flow is affected by packet loss and delay.

The following diagram specify that how to detect jellyfish types of attack on a given networked path, each and every intermediate node have to be checked and identified if and if not, there is these types of attack or not depending on the calculated values from the given parameters. Jellyfish attack have its own specific behavior in the network that shown during transmission of packet. One of these behaviors is introducing extra delay to the coming packet and sends after some period of time, this period of time in our study we call it holding period[41]. Finding out the holding period of each intermediate node is our main work in this study. Holding period is the time that specific node takes to hold the coming packet before transmitting to the next neighbor node and it is additional time introduced to the packet. These types of delay is due to jellyfish attack, since we discuss that jellyfish attack have the behavior of introducing extra delay to the coming packet due to these network congestion is occurred and these decreases the throughput of the network[2]. Therefore, the appearance of such attacker in the network minimizes the performance of the network and the quality of service in the network. These is prevented by finding out the holding period of each node in the path especially the intermediate node and breaking link with the node that have holding time and establish new path in the link with the node which is free of holding period. The following diagram shows the detected attack node in the path having holding period in the path.

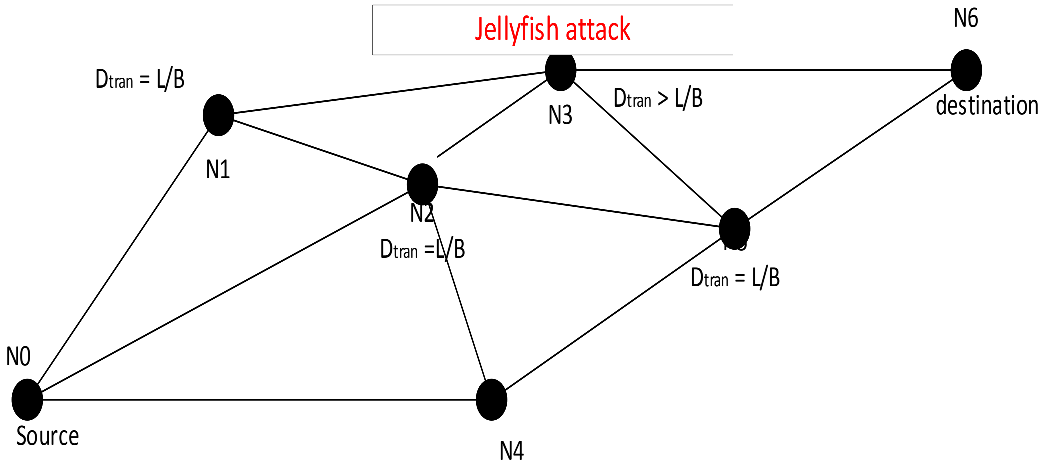


Fig 11: path with jellyfish attack node

In the fig 16 above the path is in the way of N0 ---- N1 ------N2------N3-----N5-----N6. From this path node N3 is detected as jellyfish attack after calculation of transmission delay result where the formula is performed as follows.

D_{tran} = $(\text{size } of\ packet)/\text{bandwidth}$ in second

Transmission delay for Node1 = $(packet\ size)/bandwidth$

Transmission delay for Node2 = $(packet\ size)/bandwidth$

Transmission delay for Node3 > $(packet\ size)/bandwidth$

Transmission delay for Node5 = $(packet\ size)/bandwidth$

Transmission delay for Node6= $(packet\ size)/bandwidth$

Algorithm for Transmission Delay…………………………

```
#Detection of JF node
For each node do
End
      Create JFPkt = broadcast data packet;
      Save end to end   time for a packet;
     Save transmission time for all packet.;
     Broadcast  JFPkt with TTL ;
      For each neighbouring node do
End
Broadcast  JFPkt  to  their neighbouring nodes;
# including  the node which sent  it
        For each node do
if(  holding period packet  received)  {
check transmission time.;
}
if(end to end time. present) {
compare recorded time and currently calculated time;
if(recorded transmission time>  current calculated transmission time )
sender node is JF node;
#Remove  the JF node from  forwarding path
# Each  node  neighbouring  JF  node  knows  that  it  is
JF node
For each node do
While( Route Discovkery)  {
If( routeReply received from JF node) {
Discard routeReply;
}
# RouteReply from an alternate path is chosen
# JF node never enters the forwarding path
```

These is because of the jellyfish attack always introduces extra delay while transferring of packet is done and, in our case, this extra delay generated by attacker is called holding period that causes the network congestion. After the node with holding period is detected, as we have shown in the above Fig16, the path that includes the node having holding period is broken and establishes the new path to send the packet from source node to the intended destination node with the path which is free of holding period. In this way we increase the performance of the network and the quality of service.

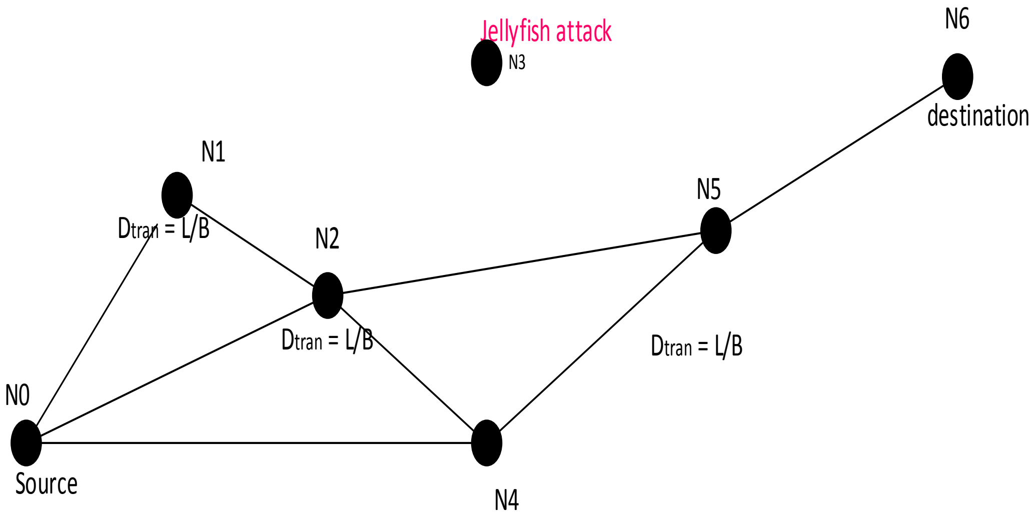


Fig 12: After detection of jellyfish attack node

In the fig 17 above the new path is established in the following way where all the node in the path delay is calculated and checked therefore the new path is

N0----------------N1--------------N2 --------N5--------N6.

N3 is detected as jellyfish attacker and prevented from the path after calculation of transmission delay. The calculation of transmission delay is done as the ratio of the coming packet to the linked in bandwidth in second, these is shown as the following format.

Transmission delay for node 3 $>$ $(size\ of\ packet)/bandwidth$ in second

Algorithm for Sequence……………………………………

```
#Detection of JF node
For each node do
End
Create JFPkt = broadcast data packet;
Save delay   time;
Save transmission time.;
Save sequence no;
Broadcast  JFPkt with TTL;
For each neighbouring node do
End
Broadcast  JFPkt  to  their neighbouring nodes;
# including  the node which sent  it
For each node do
if(  looping packet  received)  {
check sequence no.;
}
if( sequence no. present) {
compare receive time and send time;
if( rtime- stime >  current calculated one )
sender node is JF node;
#Remove  the JF node from  forwarding path
# Each  node  neighbouring  JF  node  knows  that  it  is
JF node
For each node do
While( Route Discovery)  {
If( routeReply received from JF node) {
Discard routeReply;}
# RouteReply from an alternate path is chosen
# JF node never enters the forwarding path
```

In this method the transmission delay of all node in the path is calculated and checked if it is normal delay or not, if it is greater than what expected from calculation, we assume it is an attacker node because there is the holding period if the calculated result and the default result measured by device is not the same. If there is holding period there is congestion of network that shows the behavior of jellyfish attack node.

