PACKET DROP ATTACK DETECTION AND CLASSIFICATION IN MANET USING A NEURAL NETWORK BASED INTRUSION DETECTION SYSTEM

A Dissertation submitted in partial fulfilment of the requirement for the award of degree of

Master of Technology

In

Computer Science & Engineering

Submitted by

INNOCENT MAPANGA

(Roll No. - 2K12/CSE/32)

Under the guidance of

Mr. VINOD KUMAR

Associate Professor, Computer Engineering Department, DTU

DEPARTMENT OF COMPUTER ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
BAWANA ROAD, DELHI
2012-2014

ABSTRACT

In wireless ad hoc networks, communication for end-to-end delivery of packets is achieved cooperatively. The cooperative model ensures that several nodes forming the network coordinate whenever communication is to take place between a sending node and a desired recipient node, which fall out of the sending node’s communication range. This model assumes that an intermediary node will always forward traffic originating from other nodes willingly, other than traffic emanating from the node itself. Conversely, in hostile environments where we find most applications of our ad hoc networks, an always cooperative and submissive behavior on behalf of the other nodes of the network cannot be presumed as the ultimate action undertaken by all the nodes. Misbehaving nodes, which are part of the network, may refuse to pass on traffic to other nodes, for a different many reasons, including to preserve energy or to deliberately degrade performance of the network.

Our focus in this thesis is on detecting the presence of malicious nodes that selectively or randomly drop packets intended for other destination nodes, we further classify each packet drop attack, according to its attack type by observing and analyzing how each packet drop attack affect the network characteristics. To effectively detect and classify the misbehaving nodes in MANET, we have developed a system based on the intelligent use of artificial neural networks that makes use of local data collected at each node. Our system has a number of components that work together to achieve the desired objective. The three components are (i) data collection component (ii) data analysis component (iii) detection and classification component. Our three components are integrated together so as to ensure that all malicious nodes present in the network can be detected at high rates with a very low false positive rate. Our technique fares well in comparison to previously proposed methods, with our data analysis component extracting useful metrics and parameters required as input for training our detection and classification engine, which then monitors and evaluates the behavior of each node on the basis of each packet. Using a simulated MANET environment and ANNs modelling we can illustrate that our technique can successfully detect malicious packet droppers as well as classify the several packet dropping attacks at work on the misbehaving nodes.

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY

(Govt. of National Capital Territory of Delhi)

BAWANA ROAD, DELHI – 110042

This is to certify that dissertation entitled “***Packet Drop Detection and Classification in MANET using a Neural Network Based Intrusion Detection System***” has been completed by **INNOCENT MAPANGA** (Roll Number: **2K12/CSE/32**) for partial fulfilment of the requirements for the award of **Master of Technology** degree in **Computer Science & Engineering**. This work is carried out by him under my supervision and has not been submitted earlier for the award of any other degree or diploma in any university to the best of my knowledge.

Mr Vinod Kumar
Associate Professor
Project Guide
Department of Computer Engineering
DTU, Delhi

ACKNOWLEDGEMENTS

This dissertation is the result of a study in "***Packet Drop Detection and Classification in MANET using a Neural Network Based Intrusion Detection System*** " which was conducted in the fourth semester at the Department of Computer Engineering in Delhi Technological University (formerly DCE), Delhi.

My greatest gratitude goes to my advisor Associate Prof Vinod Kumar, for his consistent support and guidance. His knowledgeable and profoundly inspiring discussions guided me throughout my entire Masters studies.

Last but definitely not least, I wish to thank all of my friends and fellow students who made my life at Delhi Technological University cheerful and memorable. I thank you for all your ideas and assistance you rendered to me. Your friendship was most invaluable to me.

INNOCENT MAPANGA
Roll Number 2K12/CSE/32
M. Tech Computer Science & Engineering
Dept. of Computer Engineering
DTU, Delhi

Table of Contents

List of Figures

List of Tables

Introduction

1.1 Background

Mobile Ad hoc Network (MANET) is a collection of nodes which are mobile and self-organize themselves into a network, with no fixed topology. As such nodes can freely roam around, join or leave the network randomly. MANETs can be established devoid of any infrastructure hence are becoming very useful, especially in environments that are geographically constrained for instance in next generation of battlefield applications envisioned by the military as well as in applications like disaster recovery and message exchanges in rescue missions. Also, each node can function as a router, utilising its multi hop routing facility. This eliminates the need for a dedicated router or access point for communication between nodes. However, the MANET is vulnerable especially due to its continuously changing topology, open medium, lack of central monitoring point as well as no clear defensive mechanism. For instance any un-trusted node is capable of joining the network, subsequently posing threats to it. This can be done either by dropping the packets or by providing wrong information to the network and so on.

Ensuring and enforcing security in MANET is of prime importance since data in transit may be confidential and delivery of packets must be ensured by the network. Attacks targeted at MANETs can emanate either from within or outside the network and most times attacks come from trusted nodes within the network. The Wireless medium is susceptible to attacks, due to ease of access into the network [87]. The costs of damages in the event of an attack as a result of malicious activities in the network can have unbearable consequences hence a need for systems that can monitor data flow within the network in order to outwit possible malicious activities. Such a system for monitoring the network is called an intrusion detection system (IDS). An IDS collects and evaluates information from different areas within a node or network to discover suspicious patterns that may signify an attack or attempt to compromise a system. IDS design are based on two approaches namely, anomaly detection and misuse based IDS. Misuse-based IDS looks for behavior corresponding to predefined intrusion or vulnerability signatures. Anomaly detection based IDS searches for abnormal network traffic, which can either be a violation of acceptable thresholds for an occurring event or a violation of a user's normal behavior in the network.

1.2 MANET Packet Drop Attacks

The substance of this research is an organized analysis of MANET Packet Drop Attacks. We use an attack detection and classification based on an artificial intelligent technique applying the capabilities of artificial neural networks (ANN). In this light, we propose anomaly detection and misuse methods fit for identifying various attacks, both known and unknown. We base our approach on AODV routing protocol using both statistical and categorical measures. The data gathered within each node is used as input to our approach.

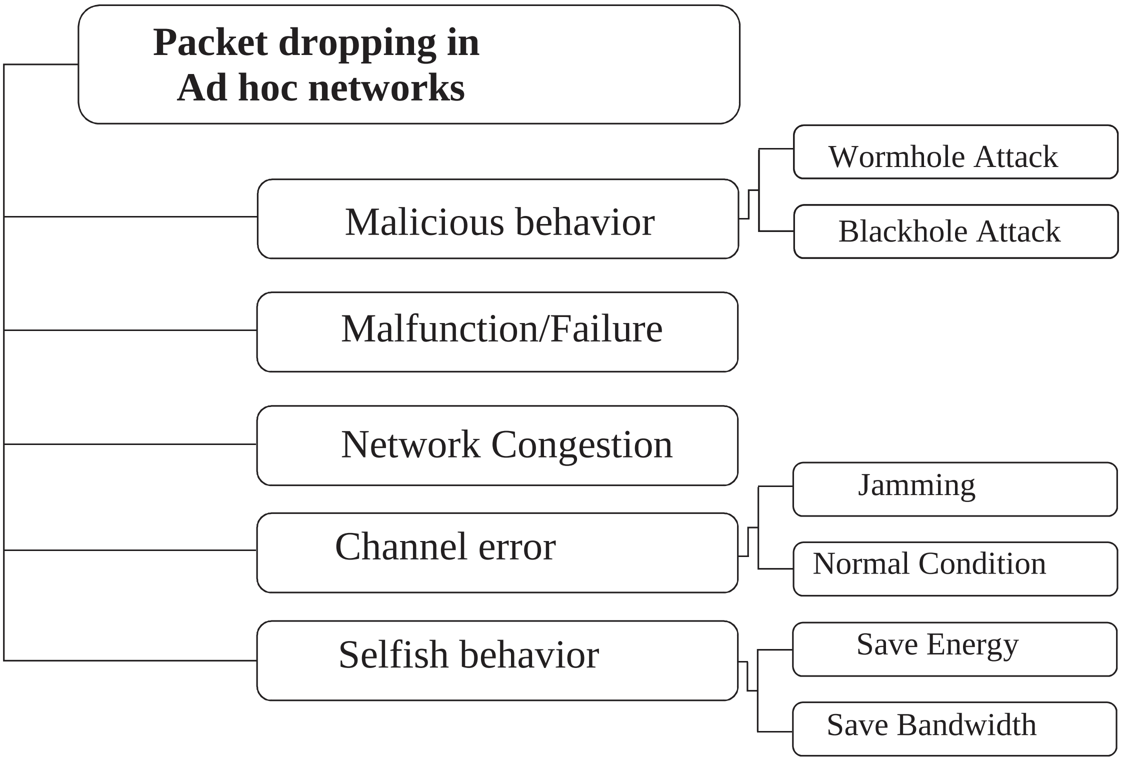


Figure 1: Packet dropping in ad hoc networks

1.3 Research Problem

In this thesis the problem addressed is shown diagrammatically in Figure 2. A set of *n* nodes formulating a MANET having *m* nodes acting maliciously by dropping packets either continuously or selectively is illustrated. A fraction *m* of the nodes deployed in the network is assumed to be misbehaving. Packets within the traffic are dropped as they are moved from a source to a destination. Given a path *P* of length *k*, we make an assumption that a set of *m* malicious nodes, where $|m| \leq k$, are present on *P*. These nodes can be located anywhere along *P*.