Summary of Algorithm……………………………

```
# detect attack node
For each node do
Record the packet end to end delay
Record the transmission delay
Compare end to end network delay
If (end to end recorded delay > currently calculated delay) {
Compare transmission time;
If(transmission recorded delay > currently calculated delay){
compare the support vector machine value with the calculated value;
if(support vector machine information = node property calculated value)
the node is JF node}
else
check again and compare;
end
}
```

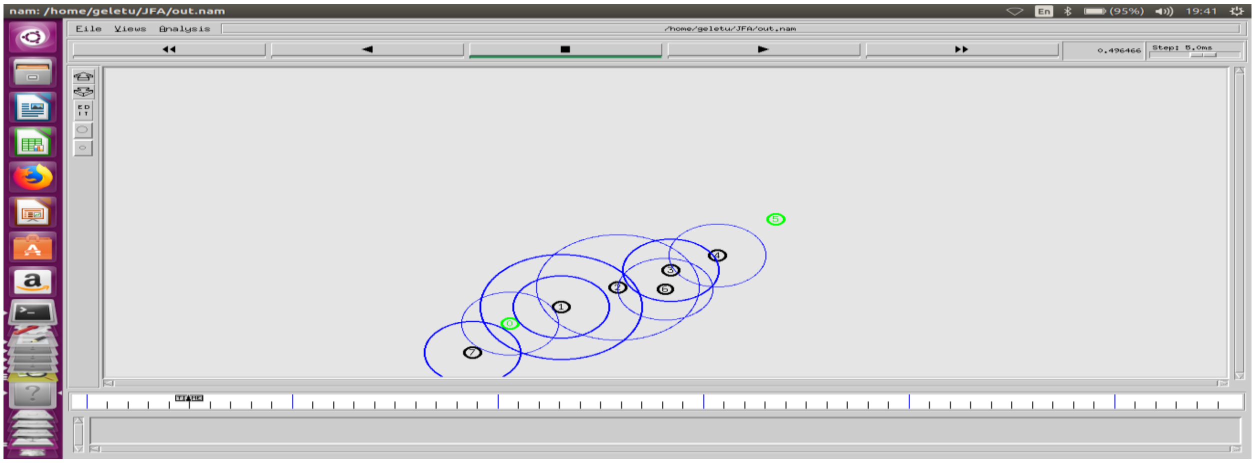
CHAPTER FIVE

5. IMPLEMENTATION

In this chapter we discuss about the implementation of our proposed system model which is discussed in previous chapter. It is a complex task to make comparison between existing system protocol with current proposed system protocol on a real time network. The simulation environment is used to simulate the currently proposed protocol with existing routing protocol[42]. For this study we select network simulator two as the simulation environment. Network simulator two is one of the most commonly used simulation environments for wireless network and it is used in many other studies discussed in the previous chapter.

5.1 NS2 Overview

Network simulator version two is an event driven simulation tool which can be used to study the nature of the communication network [14]. Network simulator 2 can be used for the simulation of both wireless and wired networks and protocol such as UDP, TCP and other different protocols. In networking study NS2 simulation tool has gained a lot of popularity due to its flexibility. The following is Ns2 simulation scenario



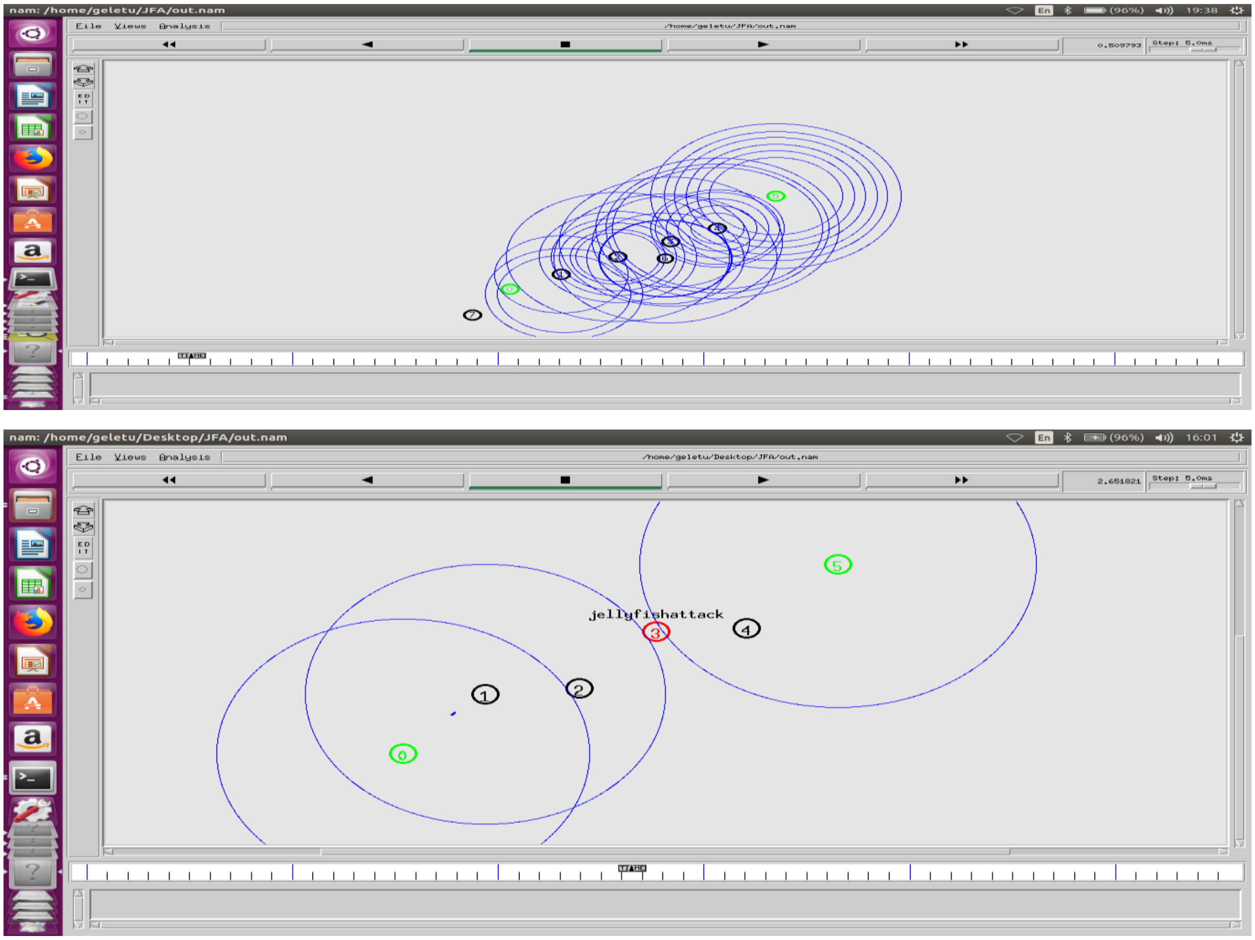


Fig 13: AODV with attacker scenario

5.2 Basic Architecture

Network simulator 2 simulation tool provides user with ns executable command that take on the name of tcl simulation scripting file as an input argument. User input the name of tcl script as an input argument of an NS2 executable command ns[43]. At the end of our simulation a trace file is created which is used to plot graph and to create animation. Network simulation 2 includes two key languages, which is C++ and object-oriented tool command language (otcl). The C++ defines the internal (a back end) of the simulation object and the otcl sets app simulation by configuring and assembling the object as well as scheduling discrete event. NS2 simulator provides a large number of built in C++ objective. It is advisable to use this C++ object for this simulator to setup a simulation using a tcl simulation script[44]. However, advantage users may find these objects insufficient[18]. They need to use otcl configuration interface and to develop their own C++

objects in order to put together these objects. Now this time NS2.35 is popularly used network simulator. This tool converts a .tcl files in to .tr and .nam files[45]. The network simulator can be described as a software or hardware that predicts the behavior of the network without the presence of the actual networks. After simulation NS2 outputs either animation based or text-based simulation results. Tools such as nam (network animator) and graph are used to interpret this result graphically and interactively. Then after in order to analyze a particular behavior of the networks, user can extract a relevant subset of text-based data and transform it to a more conceivable documentation or presentation[46]. The following figure illustrates the simulation model as follows.

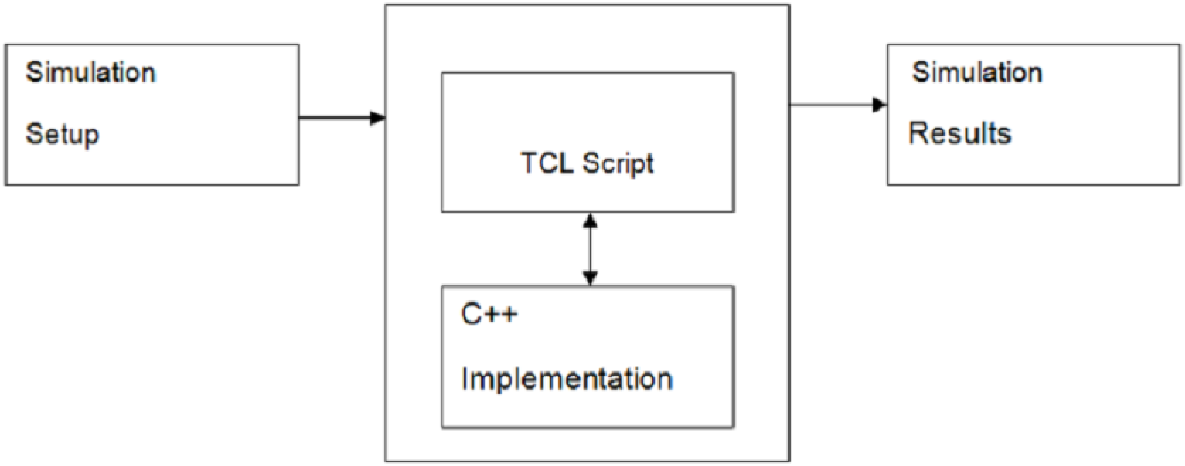


Fig 14: NS2 operation phase

5.3 Performance Evaluation

This section investigates the performance of our proposed approach detecting jellyfish attack and finding out the holding period on each node through calculating network delay and comparing the transmission delay. To analyze the performance of the proposed attacker detection scheme mechanism, the network simulator two is used.

5.3.1 Simulation Environment

For performance evaluation, nodes are randomly deployed in 100m x 100m area. Simulation is carried out using network simulator NS2. Each node is equipped with transceiver. Different node communicates via radio signal having transmission range of 100m. in our simulation, IEEE 802.11 is used as MAC layer protocol. The mobility of the node is determined by random waypoint mobility model. For constant bit rate data sessions, node pair is randomly selected and the packet size is used. The performance of improved AODV routing protocol is evaluated by using NS2.35 by illustrating different parameters. The following table illustrates the simulation parameter. After running C++ objects with Tcl simulation script, the result of simulation generated as trace files and the awk file is prepared for analyzing the trace file into graph for various performance metrics. The simulation experiment is carrying out using Linux. In this paper various parameter is simulated for number of nodes under simulation time 10 seconds. The following table gives the list of simulation parameters used for analysis of our proposed approach

Table 4: Simulation parameters

Parameters	Values
Operating system	Ubuntu 16.04
Simulator	NS-2.35-all-inone
Traffic type	CBR, TCP, UDP
Number of nodes	100
Attack	Jellyfish delay variant
Packet size	512kbps
Simulation time	10 second
Mobility model	Random waypoint model
Routing protocol	AODV
Antenna model	Omni
Radio propagation model	Two ray ground
Dimension	100m x 100m
Mobility Speed(variable)	(10,20,25,30) seconds
Channel type	Wireless channel
MAC type	802.11
Transport protocol	TCP
Pause time	0s

5.3.2 Performance Metrics

The performance of new improved protocol is evaluated using following metrics to compare the performance of the protocol with attacker node and without attacker node of AODV in MANET

i) Packet delivery ratio: this is the ratio of the data packets delivered to the destination node those generated by the source.

ii) Packet loss: is the difference between the number of data packets sent and the number of data packets received. it is calculated as follows

Packet loss = number of data packet sent – number of data packet received

iii) Throughput = the amount of data packet received by the destination per unit time.

In these studies, based on the proposed algorithm, parameter such as maximum average bandwidth are coded in NS2 and the code is validated by simulating and comparing the result obtained in the respective established work.

5.3.3 Simulation Result and Discussion

The performance of newly improved AODV routing protocol is evaluated in terms of variation in simulation time (i.e., transmission delay of each nodes in the included path)

5.3.4 Network Configuration

the simulation situation and parameters utilized for playing out the elaborated investigation is depicted underneath. This feature speaks to that how the power full execution parameters have been examined to simulate the conventions following advances have been utilized for simulation

- Inputs to simulator: scenario file having development of nodes, activity design document and simulation TCL record.
- Output file from simulator: trace document, network animator.
- Output from trace analyzer: xgr document

5.3.5 Performance Parameters

The analysis of routing protocol is done using two important performance metrics named as throughput and end to end delay.

5.3.6 Average End to End Delay

It is the normal time taken by an information packet to touch base at the goal. It incorporates all conceivable postponements caused by buffering amid course disclosure inactive, lining at the interface line, retransmission delays at the MAC and engendering exchange times.

D = $\sum Tr - Ts / \sum$no. of connection

Where Tr = received time

Ts = sent time

5.3.7 Throughput

It is the normal rate of effective message conveyance over a correspondence channel. It is likewise known as packet sent per unit interval of time. Throughput is generally estimated in bits every second or data bundles per time slot.

Throughput = Total packet received/ total time

This parameter is calculated and drown as graphs so that the performance can be compared. Ather performance parameters are also present to analyze the performance of wireless networks. Packet delivery ratio, normalized load and jitters are some parameters that define the credibility of the network. In this section end to end delay and throughput is calculated for AODV routing convention using jellyfish assaults. The following table and diagram respectively show the evaluation of end-to-end delay for AODV.

Table 5: comparison of End-to-end delay

End to End delay comparison					
No. nodes	ABC	MABC	AR AIDF JFRS	APD JFAD	JFA DP
20	20.56	15.36	11.25	9.56	9.5
40	21.43	16.98	12.58	9.15	8.71
60	23.58	17.87	13.17	8.2	7.65
80	24.15	18.46	13.95	7.65	5.01
100	25.63	20.51	14.47	7.1	4.78
Avg	23.07	17.836	13.084	8.332	7.13

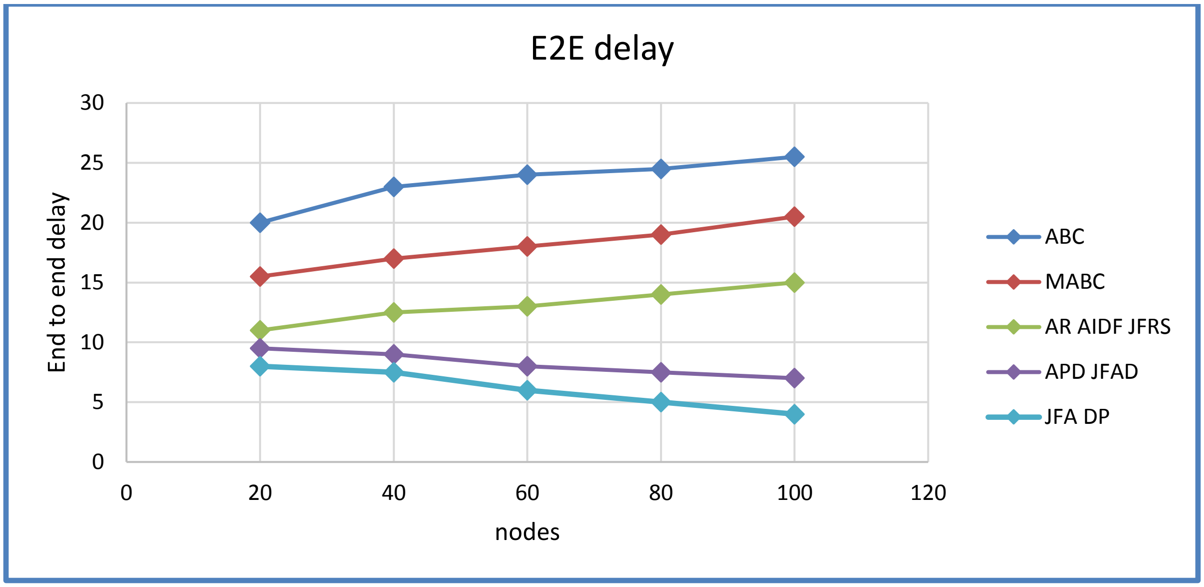


Fig 15: AODV end to end delay

Average end to end delay for a network under AODV routing protocol is taken to understand the effect of jellyfish attack. Three scenarios are taken with vary number of nodes as 20,40,60,100. Firstly, simple AODV protocol is implemented without any attack. It is noted that there is no delay in transmission. In second case behavior of AODV is seen under varying jellyfish attacker nodes. As the count of attacker hubs present end to end delay keeps on increasing. Afterwards the attacker identification and removal algorithm are applied to the network and result are shown. It is seen that the performance in account of end-to-end delay improves substantially especially the existence of attacker nodes.

5.3.8 Throughput for AODV

Throughput under AODV routing convention is analyzed under an impact of jellyfish attack. Three scenarios are used with attackers in the network nodes as 20,40,60 and 100. In this scenario since there is attacker node, AODV gives minimum throughput as attacker node is present in this case. Afterwards throughput increase significantly with the detection and prevention of attacker nodes, when the protocol got affected by jellyfish attacker node. Further after the implementation of the attacker detection and prevention algorithm, significant in increased throughput is observed. The following table and figure respectively represent the behavior of AODV routing convention with the presence of attack node.