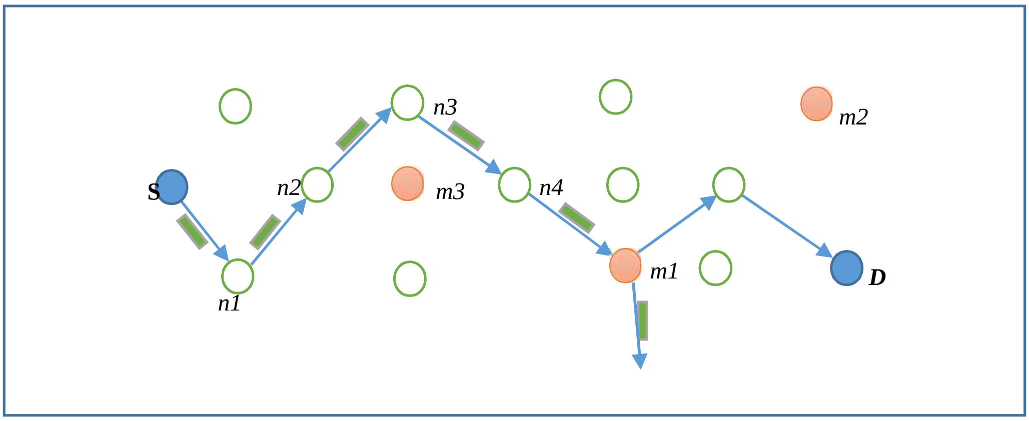


Figure 2: Problem Scenario

In Figure 2 *S* the source node is sending traffic to the destination node *D* along *P*. Node *m1* is dropping all the packets it is receiving. Our goal is to identify *m1*, provide evidence of its misbehavior, and identify the type of attack manifesting on *m1,* so that we can classify it accordingly for further action to mitigate the attack.

In our model, we consider several types of attacks that lead to an ultimate packet dropping action, for example: Wormhole attack, Black hole attack and several others. Each attack can be observed to affect network characteristics differently and analysis of the network characteristics can give us the state of the network, describing whether it is under an attack or in normal operation.

In the first phase of our work, we simulate different attacks using ns-2.35 simulator under the AODV routing protocol and in this phase we collect network characteristic data that is crucial for the next phase. The second and final phase we use the collected data of our desired parameters for training a neural network used for attack classification and detection in the Matlab platform.

1.4 Our Contribution

In this research we develop an IDS system to address specific, “Packet Drop Attacks” in MANET using an approach based on ANN that reinforces protection against such attacks able to support existing mechanisms to secure the network. We have developed a detection approach consisting of three components working together to detect and classify packet drop attacks within our MANET. Our system has three tightly coupled components namely: (i) data collection component (ii) data analysis component (iii) detection / classification component. Thus in this research will we monitor MANETs by use of IDS and analyze attacks by examining the network characteristics applying an artificial intelligent method for detecting malicious nodes at high computational and learning rates through pattern presentation with low false positive rate. Opposed to other existing approaches, our IDS allows behavior analysis of nodes packet by packet, without incurring the per-packet overhead. In a series of simulations we did, results obtained validate that our IDS can practically detect misbehaving nodes by identifying packet dropping activities in the network. Our technique achieves our research goals at significantly lower communication and energy cost compared to pre-existing methods. It is hence our goal to detect and identify the subset of misbehaving nodes that are dropping packets

Aims and objectives of this thesis work are summarized as follows:

- To understand operation of AODV protocol.
- To utilise ns-2.35, to simulate the network attacks to be illustrated in our work.
- To cultivate understand of Neural Networks and their implementation in Matlab, where the training will be used for attack detection and classification.
- To simulate several attacks using Reactive routing protocol (AODV).
- To obtain parameters :collecting network characteristic data e.g. packet drop rate, average number of hops per route, maximum end to end delay
- To training the ANN using the collected data
- To perform attack detection and classification.

- To analyze the results in the light of the Methodology used, make conclusion on detection technique applied and check detection performance rate paying attention to the number of false positives.

1.5 Motivation of the work

Many existing solutions for securing wired networks are not suitable for application to MANETs without modifications. The requirement for use of anomaly detection or misuse detection in the monitoring of the network requires that the IDS be able to discern normal from abnormal behavior. In this work we seek to address the following research problems:

- Lack of a systematic and organized way of collecting normal and abnormal behavioral patterns in MANETs owing to the dynamic nature of the topology and absence of centralized point of administration (router, switch).
- How to determine causes of malicious behavior as this may not necessarily be due to attacks but can also be caused by host movement or unexpectedly long delay due to unreliable channel.

1.6 Structure of Dissertation

The rest of this thesis is organized as follows:

Chapter two explores wireless networks, MANETs and routing issues. It will touch on misbehavior of nodes in MANETs, examining the vulnerabilities as well as the attacks aimed at them. An insight into intrusion detection techniques as a way to solving the problem presented in chapter 1 will also be given.

Chapter three outlines our methodology and focusses on how we will tackle the research problem outlined in chapter 1. An insight will be given of our technique and how it solves the presented problem, with detailed case studies also given.

Chapter four discusses about the simulation and implementation of our technique and show how to detect and classify the several types of attack under consideration in the study we choose and also how we implement and validate our technique and obtain result with tools like NS2 and Matlab.

Chapter five discusses the end result and the future work that can be carried out in order to enhance the solution and what are the conclusion of this work.

A LITERATURE REVIEW

In this chapter we initially provide a background on wireless networks, MANETs and routing issues. We touch on misbehavior of nodes in MANETs, examining the vulnerabilities as well as the attacks aimed at them. Lastly, we give a detailed explanation of malicious node detection as suggested in various literatures. However, this work focuses on a packet drop attacks detection mechanism. We also look at proposed techniques aimed at mitigating the impact of malicious nodes.

2.1 Wireless Networks

The turn of the twentieth century has seen a tremendous increase both in usage and adoption of Wireless technology as a de-facto communication standard within the academia circles as well as in industry. Currently, two variants of mobile networks are in existence. A mobile network that is infrastructure based with a gateway connecting to other networks fixed and wired like the internet. Base stations are used to bridge connections in these networks. Nodes in this environment roam liberally and connect to the nearest base station within range and can move out of one base station range to another creating a hand off between the two base stations. Hand off will enable the mobile unit to continue being in communication seamlessly even though it exchanged base stations [5]. These wireless networks are mostly found in business environments and include the wireless local area networks (WLANs).

Another variation of the wireless networks is widely known as ad hoc networks. It does not require any infrastructure to set up, hence it is usually termed as infrastructure-less mobile networks. In this mobile network, they are no fixed switches, routers and base stations. All nodes taking part in this network are capable of movement. Communication between nodes in the ad hoc network that are out of the transmission range of each other can be handled through a multi-hoping technique. Each mobile ad hoc network is endowed with routing capabilities, enabling it to re-route packets intended for other nodes besides itself through intermediary nodes, which are neighboring nodes within its range. Ad hoc networks have various applications including in military for search and rescue operations, business meetings for rapid sharing of information as well as satisfy data acquisition operations in hostile environments.

2.2 Ad Hoc Wireless Network

Ad hoc wireless networks are formed dynamically as nodes wander about arbitrarily in an infrastructure-less networks. These networks are different from other wireless networks in that they are self-organizing, do not have a fixed infrastructure, and communication using multi-hop routes [1]. An ad hoc network is referred to as a self-organized network due to lack of a central management point. Nodes formulating the ad hoc network utilise such functions as addressing, routing, power control to be able to function and communicate with each other. Ad hoc networks characteristic include the capability of a mobile node within the network to roam liberally while it is connected and communicating to other mobile nodes there in. Communication happens freely for any mobile node moving in any pathway. Either mobile nodes only or a mixture of both mobile and static nodes may formulate the composition of an ad hoc network.

A mobile node is usually constrained in terms of battery power and runs with reduced performance. Its power utilization needs should be carefully managed. Once computational demands for a mobile device increase or when high communication activity takes place, the power will be intensely expended. Thus, a compromise between the performance and energy consumption is required. In an ad hoc network, nodes are located at any place. Nevertheless, direct communication between a source node and a destination node over a single hop may not be possible due to the limitation of distance covered by the communication signal. Thus, packets being sent may go through several intermediate nodes to get to its destination. The intermediary nodes must forward the received packets to their neighbors to complete the communications between a source and a destination.

Another infrastructure-less wireless network is a Wireless Mesh Network (WMN). It provides affordable Internet services. In this network, the topology is organized such that it integrates several dissimilar networks such as the Internet, and Wireless LAN networks to enable communication. A WMN is comprised of mesh routers and several clients. The mesh routers perform routing functions and establish communication with other networks, nevertheless they have limited mobility [15].Mesh clients have two variants when accessing network resources via mesh routers, they will be either static or mobile. Mesh routers and clients communicate through wireless links to connect to other networks. Communication among mesh clients is facilitated by a mesh router. The infrastructure-less nature of the WMN, leaves the mesh routers with the responsibility to pass on packets in order to form a properly connected network. Mesh routers in the WMN can belong to several establishments, organizations, or individuals who attach their wireless equipment to the network. Although a cooperative assumption is always presumed in infrastructure-less networks, it may be inapplicable in WMN. This may be a result of a mesh router not passing on packets to neighboring routers in a way to preserve energy resources and bandwidth for its own communications. A router which is not cooperative is considered as a selfish or malicious node in this thesis. The presence of selfish or malicious nodes in certain locations in the network can have negative impact on throughput performance. Thus, in this thesis, we study ad hoc networks to illustrate how the performance is affected by the existence of malicious nodes. In ad hoc networks energy efficiency is a great concern owing to the fact that an ad hoc node is a resource constrained device.

2.2.1 Ad Hoc Routing

Routers are present in wireless infrastructure network to route traffic to desired destinations. Nevertheless in ad hoc network, each node acts as a router so as to pass on a packet to the destination, since there are no special routers. Therefore, a routing protocol is required to specify how communication will take place including route selection between sources to destination. There are two classes of routing protocols which are proactive and reactive routing protocols. A proactive routing protocol establishes a path to every node within the network at any given time. However, a reactive (on demand) routing protocol establishes a route to a given destination on demand depending on availability of a packet to forward.

2.2.2 Proactive routing

A proactive routing protocol (also termed table-driven routing protocol) generates a routing table of its neighbors and all nodes within the network before forwarding a packet even though some routes may never be used. An up-to-date routing table is essential for operations in an ad hoc network, hence routing updates are propagated to neighbors from time to time to keep the entire network information updated. This class of routing protocols is best in a more stable ad hoc network, with less changing routes, hence no need for rapid propagation of routing update packets. This will lower the traffic congestion in the network owing to reduced routing updates. Proactive routing protocols have the benefit of reduced latency in set-up since routes in the network are predefined and packets can be forwarded at any time. Nevertheless, route information is kept updated regardless of the fact that it may never be utilised. It is worth to note that when the nodes are mobile, the routing load in the network may increase. Destination-Sequenced Distance Vector (DSDV) protocol is one well known proactive routing protocol for ad hoc networks [16]. Sequence numbers for each packet are used to verify freshness of route information in the DSDV protocol. Each node maintains a local routing table, which get updated from time to time, when routing update packet bearing the most recent sequence number, is forwarded to all neighbors.

2.2.3 Reactive Routing

Reactive routing, also known as on-demand routing, establishes routes depending on availability of a packet to send to a given destination. Thus, it is not required for an ad hoc node to update its routing table occasionally. Accordingly, it is appropriate for a very dynamic ad-hoc network with a network topology that can change rapidly requiring the routing protocol to adapt quickly. However, this poses some latency in the initial establishment of a route prior to packet transmission. Dynamic Source Routing (DSR) protocol [17] and Ad hoc On-Demand Distance Vector (AODV) routing protocol [18] are examples of reactive routing protocols.

To initiate a route request, the DSR protocol broadcasts a route request (RREQ) packet to all its neighbors, whenever a node requires to forward a packet to a specific destination whilst not having a route stored which points to that destination. Neighboring nodes which receive this broadcast but are not the desired destination attach their addresses to the RREQ packet and rebroadcast it. Upon receiving the RREQ packet, the destination node will forwards a unicast route reply (RREP) packet back to the sender over the reverse path of the source route information in the RREQ packet received. The DSR protocol has supplementary means that help in learning latest routes by working in a promiscuous mode to tap on any communications. Thus communication taking place within the hearing range of the nodes can be overhead, therefore nodes are able to snoop on the RREP packets and adding a new source route to the route cache. This is time conserving and removes the traffic overhead of establishing a path in the communications that follow.

AODV (on-demand) routing protocol, which operates similarly to DSR utilises the notion of sequence number and routing table as in DSDV. In the existence of a source node having a packet to send, it makes a look-up for the route of the intended destination that is valid in its routing table. Upon finding a route, it will forward a packet directly to the nearest neighbor towards the destination. However, if no route record is found, the RREQ packet is broadcasted to all its neighbors. Intermediary nodes update their routing tables with the sender ID as soon as they receive this packet that is the current neighbor, and rebroadcasting the packet to its neighbors.

The destination node upon receiving the RREQ packet, checks whether the sequence number is higher than the previously observed one. This RREQ will be deemed as a fresh packet. The new sequence number will be noted down, and then a RREP packet will be sent back to the sender. In AODV each intermediate node has its own routing table and there is no source route information in each packet header. Thus the packet header for AODV is rather smaller than that of DSR.

More traffic is generated in DSDV than in AODV routing protocol. This is because nodes using the AODV routing protocol have no need to maintain routes to all destination in the network except routes for specific destinations. Hence, in this thesis, we adopt AODV routing protocol for the reason that it establishes a route as and when it is required.

2.4 Vulnerabilities and attacks in Ad Hoc Network

A very pertinent issue in a mobile ad hoc network is the provisioning of secure communication in the network. Distinct features of these networks for instance the inherent open network architecture, a shared wireless medium and a highly ever changing topology bring about countless challenges to the designing of protection schemes. In light of these challenges highlighted, it becomes imperative that there be development of security solutions that achieve broader security scope while ensuring desirable network performance. Mobile ad-hoc networks are open to authentic users as well as threats. The security design in these networks lacks a standard security mechanism to resolve this issue. Absence of centrally managed secure routers enables threats to easily exploit and compromise a mobile ad hoc network. Security goals attained through cryptographic mechanisms and the likes may require infrastructure support not available in MANETs.

The dynamic nature, quick and random network topological changes typical of mobile ad hoc network bring about connectivity variations among the nodes at any given time. Routing path amongst the mobile nodes in the network is constantly varying as nodes roam about within the network. Thus Mobile ad-hoc networks need to adapt the dynamic network conditions. Communication related functions for the mobile nodes, ought to be optimized to prevent needless power expenditure. Designing power efficient systems for wireless ad hoc networks pose quite many challenges. The infrastructure-less nature of an ad-hoc network makes each node possess routing capabilities. The MANETs existence can be a success if nodes forming the network can be receiving packets often .The ability to balance traffic load in ad hoc networks is crucial amongst nodes so as to release power constrained nodes while traffic is forwarded through more active nodes. Selfishness of an individual node is another area of concern. Nodes participating in the network are required to cooperate in utilizing their energy resources in forwarding packets as intermediate nodes despite not originating any packets in the ad hoc network. Therefore some nodes are not willing to cooperate for others, hence resultantly compromising network performance [9].The usual assumption in MANETs that every node cooperates in coordinating processes amongst distributed nodes is inapplicable in hostile environments. Malicious attackers can easily compromise the network in MANETs since cooperation is based on assumption rather than being enforced.

Routing and data packet forwarding serves as the chief operations in MANETs, interacting with each other to fulfill the mandate of end to end packet delivery. In a MANET, the nodes exchange data between them through ad hoc routing protocols and maintain associated routing states at each node accordingly. Data packets are relayed based on the routing states by intermediate nodes through a predefined route to the end node. However, both routing and packet forwarding operations are susceptible to attacks, creating network problems.

Network layer vulnerabilities can be categorized as either routing or packet forwarding attacks. This depends on the target operation of the attacks. At any given instance an, attacker may alter the source route contained in the RREQ or RREP packets by removing, swapping the order of or appending a new node into the list [10]. An attack on the routing protocols can enable re-routing of traffic to specific destinations. An attack can form routing loops, cause unbearable congestion and channel contention in parts of the network. Source nodes might be prevented from identifying a desirable route to an end node by multiple colluding attackers even cause a network partition in the worst case.

Packet forwarding operations may be disrupted by an adversary but have less impact on the network. However the data packets are sent in a deliberately unpredictable way. An instance is that an adversary may alter content, drop packets or make replica of the packets that have been forwarded already. An attack known as Denial-of service (DoS) is another type of packet forwarding attack in which the attacker floods the network with useless packets. This results in crippling of the network in the MANET.

Ad hoc networks are established on the notion that all nodes cooperate to realize network services. However, this implied trust model depends on the willingness of the individual network node’s cooperation. Many a times the trust model is often breached. Some adversaries eavesdrop on packets in the radio range, by working in promiscuous mode and sniffing packet. Some nodes act in a selfish manner, in order to conserve their energy, by declining to pass on traffic of other nodes. Malicious nodes can drop all packets channeled through them and hence compromise network performance. Regardless of the intention, such mischievous nodes infringe ad hoc network norms of node-cooperation. Node misbehavior in the routing function has been shown to have an undesirable bearing on the network throughput [1, 2, 3, 4, and 5]. Additionally, owing to the challenges imposed by the medium, disruptions to communication can occur. Several attacks become possible, an example is congesting medium, or creating noise in the communication.

Numerous threats aimed at routing protocols in ad hoc networks have been revealed [6, 10, 12, 14, and 18]. Here we give a brief explanation of some chosen attacks. A replay attack is a type of attack that can adversely affect a MANET. A malicious node can carry out a replay attack by re-producing and sending old packets causing stale updates to the routing tables. This type of attack becomes effective as long as the control messages does not have a timestamp.