Table 6: comparison of throughput

Throughput (Mbps) comparison					
No. nodes	ABC	MABC	AR AIDF JFRS	APD JFAD	JFA DP
20	178	205	236	245	250
40	278	312	356	378	410
60	389	423	569	591	620
80	425	476	587	599	611
100	483	561	621	645	655
Avg	350.6	395.4	473.8	491.6	509.2

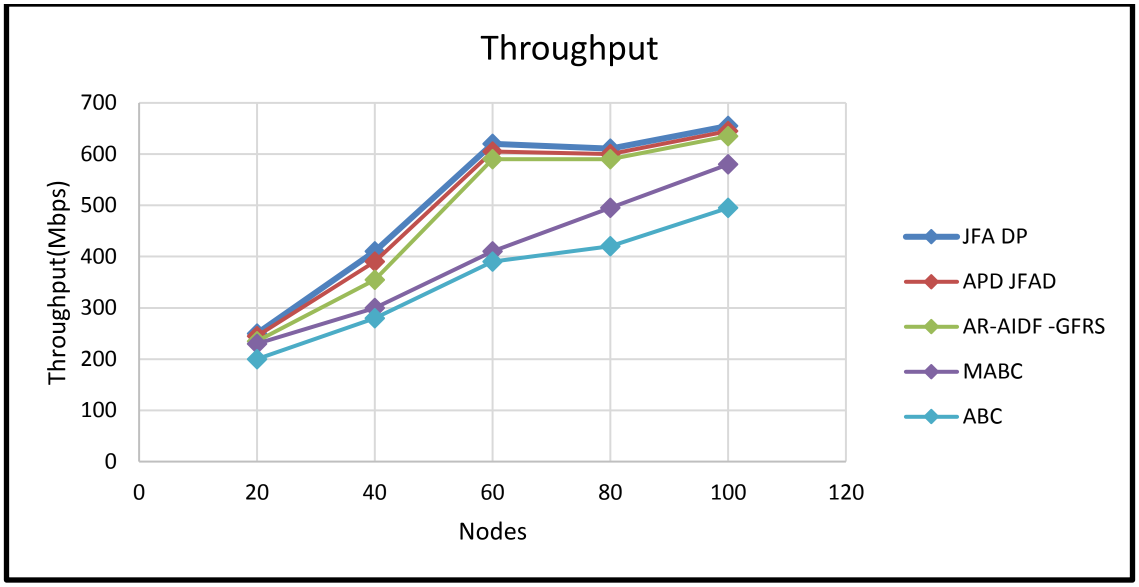


Fig 16: AODV throughput

5.3.9 Packet Delivery Ratio

In AODV the packet delivery ratio can be characterized as the proportion that are utilized to ascertain the volume of packets transmitted by source node to the volume of packets got by destination node. Packet delivery ratio is calculated as

Packet delivery ratio = $\sum$ the number of packet received / $\sum$ the number of packets directed

Table 7: comparison of packet delivery ratio

Packet delivery ratio (PDR %) comparison					
No. nodes	ABC	MABC	AR AIDF JFRS	APD JFAD	JFA DP
20	75.89	82.56	89.52	92	93
40	76.93	83.87	90.51	93.5	95.5
60	77.52	84.79	91.45	94.6	96.1
80	77.86	85.31	91.87	95.1	97.2
100	78.52	86.52	92.53	96.8	97.9
Avg	77.344	84.61	91.176	94.4	95.94

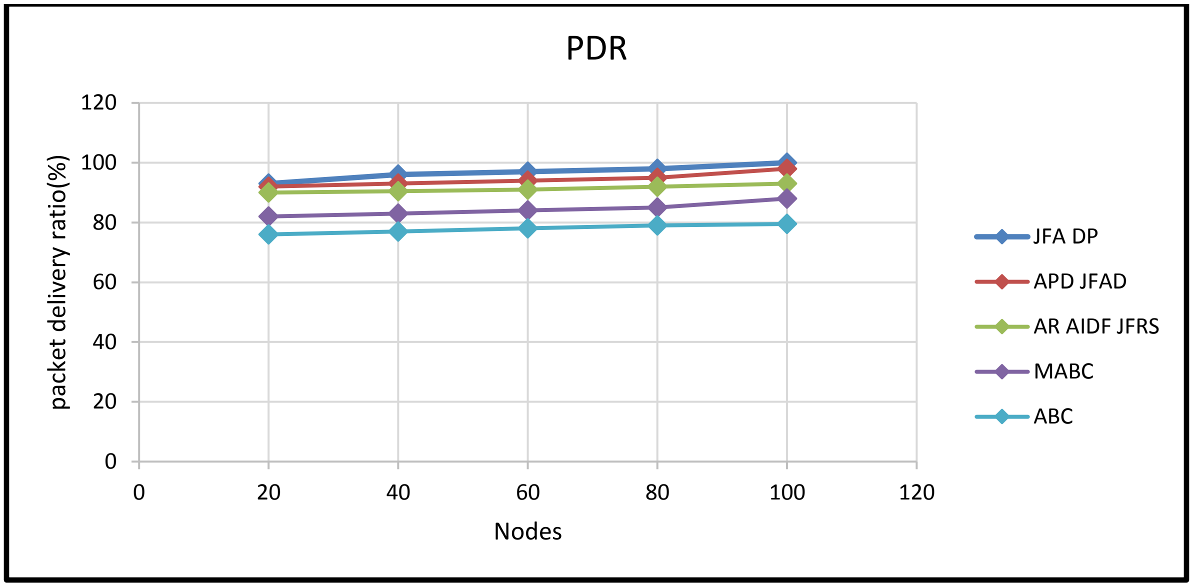


Fig 17: AODV packet delivery ratio

CHAPTER SIX

6. CONCLUSION AND RECOMMENDATION

6.1 Overview

In this chapter we are going to summarize what we have done in these studies and draw conclusion based on the result presented in previous chapter. Also in this chapter, we try to give a recommendation on the application scenario in which our protocol can work well. Finally, we suggest to which direction our work could be extended under future work section. In this section we have propose quality of service (QoS) extension of the AODV protocol for mobile ad hoc networks based on available bandwidth and delay in transmission, delay in propagation, delay in queue and delay in processing and network congestion occurred due to delay among these. Based on the proposal we have designed our new protocol to gather information about quality-of-service metrics our proposal uses hello messages available bandwidth, delay and jitter on each link gathered.

Our proposed protocol was implemented and simulated in network simulator (ns2) version 2.35 one of the network simulator used by various researchers. Our protocol showed better results in terms of throughput, end to end delay and packet delivery ration. After the detected node is prevented and normal node only left in the entire path.

Conclusion

Mobile ad hoc network as we described before, have unpredictable link. Providing quality of service in such networks is too much difficult. Even if quality of service provision in MANET is difficult due to their potential application many researchers give their attention to the matter. As we have described in the previous chapter MANET have mostly applicable in the disaster area temporary laboratory, in a police department, in firefighting and so on. Generally, they are the only means of communication when there is damage on the other networks. They are characterized by dynamic configuration, link instability, and high mobility, frequent topology change, decentralized and trusting all intermediate nodes by default. So, providing quality of service is challenging because of these characteristics. Attacker mostly succeeded in MANET since all intermediate nodes is trusted so simply attacker works as an intermediate node and results the network congestion, this congestion produces high amounts of delay in the networks, this causes decreasing the quality of service by decreasing the throughput. To solve such problem, we compute the end-to-end delay of the network and check if there is congestion of networks and if congestion is detected, we propose to analyze the transmission delay of each node in the path and valance the result, at the end identifying the node having holding period and pause the link with that node and establishing the new path with ordinary node having no holding period. For evaluating our work, we use the NS2 version 2.35. Our result shows the path with jellyfish attacker node is high transmission delay and the path without jellyfish attacker node has balanced delay which results of high quality of service. Generally, in this study we investigated that jellyfish attacker node considerably affect the performance for AODV routing conventions, Algorithms proposed for detection and removal of jellyfish attacker node can significantly improve the performance of routing protocol, Throughput of AODV routing convention is more able to sustain itself with presences of jellyfish attacker detection and prevention algorithm, Further after the detection and removal of attacker node AODV routing protocol shows end to end delay has trust value in AODV routings. In improving packet transmission protocol by detecting and preventing jellyfish attack node in MANET is successful, this way the node gives trust value to its neighbors, so that the node configuration for the network from the jellyfish attack can be protected.

Recommendation

In this paper we have investigated the performance of packet transmission by investigating the novel method to detect and prevent jellyfish attack in MANET which is important for providing quality of service, then we have used available bandwidth, delay variant and jitter as metrics to enhance the AODV protocol. We have evaluated our protocol performance and compared them with the original one. From our results we generally recommend the application area where the enhanced protocol can be used as follows.