In a sinkhole attack, an unruly node tries to attract traffic by misleadingly advertising a shortest route to multiple destinations [16, 17]. Consequently, adjacent nodes route their traffic through the misbehaving node, allowing it to drop/modify/analyze the packets coming to it. A network in which all nodes occasionally send data to a sink like in a monitoring network, is particularly exposed to the sinkhole attack since data packets have a single destination. When a mischievous node advertises the shortest path to a particular destination node whose traffic it wants to intercept then that attack will be known as a blackhole attack [20, 25]. The mischievous node replies to the route request it receives from the source. Since most route discovery processes accept the first route discovered, the traffic is routed through the mischievous node.

Rushing attacks affect mostly on-demand routing protocols in MANETs [22, 24]. Misbehaving nodes, in this type of attack alter the route request packet and changes the routing path. After the packet has been altered, the node hurries the packet to the next hop. Since on-demand routing protocols allow forwarding only the first received Route Request packet from each route detection process, a first copy of the route request packet is then accepted by most routing protocols thereby ignoring and dropping all the subsequent ones. The forwarded modified route request packet transmitted before any others is used for the duration of the route discovery process and resultantly establish a false routing path.

A wormhole attack, involves taping of traffic from a part of the network and replaying it in another part [21, 11, and 18]. An impostor node X located within transmission range of legitimate nodes A and B, where A and B are not within transmission range of each other. Impostor node X merely tunnels control traffic between A and B (and vice versa), Packets received at one end of the wormhole are transmitted back on the other end. Since the wormhole is a low-latency link, nodes on either side of the wormhole will appear as neighbors, and eventually all packets destined from one side of the network to the other will traverse the wormhole.

An irrelevant A – B connection can be artificially created by an impostor node X by worm-holing control packets between A and B (Figure 2.1). A longer wormhole can also be created by two colluding imposters X and X′ (Figure 2.2).

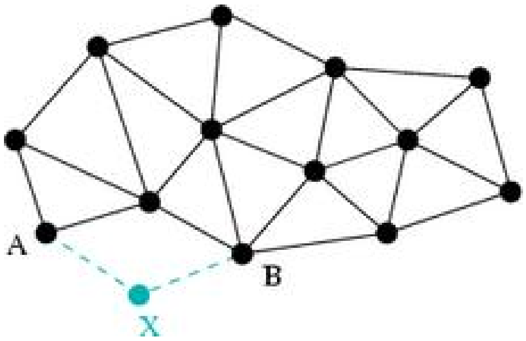


Figure 1: A wormhole created by node X

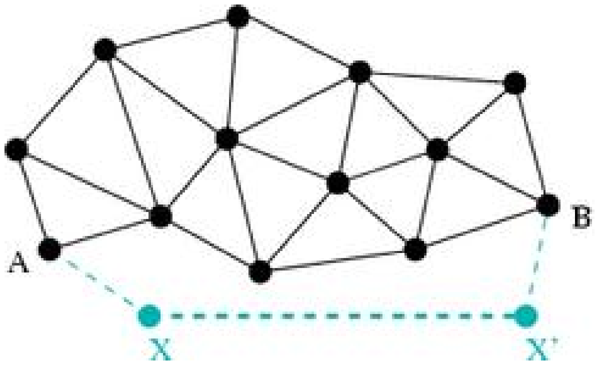


Figure 2: A longer wormhole created by two colluding nodes X and X′.

To successfully exploit the wormhole, the attacker must wait until A and B have exchanged sufficient HELLO messages (through the wormhole) to establish a symmetric link. Until that moment, other tunneled control messages would be rejected, however, once created, the A - B connection is totally controlled by the attacker

Another common attack in ad hoc networks known as Black hole attack, involves a node that fails to relay traffic control messages, hence resulting in the network experiencing connectivity problems. In a network where a fault tolerance scheme is not in place a network partition will occur.

Several attacks that affect the routing layer of MANETs have been describe in brief above, however it is the packet dropping attack that is formulating our research focus [23, 2, 3].The attacker, which in this case is the mischievous node takes part in the routing discovery process. Once the routing path is established, the misbehaving node simply rejects to forward packets to the next hop. Performance degradation can be the resultant and can be observed by source and destination nodes. The nodes can recognize that a performance drop has occurred on the routing path, but are unable to determine the problematic link. This thesis focuses on this last type of misbehavior in which the source and destination attempt to determine the node(s) that drop packets along the routing path. We now examine possible ways to handle malicious nodes and then describe related work with respect to the misbehavior identification problem.

2.4 Misbehaving nodes in wireless ad hoc networks

Misbehaving nodes cause undesirable activities within the network by performing certain operations that lead to network performance degradation. Several mechanisms were proposed to mitigate the effect of misbehaving nodes. These can be divided into malicious nodes and selfish nodes mitigation as shown in Figure below. We now turn our focus on mitigation of packet dropping attacks in this section.

2.6 Approaches to Selfish Node Mitigation

Previous studies have contributed immensely towards mitigation of selfish nodes in ad hoc networks. The notable proposed mechanisms are incentive-based approaches, and reputation-based approaches. The former seeks to discourage a node from becoming selfish. On the other hand a reputation-based technique seeks to detect a node behaving selfishly and provide a suitable remedy.

2.4.1.1 Incentive-based mechanisms

This approach is based on a currency (incentive) mechanism in which communicating parties must pay for packet relay to occur [40, 41, 42, 43, 44, 45, 46]. If a node takes part in forwarding other nodes' packets, it gains credits to communicate for itself. Hence this eliminates selfish behavior in nodes. A proposal to make use of virtual currency termed as nuglets was put forward by Buttayan and Hubaux [40], in which when a node forwards a packet for others, a nuglet counter of that node is incremented by one. A node that wishes to send a packet, will only be allowed to do so if it has accumulated adequate nuglets. The nuglet counter must be maintained by a tamper proof device.

An algorithm to discourage selfish behavior was proposed by Miranda et al. The algorithm keeps track of each neighbor's state, namely friend, foe or selfish [41]. Nodes constantly observe their neighbors behavior and exchanges control message from time to time. Upon receiving control messages, as indicated in the proposed algorithm, the state information for each neighbor is updated. This technique involves a high overhead as a result of broadcasting of control messages from time to time undertaken by nodes.

“Sprite" is a credit-based system that was proposed by Zhong [42].Nodes using the Sprite mechanism, will keep a record of receipt after receiving a packet. A node periodically report the recorded information to a Credit Clearance Service (CCS), which helps in determining the charges or credits of each node involved in packet transmission. Raghavan [43] came up with priority forwarding mechanism for “self-interested" nodes to participate in packet forwarding. Two classifications of packet forwarding mechanism namely, priced priority forwarding and unpriced best-effort forwarding which aimed at rewarding nodes that forwarded priority packets by credits were the work of Raghavan, et al. To send priority packets, each node will be charged taking from the credit account. In the best-effort mode nodes may send packets free of charge. This mechanism encourages nodes to participate in forwarding priority packets to acquire credits to forward own packets. Nodes without credits can also send packets, which when received by other nodes are treated without priority.

Crowcroft in [44] proposed “congestion prices” where bandwidth and energy are used for modeling of an incentive scheme. This mechanism uses directional wireless antennas for broadcasting packets through multiple routes. A rate control model was used [47], operating in a distributed manner. Updates based on current power and bandwidth usage will be embraced. Fee Arbitrated Incentive Architecture (FAIR) was proposed, which has its main focus on fairness as well as collaboration [45].Fairness in this context is achieved by ensuring that a node’s gain is proportional to its contribution in the network. To enable FAIR performance Feedback schemes are proposed and are in place to make dynamic adjustment as and when unfairness has been pointed.

In [46], Zhang proposed a Secure Incentive Protocol (SIP). It uses a session-based approach, which differs from a motivation-based approach. In the SIP protocol, a session initiator and a session responder are present, they are an equivalent of a source-destination pair. Three phases are there in SIP namely, Session initialization phase, Data forwarding phase and Rewarding phase. Intermediate nodes are given an amount of credits depending on total packets forwarded. Security for the protocol is assured using Asymmetric key cryptography against credit fraudulence and various other attacks. In [50], Hauspie and Simplot-Ryl proposed to use a virtual money mechanism over a route discovery protocol. This enables source node to maintain correct charges for packet forwarded over a route, hence a contrast on incentive based schemes where source nodes do not know charges for forwarding a packet. The proposed algorithms given above attempt to resolve the selfishness problem by providing ways that encourage a selfish node to contribute in network operation.

The different incentive algorithms given above seek to solve the collaboration problem in MANETs, but also bring about much complexity. The motivation-based approach proposed by Huang, et.al, is meant for specific applications rather than general applications because of its complexity issues [51].Incentive schemes are considered not reasonable to all nodes, since position of node counts for amount of credits it will gain. Marginalized nodes will not have a good chance to forward their own packets. The collaboration of nodes is demonstrated mathematically through the use of game theory. Several assumptions are put forth in most of these models practically infeasible in a real MANET. Certain key issues like complexity also make this approach inapplicable for use in real networks.

2.4.1.2 Reputation-based mechanisms

In a reputation-based mechanism, a reputation system is in use for the purpose of identifying and rating a selfish node. In this mechanism, decisions on who to trust are based on the node's reputation, the reputation is also used to encourage other nodes in the network to be trustworthy. Whereas reputation in this instance is a measure of how a node taking part in the base protocol performs as perceived by others [40].Normally, a selfish or malicious node may drop data packets for many varied reasons including a bid to preserve its energy or to intentionally degrade network performance. By making use of the data of packets that are not forwarded by every node the selfish or malicious state of the node can be established. Communication signals in MANETs are propagated through the airspace in all directions using omnidirectional antennas. Thus, direct communication happens between any two nodes within each other's communication range. These nodes overhear their packets being forwarded by their respective neighbor and this property is referred to as a passive acknowledgement (PACK) technique. Watchdog, another technique in which all neighbors' behavior are regularly monitored by operating in a promiscuous mode was proposed [48]. It is used over the DSR routing protocol. In the Watchdog approach a source node continuously eavesdrop on its neighbors and maintains a count of the packets not forwarded so as to establish its neighbor's participation in forwarding packets. When a certain threshold limit has been exceeded by the counter, a report will be sent to a pathrater mechanism. This mechanism is called in after the detection process has completed [48], it then act responsively to assess every route to allow packets to be sent through the most efficient route.

In the case of a selfish node, there is no punishment given in Watchdog/Pathrater. The battery power consumption is a major drawback in Watchdog, because all nodes have to be active all the time putting more demand on the power uses and needs [41]. A collaborative reputation mechanism known as “CORE" was proposed in [452], utilising “Watchdog" in monitoring. Each node has a reputation table to help keep track of reputation values of other nodes. Favorable ratings can be circulated to prevent acts of selfish nodes aimed at discrediting the reputation system by sending false negative rating factors to other nodes. Penalty action is given to a selfish node in this reputation system by refuting it any support. A proposal to use cooperative games and non-cooperative games approach to evaluate the effectiveness of the CORE mechanism to detect and rate neighbors’ reputation was tabled by Michiardi [42]. In [12] another reputation mechanism known as “CONFIDANT" (Cooperation of Nodes: Fairness In Dynamic Ad-hoc NeTworks) is proposed. It has four main components that is a monitor, a reputation system, a path manager, and a trust manager. Each component should be implemented as part of each node.

Each node listens to the transmission of the surrounding neighbors and also monitors routing protocol behavior within the network. A trust manager is assigned to control ALARM messages, sent when a misbehaving node is identified. Every node in a network is rated by a reputation system, on the other hand a path manager is in existence to rank a path according to a security metric, e.g., reputation of the node in the path and to get rid of any path containing a selfish node. A path manager goes on further to penalize a selfish node by denying all services to it.

A technique known as “CineMA” (Cooperation Enhancement in MANETs) proposed in [43]. It responds to a selfish node through restricting packets it can forward. Penalty schemes used in CORE and CONFIDANT are also employed in CineMA. However, CineMA differs from these two in that it only requires a set of nodes to execute required operations. CineMA consists of three main modules that is a Watchdog module, a reputation system module, and an interface queue module. System monitoring to collect information is performed by the Watchdog module. The collected information is passed to the reputation system so as to establish the degree of cooperation given by the number of communicated packets. At the interface queue module, a restriction to the number of packets which a selfish node is allowed to transmit can be imposed based on the collected information. Cryptographic mechanisms are used in CineMA to provide security among communicating nodes. CineMA has the advantage of controlling the transmitting rate of selfish nodes. The work explored up to now utilises the Watchdog mechanism as an important part of the protocols. However, the watchdog mechanism has numerous shortcomings, including ambiguous collisions, receiver collisions, and inability to detect packet drops using limited transmission power, false misbehavior, collusion and partial dropping [48].

In light of the weaknesses inherent in the watchdog mechanism a replacement mechanism known as “TWOACK" is proposed in [44, 45]. Use of the TWOACK mechanism ensures that a forwarded packet has been received two hops away using a TWOACK acknowledgement packet. A counter will increment by the number of non-forwarded packets if there is a timeout when no TWOACK packet is received. If a selfish node is discovered within the network, a route error (RERR) packet will be sent back to the source node. A RERR packet is overhead by other neighboring nodes as they traverse the network and in turn keep track of the selfish node to avoid its path.

2.4.2 Handling Malicious Nodes

The existence of ad hoc networks is increasingly under threat to malicious nodes present within or at times intruding into the network. Dealing with such nodes is vital so as to stop genuine nodes getting affected so as to ensure that the MANET performs its various functions without a glitch. Mitigating malicious nodes can be handled by employing two different techniques classified as: (i) detection and response, (ii) prevention and protection. In prevention and protection mechanisms cryptographic methods are utilised to safeguard nodes against attack from malicious nodes. However, insider attacks are not prevented by this mechanism. A detection and response mechanism may be effective in the detection of misbehaving activities and in responding to attacks. In this thesis, we focus on addressing detection, response mechanisms and their impact on overall performance of a MANET. We now turn to Detection and Response Mechanisms and have an understanding of how they have been used to outwit malicious nodes within an ad hoc network

Mechanism	**Operation**	**Nodes with detection capability**
Watchdog	• Using watchdog mechanism and avoid using a selfish node in a path	Every node
CORE	• Using weighted average rating to combine direct and indirect reputations • All nodes Only positive reputation is exchanged • Selfish node is isolated upon detection	Every node
CONFIDANT	• Using weighted average rating to combine direct and indirect reputations • All nodes Alarm is sent upon detection • Selfish node is isolated from network	Every node
CineMA	• Use number of received and forward packets as level of cooperation • Limit the network usage upon detection	Not every node
TWOACK	• Using TWOACK packets to guarantee forwarding packets in two hops • Avoid using a selfish node path	Every node

Table 1: Error! Reference source not found.

2.5 Intrusion detection in Wireless Ad hoc Networks

A system aimed at finding an unwanted or unexpected activity happening in the network as well as abnormal behavior is called as "Intrusion Detection System (IDS)" and the way that it carries out these actions can be termed as "Intrusion Detection". Malicious behavior within the network might consume network resources excessively and affect integrity as well as confidentiality of the packets being transmitted in the network.

Many approaches have been proposed for Intrusion Detection in MANET. IDS are classified either as behavior based or authentication based. The behavior based approach focusses on the behavior of a node and its nodal activities. On the other hand authentication focusses on authenticating the identity of a node and the usage of encryption keys (public key and private key pairs) falls into this category [50, 51, 52, 53].Behavioral based approaches employ algorithms which define intrusion based upon its nodal activities instead of its identifier. Following the behavioral approach is better because of the following reasons [50, 27]:

i. Difficult in replicating a node behavior.
ii. Identities of nodes are not stored anyway.
iii. As the node identifiers should be unique, the process of deployment will become expensive both in terms of time and cost.

Reasons posed above serve as challenges, our focus will be on behavior-based intrusion detection. Behavioral based intrusion detection are efficient, lightweight and easily scalable to Intrusion Detection in MANET. Intrusion Detection, Intrusion Detection System (IDS) and its classification is further discussed in this chapter.

2.5.1 Background on Intrusion Detection

Intrusion was defined in [51] as a process which may compromise the confidentiality, integrity and availability of a resource. Intrusion can be prevented with techniques like encryption and authentication. However, prevention on its own is inadequate, there should be a strong defensive system which protects the network.

IDS provides a second line of defense to secure the network systems. Security remains a crucial and the most important concern for any network. Maintaining security of a network becomes more complex in proportion to the growth in the size of the network. The need for security in MANET is very high given that there is no fixed infrastructure, lack of centralized authority and an ever dynamically changing network topology. Security attacks emanate mainly from inside and outside attacks. Prevention techniques serve to defend against outsider attacks such as authentication and encryption. Tackling insider attacks effectively, require intrusion detection techniques. Intrusion detection can be categorised depending upon the audit data as either host based or network based. The former use the operating system and the logs for analysis whereas the latter deals with capturing and analyzing packets from the network traffic [52, 28].

Many proposals have been given in light of intrusion detection approaches, a cluster-based detection approach was proposed by Huang et .al in [29]. Here, a node is elected the clusterhead to perform the functions of IDS for all nodes which are within a cluster. The cluster formation employs protocols that achieve fairness also securing the clusterhead election.

2.5.2 Intrusion detection and response mechanisms

Detection and response mechanism can be used at any time a prevention mechanism is unsuccessful. Reasons for the failure occurring can be compromise by a malicious attacker or a misbehaving insider amongst other reasons. Detection mechanisms can prove to be efficient when we need to secure a network against such types of attacks given above. We can further minimize the magnitude and scope of a successful attack. Three categories based on classification of the detection analysis methods can be found on the detection and response mechanism namely anomaly, specification and signature-based detection.