- Our new enhanced protocol method can be used to identify the attacker node in the path, during this time the calculated result after transmission of packets compared with the origin result, since we are using the size of the packet and the available bandwidth, the output is more trust. We recommended that it is most trusted if transmission delay value is supported by machine learning.
- It is worthy that checking all nodes delay by scheming their normal transmission delay, there for it is better if any other method is more investigated.
- This study is limited to number of nodes during evaluation, it is better to check for multiple number of nodes.

Generally, this investigation can be extended by studying the other two jellyfish assaults namely jellyfish periodic dropping assault and jellyfish reorder assault with the help of machine learning.

References

[1] M. Al Mojamed, “Integrating Mobile Ad Hoc Networks with the Internet Based on OLSR,” Wirel. Commun. Mob. Comput., vol. 2020, 2020, doi: 10.1155/2020/8810761.

[2] E. O. Ochola, L. F. Mejaele, M. M. Eloff, and J. A. Van Der Poll, “manet reactive routing protocols node mobility variation effect in analysing the impact of black hole attack,” 2017.

[3] St. Peter’s College of Engineering and Technology, Institute of Electrical and Electronics Engineers, Council of Scientific & Industrial Research (India), and Indian Council of Medical Research, Proceedings of 2017 Third IEEE International Conference on Sensing, Signal Processing and Security (ICSSS 2017) : May 4th and 5th, 2017 : St. Peter’s College of Engineering and Technology, Avadi, Chennai, Tamil, Nadu, India-600054. .

[4] SCAD Institute of Technology, IEEE Electron Devices Society, and Institute of Electrical and Electronics Engineers, Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud (I-SMAC 2017) : 10-11, February 2017. .

[5] S. Hijazi, M. Moshref, and S. Al-Sharaeh, “Enhanced AODV Protocol for Detection and Prevention of Blackhole Attack in Mobile Ad Hoc Network SUBJECT CLASSIFICATION TYPE (METHOD/APPROACH),” 2017.

[6] K. Naveeda and V. N. Poorani, “Defending Against Intrusion and Prevention of Jellyfish Attack Approach for Detecting Malicious Node in MANET,” Int. J. Adv. Res. Electr. Electron. Instrum. Eng. (An ISO, vol. 3297, 2007, doi: 10.15662/IJAREEIE.2016.0506054.

[7] M. Ankit, M. Vaghela, P. M. Gour, P. A. Patel, and E. Department, “Survey on Delay Based Jellyfish Attack,” 2017. [Online]. Available: www.ijedr.org.

[8] V. Laxmi, C. Lal, M. S. Gaur, and D. Mehta, “JellyFish attack: Analysis, detection and countermeasure in TCP-based MANET,” J. Inf. Secur. Appl., vol. 22, pp. 99–112, Jun. 2015, doi: 10.1016/j.jisa.2014.09.003.

[9] A. Thomas, V. K. Sharma, and G. Singhal, “Secure Link Establishment Method to Prevent Jelly Fish Attack in MANET,” in Proceedings - 2015 International Conference on

Computational Intelligence and Communication Networks, CICN 2015, Aug. 2016, pp. 1153–1158, doi: 10.1109/CICN.2015.224.

[10] S. Doss et al., “APD-JFAD: Accurate prevention and detection of jelly fish attack in MANET,” IEEE Access, vol. 6, pp. 56954–56965, Sep. 2018, doi: 10.1109/ACCESS.2018.2868544.

[11] V. Hnatyshin, R. Hussey, E. Huff, and Z. Shinwari, “A Comparative Study of Proactive and Reactive Geographical Routing Protocols for MANET,” 2018. [Online]. Available: https://www.researchgate.net/publication/326065983.

[12] M. Yu, M. Zhou, and W. Su, “A secure routing protocol against byzantine attacks for MANETs in adversarial environments,” IEEE Trans. Veh. Technol., vol. 58, no. 1, pp. 449–460, 2009, doi: 10.1109/TVT.2008.923683.

[13] M. Shrivastava, “iaodv: an improved aodv routing protocol for MANET,” Int. J. Adv. Res. Comput. Sci., vol. 9, no. 2, pp. 167–174, Apr. 2018, doi: 10.26483/ijarcs.v9i2.5483.

[14] SVS College of Engineering, IEEE-USA, Institute of Electrical and Electronics Engineers. Madras Section, and Institute of Electrical and Electronics Engineers, Proceedings of 2015 IEEE International Conference on Electrical, Computer and Communication Technologies. .

[15] H. Ghayvat, S. Pandya, S. Shah, S. C. Mukhopadhyay, M. H. Yap, and K. H. Wandra, Advanced AODV Approach For Efficient Detection And Mitigation Of WORMHOLE Attack IN MANET. .

[16] D. Milkiyas, “Optimal Path Selection for AOMDV Routing Protocol with Efficient Energy and Bandwidth in Mobile Ad hoc Network.”

[17] Y. Liu, Y. Cui, and X. Wang, “Connectivity and Transmission Delay in Large-scale Cognitive Radio Ad Hoc Networks With Unreliable Secondary Links.”

[18] B. Renu, M. Hardwari, and T. Pranavi, “Routing Protocols in Mobile Ad-Hoc Network: A Review,” 2013.

[19] P. Belimpasakis -Nokia et al., “About ENISA Legal notice Thanks We would like to thank the following members of the ENISA’s expert group on Priorities of Research on Current & Emerging Network Technologies (PROCENT) for their input and advice: Ioannis Askoxylakis-FORTH ICS Vasilios Siris-FORTH ICS.”

[20] 2017 International Symposium on Wireless Systems and Networks (ISWSN). IEEE, 2017.

[21] V. Kanakaris, D. Ndzi, and K. Ovaliadis, “improving aodv performance using dynamic density driven route request forwarding.” [Online]. Available: www.port.ac.uk.

[22] “ENSC 427 Project Report.” [Online]. Available: http://dsn.tm.uni-karlsruhe.de/english/ns3-physim.php.

[23] D. E. Comer et al., International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014 24-27 Sept. 2014, Delhi, India ; [including symposia and workshops]. .

[24] S. Khanvilkar and A. Khokhar, “Transmission Delay Multimedia Networks and Communication.”

[25] A. R. Rajeswari, “A Mobile Ad Hoc Network Routing Protocols: A Comparative Study.” [Online]. Available: www.intechopen.com.

[26] “Performance Evaluation of MANET Routing Protocol under Black Hole Attack Using OPNET Simulator.”

[27] pardeep singh tiwana, nafiza mann"jellyfish reorder attack on hybrid protocol in manet dissection on variegated parameter"2016.

[28] G. Birhanu, “school of computing investigating multiple qos metrics for the improvement of aodv protocol in manet,” 2016.

[29] Dr. B.C. Roy Engineering College. Department of Computer Science and Dr. B.C. Roy Engineering College. Department of Information Technology, 2012 National Conference on Computing and Communication Systems (NCCCS) : 21-22 November 2012 : Proceeding. .

[30] S. Satheeshkumar and N. Sengottaiyan, “Defending against jellyfish attacks using cluster based routing protocol for secured data transmission in MANET,” Cluster Comput., vol. 22, pp. 10849–10860, Sep. 2019, doi: 10.1007/s10586-017-1202-z.

[31] S. Raju and D. A. Parikh, “Performance Improvement in VANET by Modifying AODV Routing Protocol,” 2015. [Online]. Available: www.iiste.org.

[32] A. Ahmed, A. Hanan, and I. Osman, “Description of Black Hole Attack Behaviour in MANET,” 2016. [Online]. Available: www.ijcncs.org.

[33] “A Soft Computing Approach to Analyse Aodv Routing Protocol.”

[34] Z. Fan et al., “A Time-Delay-Bounded Data Scheduling Algorithm for Delay Reduction in Distributed Networked Control Systems,” Math. Probl. Eng., vol. 2020, 2020, doi: 10.1155/2020/8290879.

[35] M. Sharma and P. Tripathi, “IJR) e-ISSN: 2348-6848,” 2015.

[36] A. Garg, S. Kumar, and K. Dutta, “An Analytical Survey of State of the Art JellyFish Attack Detection and Prevention Techniques.”

[37] S. Sachdeva and P. Kaur, “Detection and Analysis of Jellyfish Attack in MANETs.”

[38] M. Ebna and M. M. J. Babu, “End-to-End Delay Performance Evaluation for VoIP in the LTE network,” 2011.

[39] “Network Delay Network Delay Network Management & Monitoring.”

[40] IEEE Electron Devices Society, Institute of Electrical and Electronics Engineers, and Vaigai College of Engineering, Proceeding of the 2018 International Conference on Intelligent Computing and Control Systems (ICICCS) : June 14-15, 2018. .

[41] N. K. Raju and N. K. Kumar, “performance evaluation of aodv in different environments,” 2010.

[42] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, “An Efficient Decentralized Key Management Mechanism for VANET with Blockchain,” IEEE Trans. Veh. Technol., vol.

69, no. 6, pp. 5836–5849, Jun. 2020, doi: 10.1109/TVT.2020.2972923.

[43] Gv. Sai and Il. Babu, “NS2 DOS Defence for Resource Availability in Wireless Sensor Networks,” Int. J. Adv. Sci. Technol., vol. 29, no. 7, pp. 3471–3478, 2020.

[44] “The ns Manual (formerly ns Notes and Documentation) 1.” [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.

[45] Institute of Electrical and Electronics Engineers, IEEE Computer Society, and VUNOTIC (Organization), 2017 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT). .

[46] D. Rathi and R. Welekar, “Performance Evaluation of AODV Routing Protocol in VANET with NS2,” Int. J. Interact. Multimed. Artif. Intell., vol. 4, no. 3, p. 23, 2017, doi: 10.9781/ijimai.2017.434.

Appendix

Sample source code and screen shoot for proposed work taken as follows

Appendix 1 network animator (NAM) and trace file

The following image shows the NAM screen shoots for the NS2 simulations at different point of time of the simulations. The below image shows the node at the start of our simulation.

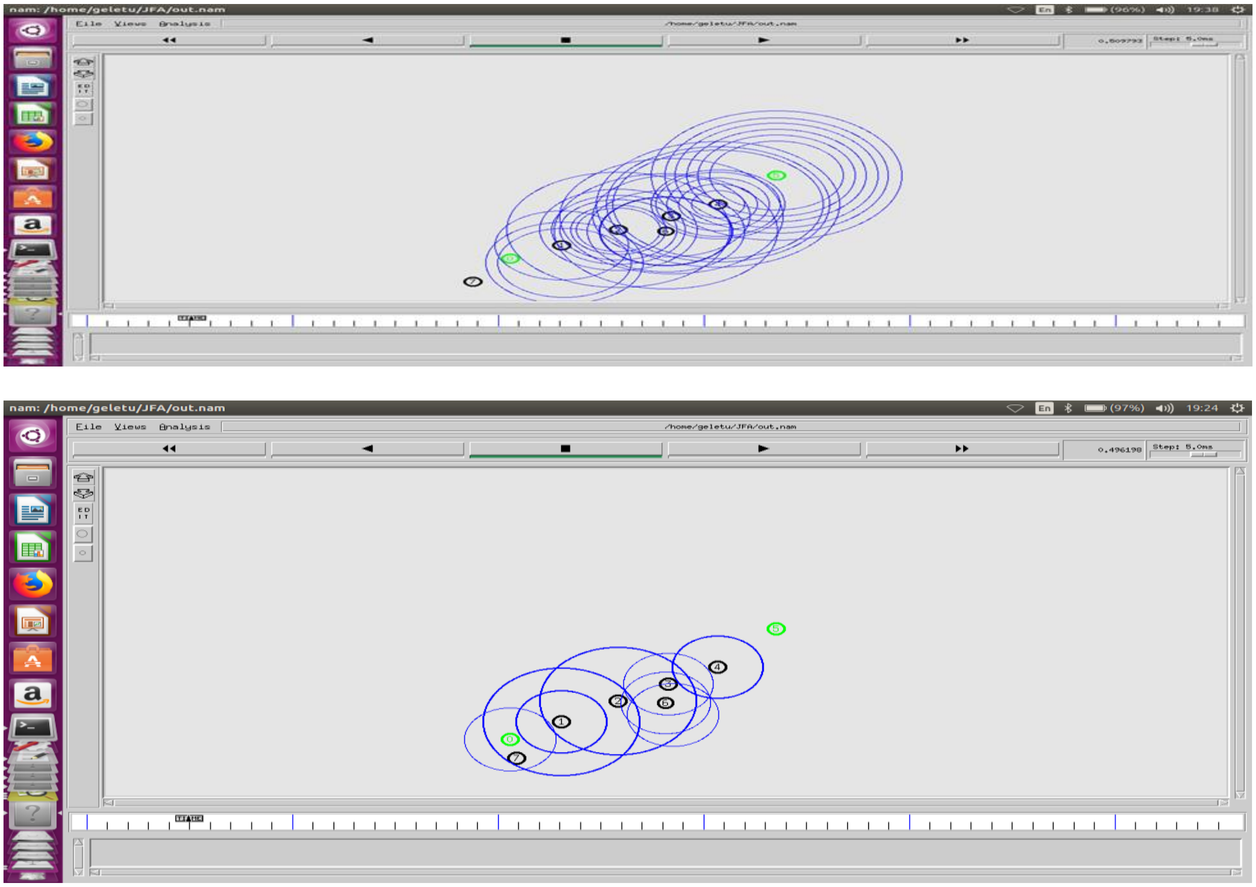


Fig 18: Node at start of simulation

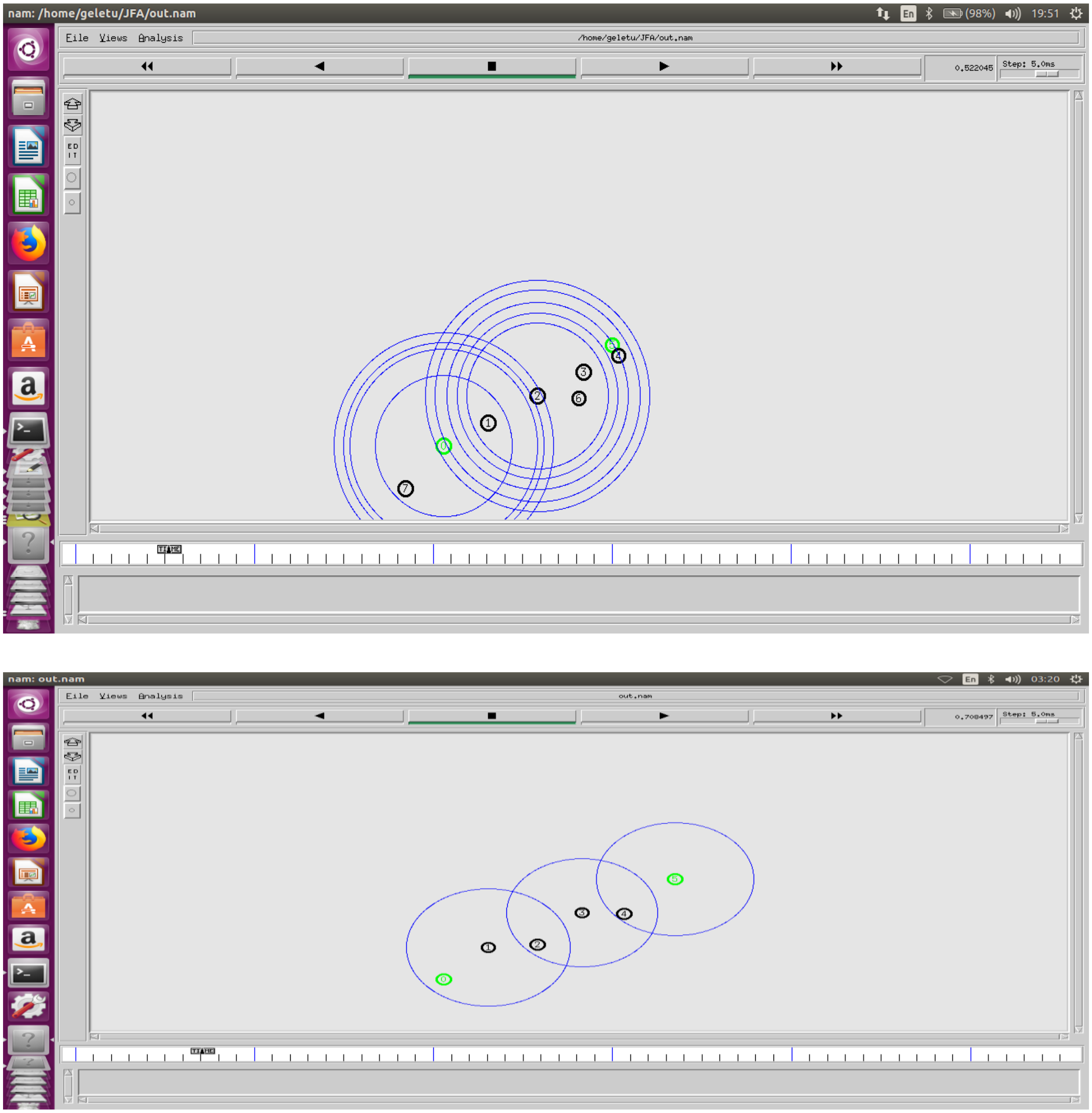


Fig 19: source and destination node identified

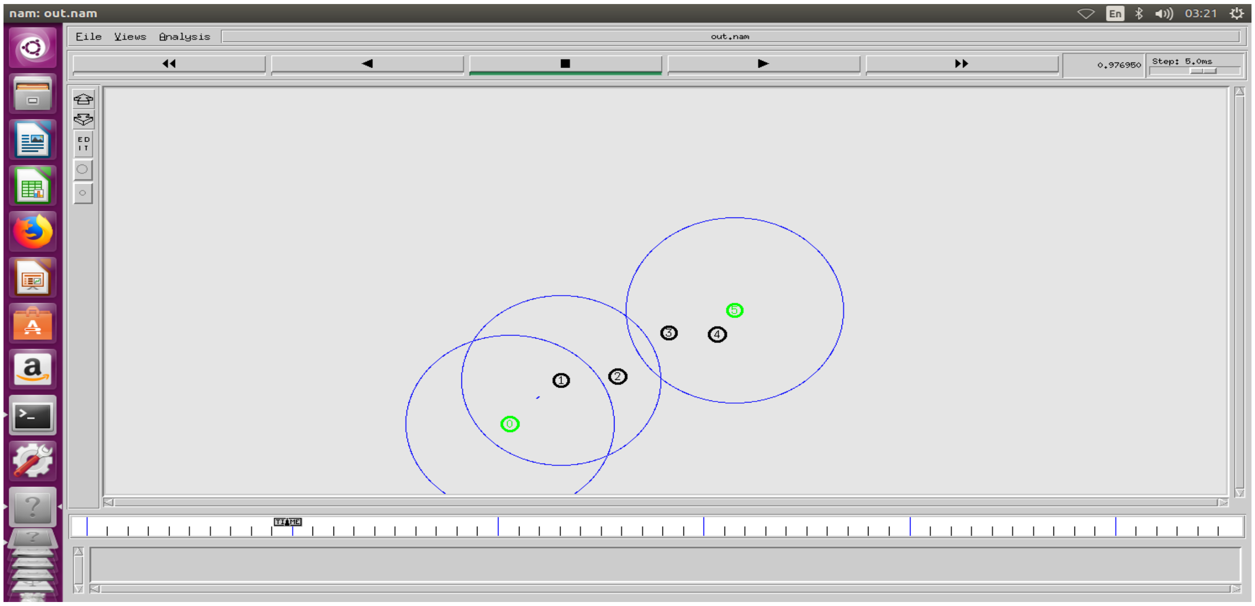
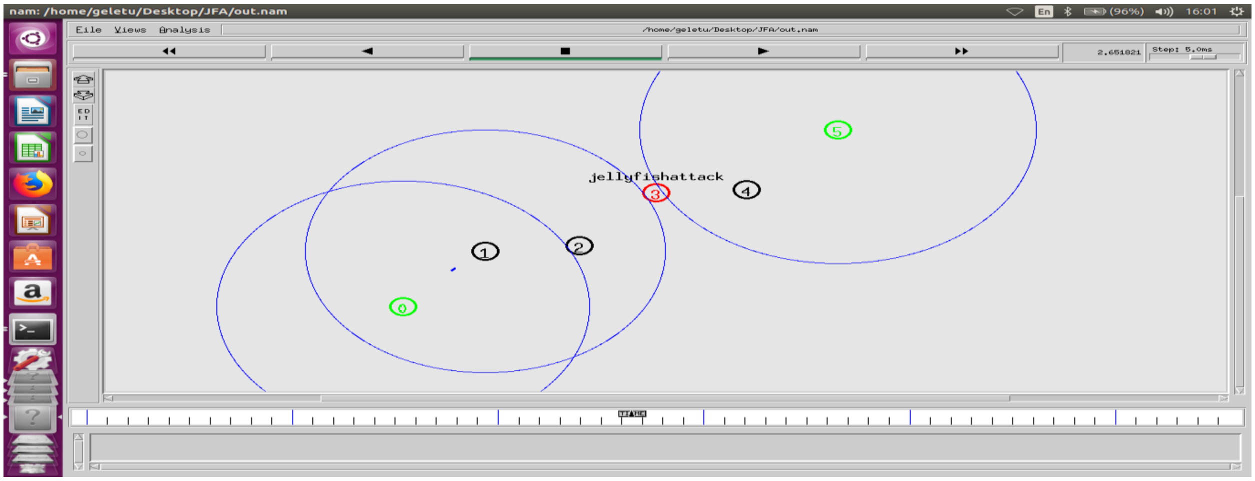
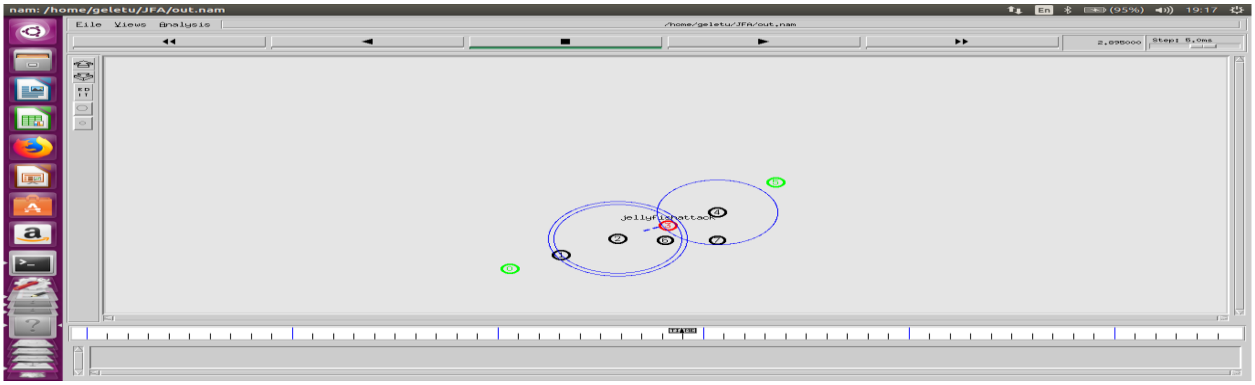


Fig 20: transmission of packet is started


nam: /home/geletu/Desktop/JFA/out.nam
(96%)
16:01
File Views Analysis
/home/geletu/Desktop/JFA/out.nam
2.651821
Step: 5.0ms
jellyfishattack
0
1
2
3
4
5