2.5.4 Classification of IDS

IDS can be classified into three important categories basing on the detection techniques: anomaly, misuse and specification detection systems [50, 17, 51, 52, 28, 30]. Further sections in this chapter will give some more detailed explanation about the above mentioned detection systems

2.5.4.1 Anomaly-based Detection

Anomaly-based detection for intrusion detection is widely used mainly because the mobility features of an ad hoc network makes it difficult for specification-based and signature-based detections to accurately detect an intrusion. A proposal for the extension to the Watchdog mechanism for detecting colluding nodes or a wormhole attack, which would improve the performance of the watchdog scheme was put forth by Patcha [61]. In the proposal ad hoc nodes would be classified into two types:

ordinary nodes and trusted nodes. An implementation of the watchdog functions will be restricted to trusted nodes only, the ordinary nodes will just perform all regular activities instead. Three threshold counters are maintained at nodes implementing the Watchdog functions. Detection of misbehaving nodes in the Watchdog scheme is reliant upon passive acknowledgement hearing from neighbor nodes. Some considerable time is taken on the information collected to identify misbehaving nodes. Watchdog detection mechanism do not guarantee high detection rates, hence the detection rates are usually very low. To solve the low detection rates, Kargl proposed a Mobile Intrusion Detection System (MobIDS) scheme [52], to be used over the Secure DSR (SDSR) protocol [57]. MobIDS aims to improve the detection rate, this technique uses sensors to collect information from an activity-based overhearing mechanism, Watchdog and probing packets.

More memory and processing power is required at each node as nodes have to listen to the wireless channel continuously. Nodes send probe packets to detect attacks. Medidi, et.al. Proposed a technique for improving the detection of packet dropping attacks by correlating inter-layer information [62]. Source node will have the detection manager implemented on it.In addition, a data collection module and a data analysis module will have to be included. The data collection unit will gather local node information, e.g. AODV route request, route reply and route error messages. Of the information extracted, only deemed as useful will be fed into a data analysis module. The detection mechanism is understood to successfully detect attacks with low false positive rate. However, considerable amounts of memory are consumed on keeping records of data. Considerable processing power is required to analyze all data.

A proposal to detect malicious packet dropping attacks using distributed probing technique was put forward in [63]. Packets are sent regularly for probing to several destination nodes. Destination nodes that receive probing packets send back a response, failing to respond to the probing packets will brand the node as malicious. Having detected any malicious nodes, a new pathway must be selected to avoid using the path passing through a malicious node. Suggestions to use cryptographic techniques to secure probing packets are also enshrined in the given proposal. However, the given probing techniques bring about extra overhead due to the distribution of probing packets, which is highly unfavorable.

Another intrusion detection technique was proposed by Wang [64] with the sole motive of identifying false destination sequence number attacks In Wang’s work a malicious node sends a higher sequence number to the destination than a normal sequence number, so that it will start using the fake routing information. AODV routing protocol is particularly vulnerable to such an attack since it uses a sequence number to determine freshness of routing packets received. Source node keeps watching a route request packet broadcasted so as to detect whether a sequence number in the route request packet has been modified. An intrusion detection mechanism over SecAODV [55] was proposed by Patwardhan [65]. It was based on the notion of threshold anomaly detection, which detects data packet dropping attacks, which only drops data packets but forwards routing packets. Simple algorithms are mostly used to detect a malicious node, use of more complex algorithms is seen found in Sinkhole Intrusion Detection System (SIDS) [66] for detecting a sinkhole or a blackhole attack. In this detection mechanism three important parameters were used namely: (i) sequence number discontinuity or duplication, (ii) previous image ratio, and (iii) route add ratio. Computated average value of differences between the current and the last sequence number at each node is known as the sequence number discontinuity. The previous image ratio is defined as a ratio of the number of images verified with proportion to the number of total images received. Lastly the number of verified images is the number of route-broadcast packets received at a node from a certain neighbor that can be traced and verified to the earlier transmitted packets by other nodes having the same sender-receiver, sequence number and with appropriately inserted route records.

2.5.4.2 Specification based Detection

In Specification based Detection a set of constraints that describe the correct operation of a program or protocol are established, the execution of a program or protocol and is monitored with respect to the defined constraints.

A specification-based intrusion detection system for AODV was proposed by Tseng [71]. Two types of nodes are used in this technique, a regular node and a network monitor node. Incorrect RREQ and RREP messages are detected using the network monitor (NM) nodes. The NM nodes will listen and keep the last received RREQ and RREP messages for source and destination nodes. They are distributed within the network and keep the RREQ and RREP messages in a tree-like structure to ease the tracking of an attack.

Authors in [30], outlined that detectors in specification based detection works by detecting the intrusion against the background of the normal traffic in the system as these detectors have a better chance of correctly detecting truly interesting events in the supervised system, since they both know the patterns of intrusive behavior and can relate them to the normal behavior of the system.

A finite state machine which keeps track of events so as to detect misbehavior actions is also used by AODV Extended Finite State Automaton (AODV EFSA) [50]. In this mechanism specification based detection is used together with anomaly based detection to try to effectively detect many types of attacks in an ad hoc network. Nevertheless, its implementation brings about high complexity.

2.5.4.3 Signature-based Detection

Also known as misuse detection, this system keeps a record of all the signatures from earlier known attacks, and then use it to compare with the present captured data .If any match pattern occurs and is noticed then it is treated as intrusion. Works mostly in virus detection systems, which cannot detect new kinds of attacks.

Authors in [36] proposed an architecture that implements a Local Intrusion Detection System (LIDS) agents on each node. Examples of misuse detection systems include IDIOT [35], STAT [31], [17, 37], and these mechanism have used patterns of well-known attacks or weak spots of a system to match and identify the intrusions.

A signature rule can take the following form, for example the “guessing password attack” can be “if there are more than 4 failed login attempts within a minute”.

Advantages of signature-based detection
are that it can be efficient and accurate in detecting instances of all known attacks, while the disadvantage is that it doesn’t have the ability to detect new attacks [17].

A signature based detection which made use of State Transition Analysis Technique (STAT) called AODVSTAT [72], was developed in a wired network, for attack detection in AODV routing protocol. An attack signature, in STAT is defined by a sequence of actions that an attacker performs towards compromising system security. In this mechanism, sensors are in use to perform a stateful analysis of packet streams to detect signs of intrusion. One hop and distributed attacks to AODV routing protocol are detected using AODVSTAT with low false-positive rates.

Methodology, System Models and Experimental design

Solutions previously proposed to solve packet dropping attacks cannot either be directly adapted to MANETs or they are too expensive to implement and might further entail modification to all the nodes in the network, which can rather be taxing. Furthermore, it is imperative to develop solutions that are scalable, implementable and capable of detecting malicious behavior while the communication is in progress. It can be established from the research literature that a malicious node can degrade network performance severely in a number of ways. In this chapter we state our proposed methodology, we go on to illustrate how our technique works on an example problem we shall consider through-out this thesis , we also state our assumptions with regards to the network and adversarial Models.

3.1 Proposed Methodology

In this thesis we motivate the use of a technique based on artificial neural network for detection and classification of packet drop attacks. The detection will be performed on the collected network characteristic data. In this technique we collect and analyse locally available data. We mainly focus on all nodes that participate in the route discovery process successfully. The data collection, analysis, detection and classification components form the core of the detection technique. Local data collected such as route request and route error messages, are used to extract important parameters that affect network characteristic data which will be passed as input for training the detection engine. This will in turn be used to detect and classify misbehaviour in the network. This information is gathered by the data collection component during the duration of the simulation period. Collected data is passed on to the data analyser component which extracts useful information or parameters from control messages being exchanged in the network, for use as input in the second phase involving training the detection engine. The detection engine will check for any deviation from normal behaviour and classify the attacks according to their types as well.

Important information extracted by the analyser constitute:

- The reason for dropping the packet, whether it’s a broken link or intentional.
- The identity of the node that was unable to deliver a packet.
- The address of a node that dropped a packet whose time-to-live had expired from an ICMP time exceeded message and the destination of the original packet.
- The destination of a TCP packet that timed out.
- Time each message was received or each event occurred.
- Type of packet sent, its size, and the layer from where it’s originating from.

Some of the problems are easily surpassed in our proposed technique which are associated with the existing techniques given in literature in the previous chapter. Our technique aim to solely detect packet dropping faults, specifically aimed at the data packet. This research focusses on malicious packet-dropping, where a node intentionally drops packets that are destined for other nodes. The methodology and the algorithm used for detecting malicious packet dropping is shown diagrammatically and discussed further with an example scenario in the following sections. This technique relies on readily available information at different network levels, to detect the presence of malicious nodes and does not require modification or cooperation of all the nodes in the network. Our technique involves collecting and analyzing locally available data in the first phase. Local data such as AODV route request and route error messages, and total data packets sent is used to detect malicious behavior in the network.

Features of this technique include:

- Portable: This technique does not require any new protocols. It works with existing protocols, such as AODV, mobile IP, and ICMP which allows the technique to be easily ported to many different systems.
- No extra processes to consume battery: this technique does not dissipate or waste battery power for exchanging extra control information with the neighboring nodes. We make use of data that is readily available in the network.
- No security associations: Since this technique does not need the cooperation of other nodes in the network, there is no requirement to have security associations between the nodes.