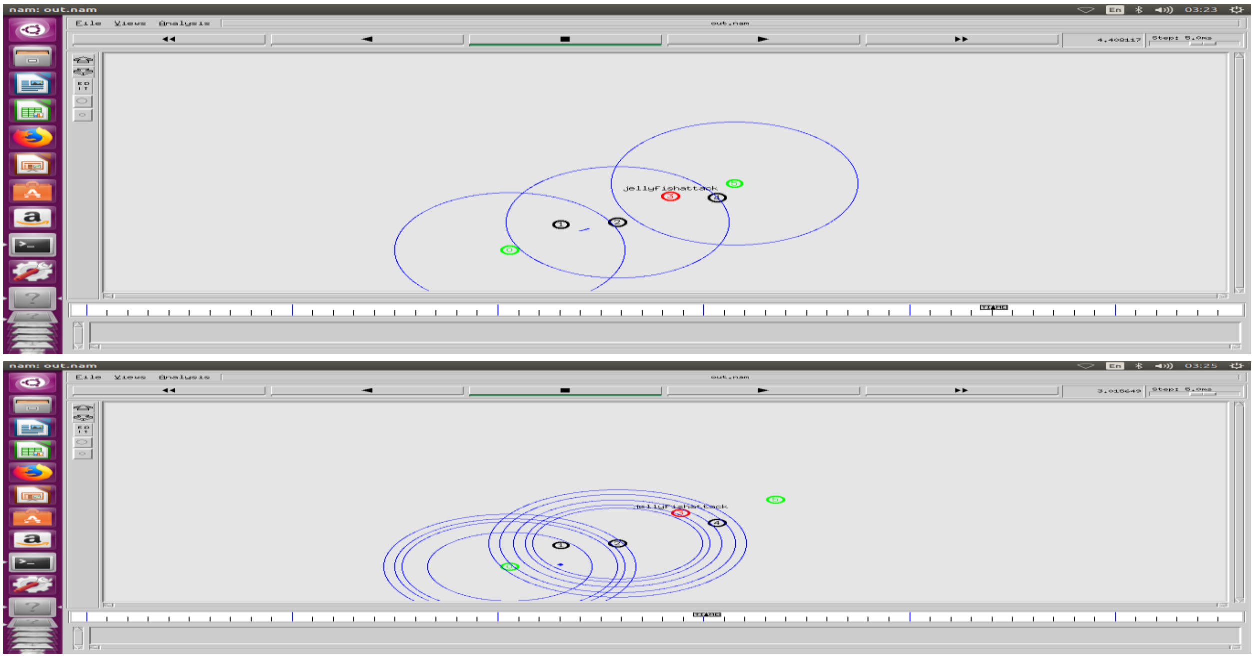


Fig 21: jellyfish attacks are detected

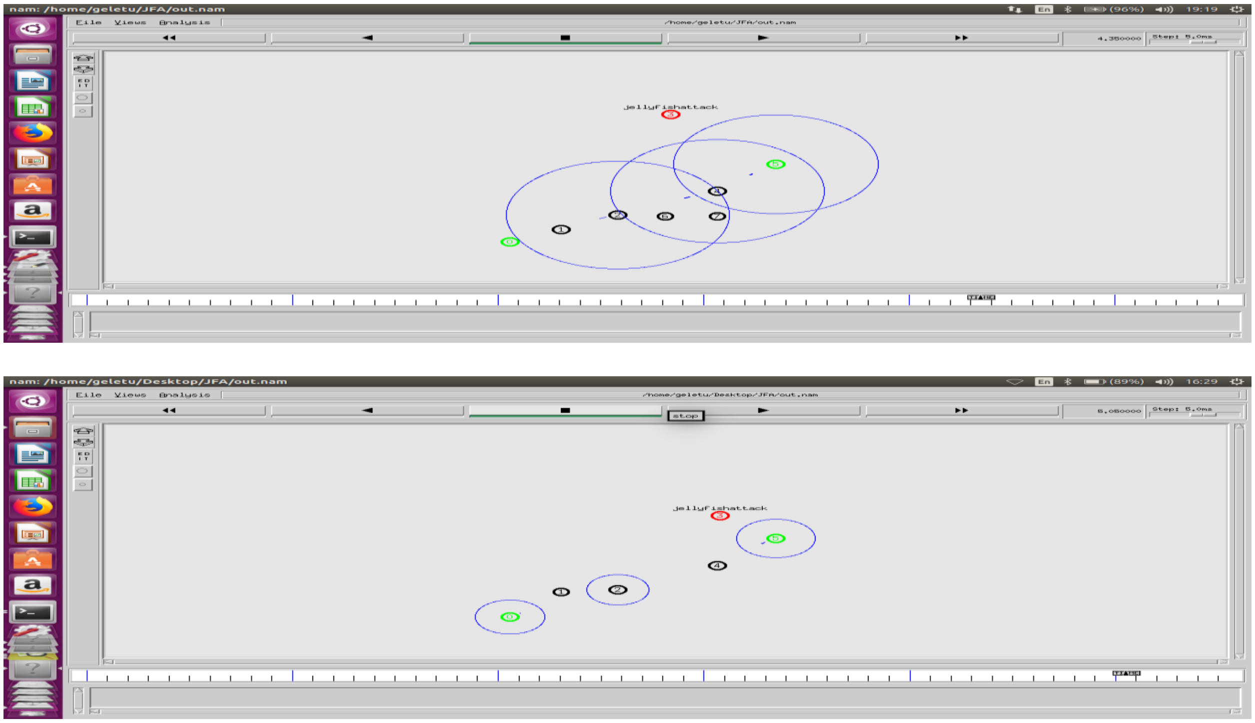

nam: /home/geletu/JFA/out.nam
File Views Analysis
jellyFishattack
nam: /home/geletu/Desktop/JFA/out.nam
stop
jellyFishattack

Fig 22: jellyfish attack is removed

Appendix 2 code snippets

The bellow screen shoot shows some of the code snippets in ns2

```
#ns2 simulation
#written by geletu hussein
#===================================================================
# Define options
#===================================================================
set val(chan)         Channel/WirelessChannel  ;# channel type
set val(prop)         Propagation/TwoRayGround ;# radio-propagation model
set val(ant)          Antenna/OmniAntenna      ;# Antenna type
set val(ll)           LL                       ;# Link layer type
set val(ifq)          Queue/DropTail/PriQueue  ;# Interface queue type
set val(ifqlen)       50                       ;# max packet in ifq
set val(netif)        Phy/WirelessPhy          ;# network interface type
set val(mac)          Mac/802_11               ;# MAC type
set val(nn)           6                        ;# number of mobilenodes
set val(rp)           AODV                     ;# routing protocol
set val(x)
set val(y)
set ns [new Simulator]
#ns-random 0
set f [open out.tr w]
$ns trace-all $f
set namtrace [open out.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
set topo [new Topography]
$topo load_flatgrid 800 800
create-god $val(nn)
set chan_1 [new $val(chan)]
set chan_2 [new $val(chan)]
set chan_3 [new $val(chan)]
set chan_4 [new $val(chan)]
set chan_5 [new $val(chan)]
set chan_6 [new $val(chan)]
# CONFIGURE AND CREATE NODES
$ns node-config  -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
#-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
-movementTrace OFF \
-channel $chan_1
proc finish {} {
global ns namtrace
$ns flush-trace
close $namtrace
exec nam -r 5m out.nam &
exit 0}
# define color index
$ns color 0 blue
$ns color 1 red
$ns color 2 chocolate
```

```
$ns color 3 red
$ns color 4 brown
$ns color 5 tan
$ns color 6 gold
$ns color 7 black
set n(0) [$ns node]
$ns at 0.0 "$n(0) color green"
$n(0) color "0"
$n(0) shape "circle"
set n(1) [$ns node]
$ns at 0.0 "$n(1) color black"
#$ns at 2.0 "$n(1) color red"
$n(1) color "blue"
$n(1) shape "circle"
set n(2) [$ns node]
$n(2) color "tan"
$n(2) shape "circle"
set n(3) [$ns node]
$n(3) color "red"
$n(3) shape "circle"
$ns at 0.0 "$n(3) color black"
$ns at 2.0 "$n(3) color red"
$ns at 2.0 "$n(3) label jellyfishattack"
set n(4) [$ns node]
$n(4) color "tan"
$n(4) shape "circle"
set n(5) [$ns node]
$ns at 0.0 "$n(5) color green"
$n(5) color "red"
$n(5) shape "circle"
```

```
for {set i 0} {$i < $val(nn)} {incr i} {
$ns initial_node_pos $n($i) 30+i*100}
#$ns at 2.0 "[$n(1) set ragent_] malicious"
#$ns at 4.0 "[$n(1) set ragent_] non-malicious"
$ns at 0.0 "$n(0) setdest 100.0 100.0 3000.0"
$ns at 0.0 "$n(1) setdest 200.0 200.0 3000.0"
$ns at 0.0 "$n(2) setdest 300.0 200.0 3000.0"
$ns at 0.0 "$n(3) setdest 400.0 300.0 3000.0"
$ns at 0.0 "$n(4) setdest 500.0 300.0 3000.0"
$ns at 0.0 "$n(5) setdest 600.0 400.0 3000.0"
$ns at 2.9 "$n(3) setdest 500.0 500.0 3000.0"
# CONFIGURE AND SET UP A FLOW
set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]
set sink2 [new Agent/LossMonitor]
set sink3 [new Agent/LossMonitor]
set sink4 [new Agent/LossMonitor]
set sink5 [new Agent/LossMonitor]
$ns attach-agent $n(0) $sink0
$ns attach-agent $n(1) $sink1
$ns attach-agent $n(2) $sink2
$ns attach-agent $n(3) $sink3
$ns attach-agent $n(4) $sink4
$ns attach-agent $n(5) $sink5
#$ns attach-agent $sink2 $sink3
set tcp0 [new Agent/TCP]
$ns attach-agent $n(0) $tcp0
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1
set tcp2 [new Agent/TCP]
```

```
$ns attach-agent $n(2) $tcp2
set tcp3 [new Agent/TCP]
$ns attach-agent $n(3) $tcp3
set tcp4 [new Agent/TCP]
$ns attach-agent $n(4) $tcp4
set tcp5 [new Agent/TCP]
$ns attach-agent $n(5) $tcp5
proc attach-CBR-traffic { node sink size interval } {
#Get an instance of the simulator
set ns [Simulator instance]
#Create a CBR  agent and attach it to the node
set cbr [new Agent/CBR]k
$ns attach-agent $node $cbr
$cbr set packetSize_ $size
$cbr set interval_ $interval
#Attach CBR source to sink;
$ns connect $cbr $sink
return $cbr
}
set cbr0 [attach-CBR-traffic $n(0) $sink5 1000 .030]
$ns at 0.5 "$cbr0 start"
set cbr12 [attach-CBR-traffic $n(0) $sink5 1000 0.30]
$ns at 3.0 "$cbr12 start"
$ns at 5.5 "finish"
puts "Start of simulation."
$ns run
```