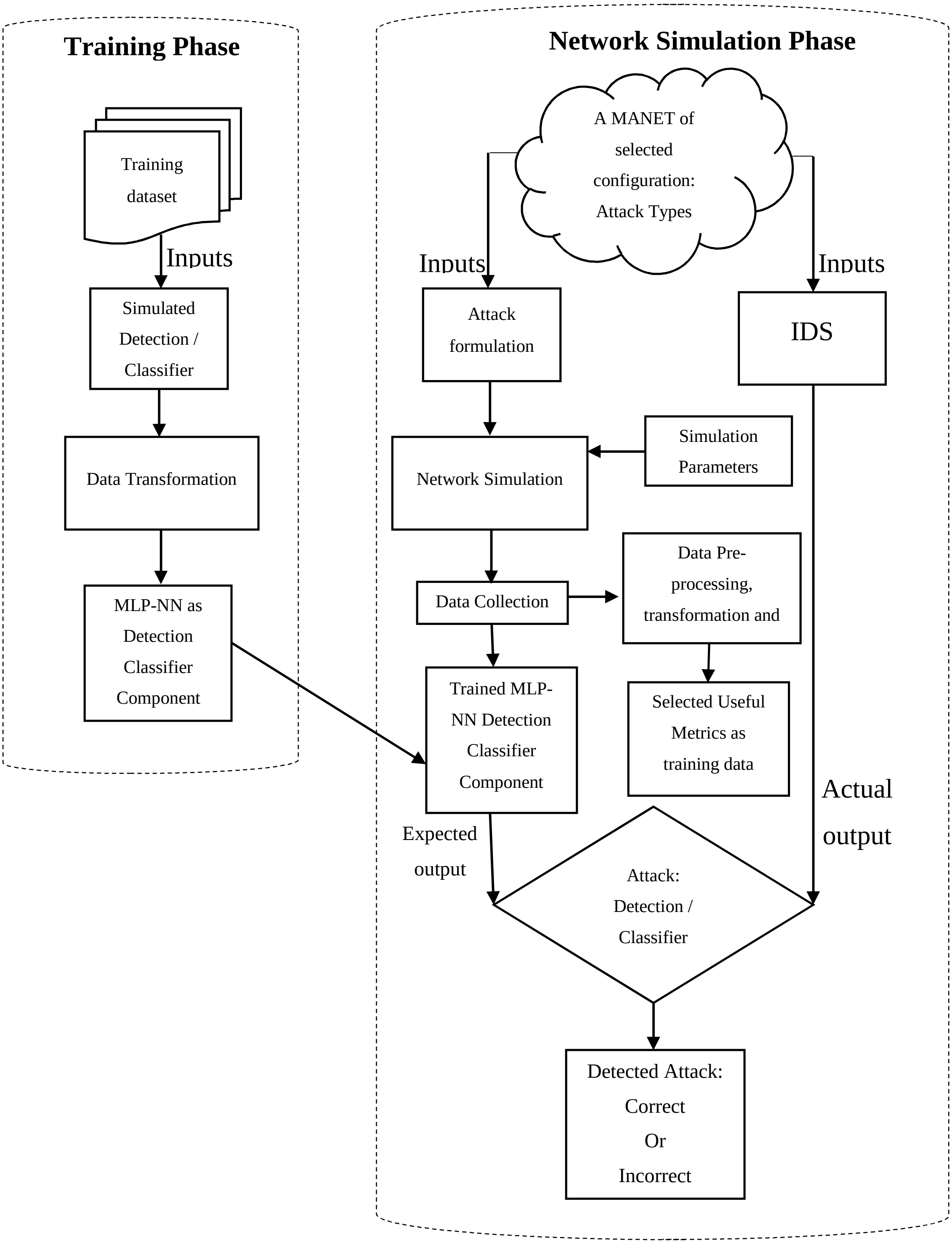

Training Phase
Training dataset
Inputs
Simulated Detection / Classifier
Data Transformation
MLP-NN as Detection Classifier Component
Network Simulation Phase
A MANET of selected configuration: Attack Types
Inputs
Inputs
Attack formulation
IDS
Simulation Parameters
Network Simulation
Data Pre-processing, transformation and
Data Collection
Trained MLP-NN Detection Classifier Component
Selected Useful Metrics as training data
Actual output
Expected output
Attack: Detection / Classifier
Detected Attack: Correct Or Incorrect

Figure 1: Schematic Diagram of the proposed Technique

- Other security mechanisms such as “Nodes Bearing Grudges” and “Intrusion Detection in Wireless Ad-Hoc Networks” require security associations between neighboring nodes to authenticate the messages passed among themselves.

- No Certificate Authorities: No additional infrastructure is required, such as network controllers or certificate authorities.
- Highly scalable: Since this technique is not tied down by the cooperation or security associations between neighboring nodes, it can be incorporated into as many nodes as needed making it highly scalable.

Our proposed solution has a set of two nodes namely (i) regular nodes which do not pose any threat to the MANET. Regular nodes are responsible for exchanging routing information and sending or forwarding data packets to a destination on behalf of other nodes and perform actions regarded as normal in the network. The second type is a (ii) malicious node with a built-in attack mechanism aimed at causing undesirable effects in the network, this node drops all data packets but responds to all routing information exchange.

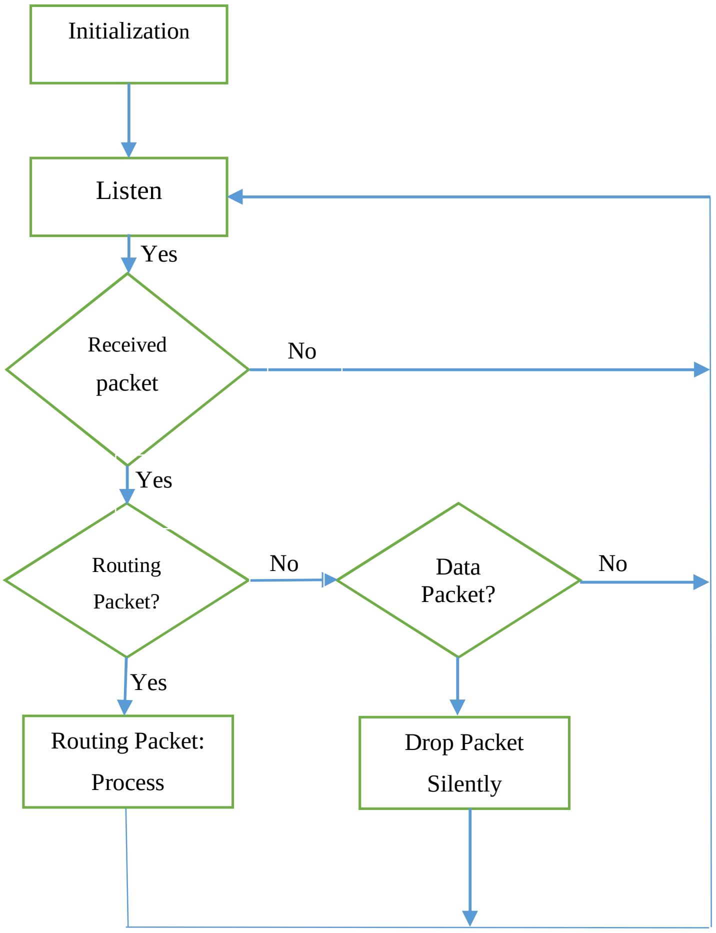


Figure 2: Malicious Node

After the data analyzer component has been trained properly to process any incoming network characteristic data, it will be used to determine if any malicious activity is taking place. If an undesirable activity are noted within the network, the nodes are identified and an appropriate action is taken. Main functionality of the “data collection” component is to record all the data relating to all events and actions taking place in the network such as sent and received packets, route errors and other useful data in the network. The collection component obtains its input from the local audit data traced on a specified trace file. Extraction of pertinent statistics from the packets and records of the occurrence of certain actions and events being sort after on a per flow basis follows is performed by the data analysis component. Information obtained by the data collection component is passed to the data analysis component for further processing.

In this technique we first simulate different attacks on the mobile ad hoc network and obtain different parameters specifically that which affect network characteristics, which we will then pass on as input into our detection /classifier component powered by ANN for training before later using it for the detection and classification of the attacks.

3.2 Data Collection Component

The detection technique given in this research consist of the data collection, data analyzer and detection / classifier components. These components form the core layer of our detection technique. Useful control information such as action taken and times of events on sent packets, dropped packets etc. is collected at the data collection component. Collected data is passed to the data analyzer component which extracts useful information .The detection component can then be applied to check for any deviation from normal behavior, so as to detect and classify any malicious activities.

Data Collection Component

```
Algorithm for Data Collection Component

   for;; do

   if    packet type   : = Cbr &
         Action        : = Dropped &
         Layer         : = Agent &
         Packet size   :> = 1000

     node id       : = id from node which dropped packet
     current time  : = get time at which event occurred
     Store the dropped packet in the store

       end if
end for
```

Figure 3: Algorithm 1

Data Analysis Component

```
Algorithm for Data Analysis Component
          for ;; do


if    packet type   : = Cbr &

      Action        : = Dropped &

      Layer         : = Agent &

      Packet size   :> = 1000


if no TCP timeout occurred then
  fid                   : = flow on which the timeout occurred
 node id                : = id from which the drop took place
  current time          : = get time at which event occurred
```

Figure 4: Algorithm 2

```
….continued

         if no route  error messages then

                  record as malicious activity

         end if

           else
                "activity not malicious, either it is a broken link or time out of
         TCP"

           end if
     end for
```

Figure 5: Algorithm 2 continued

3.3 Network Model

A multi-hop ad hoc network is assumed where nodes collaboratively relay traffic using AODV [50] routing protocol. The network consists of a set of N nodes. Any path P used to route traffic from a source S to a destination D is assumed to be known to S. Path P can be established and becomes known to S once the route discovery process has been completed. For simplicity, we number the nodes in P in ascending order, i.e., *ni* is upstream of *nj* if $i < j$:

All nodes are assumed to be working in promiscuous mode and collaborate in monitoring the Path P. Critical metrics can be periodically obtained from the local data collected from each node, such as throughput, packet dropped, packets forwarded and delay etc. among other important metrics. If a misbehaving node drops data packets and periodic updates as part of its misbehavior pattern, the analysis done on the collected data will interpret this as an occurrence of a misbehavior. This analysis is used to identify the misbehavior process and account for: (i) temporal variations of performance due to traffic or intermittent connectivity, and (ii) random behavioral patterns of the misbehaving nodes. Note that we can implicitly measure the throughput of end-to-end connection if TCP is used at the transport layer. The received end to end acknowledgements can be used to give an accurate estimation of the "instantaneous" throughput and of the round trip delay.

A further examination of the research literature shows that several solutions have been proposed to mitigate the problem of packet dropping attacks including two techniques (i) watchdog mechanism proposed in [10] and (ii) Cop mechanism proposed in [11]. However we will not dwell much on these techniques as we are motivating the use of a different technique discussed a little later in this section.

3.4 Adversarial Model

In our model we focus on detecting misbehaving nodes that participate in the routing paths establishment process and drop data packets afterwards, we will not attempt to detect misbehaving nodes that are unwilling to participate. Furthermore we will not concentrate on other types of misbehavior aimed at the routing process such as advertisement of false routing information, creation of sinkholes, blackhole, wormholes, etc. [19, 18, 46, 52, 39, 61, 27, 40, 53]. We are of the notion that such nodes would have isolated themselves from the network by not participating in the route establishment process and do not further degrade the network performance. Our model has a subset of nodes that behave maliciously and hence do this starting at some random time and from that time onwards it drops all the packets it receives. We also assume that:

- Packet dropping in the network is either due to malicious activity or due to broken link(s) at some intermediate node along path P.

- All nodes have enough memory to store information during the route establishment process.

3.5 Problem under Study

In this thesis the problem addressed is shown diagrammatically in Figure 1.1. A set of *n* nodes formulating an ad hoc network having a subset *m* of the nodes acting maliciously by dropping packets either continuously or selectively. A fraction *m* of the nodes deployed in the network is assumed to be misbehaving. This misbehavior is manifested by the dropping of transit traffic from a source to a destination. Misbehaving nodes can be continuous droppers, or adopt a selective dropping strategy. For a given path *P* of length *k*, we assume that a set of *m* misbehaving nodes, with |*m*|≤ k, exist along this path. These nodes can be located anywhere along *P*.

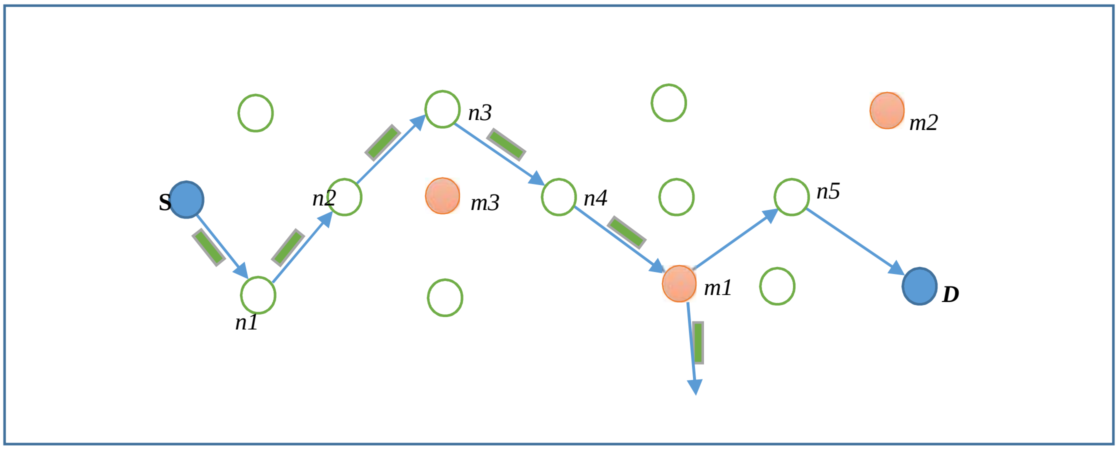


Figure 6: *m1* acting maliciously

Figure 3.5: Node S is sending traffic to D along *P*. Node *m1* drops all packets. Our goal is to identify *m1*, provide evidence of its misbehavior, and construct routes that avoid *m1*.

Our goal in this thesis is: (i) to detect the misbehaving nodes, and (ii) to able to classify them accordingly to attack types and construct routes that avoid the use of the misbehaving nodes. In achieving these two important goals, we seek to ensure resource-efficiency for the misbehavior identification and route discovery methods

The two algorithms presented above, can be understood in operational terms more clearly by looking at the figure above which gives an illustration of our problem statement scenario. In this example we show case how our proposed technique uses information that affect network characteristic data ,extracted from data collected from different network levels to detect malicious behavior in the network.

Our base routing protocol is AODV. In this example we try to ascertain the actual cause of the packet dropping activity, we therefore take into account, AODV route error messages and TCP timeout messages, to get a clear picture and be able to detect malicious packet dropping in the route from a source to some destination. AODV routing protocol takes care of broken links and intermittent links of a route by letting nodes send route error message back to the source if they are unable to deliver the packet to next hop neighbor. Broken communication can be due to node movement which could result in a new network topology, in this case old routes ceases to exist. At times the broken link will be temporal between the node and its next hop, and hence the node fails to find an alternate route in time to the destination.

A TCP timeout may also occur within the MANET, when sender does not get an acknowledgement within a specific time period. This may occur when a packet was dropped at an intermediate node due to (i) congestion at that node or (ii) the packet’s Time To Live (TTL) value had expired or (iii) by some malicious node trying to disrupt the network or (iv) due to a broken link at some intermediate node in the source route or the packet get corrupted during transmission and was dropped by an intermediate node. In this research we assume that there is no congestion in the network, hence we will not deal with any packet drops due to congestion. Our technique mainly utilizes control information passed between the source and the destination to detect the presence of malicious nodes in that route.

Figure 3.6 below shows an illustration of our network topology at a particular instance of time. Two nodes namely node S represents the source and D is the destination.

Assuming that AODV route discovery process agent in node S has found the route S-n1-n2-n3-n4-n5-n6-D to the destination D. In the scenario shown above there exists no malicious nodes in the path P to the destination D, hence everything is normal. Our data collection component will still be at work collecting local data that falls within the defined interval and at the same time, the data analyzer component will be extracting desired useful information out of that data. The absence of malicious activity in the network, will ultimately mean that our detection/classifier component will not falsely detect the presents of malicious activities, but will only silently monitor the status of the network.

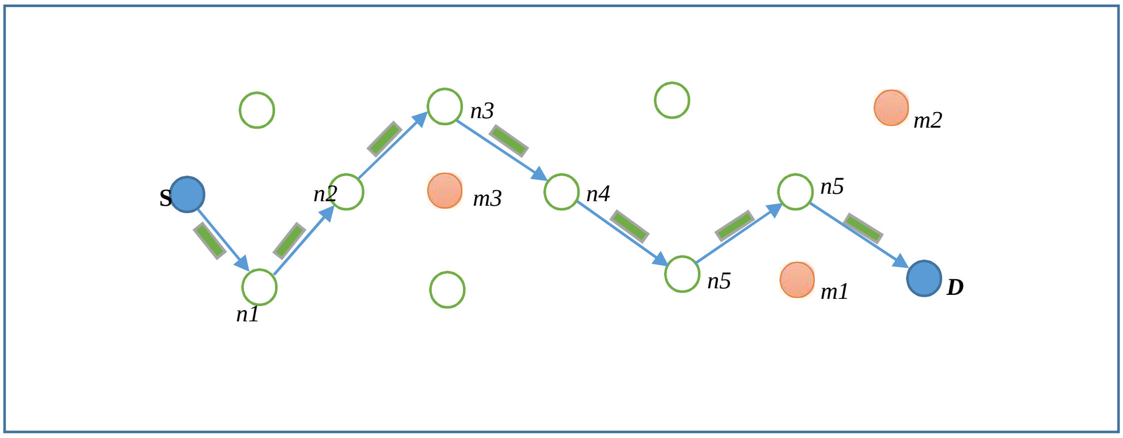


Figure 7: Normal Scenario desired

Now we depict a scenario under study, in which the communication between nodes S and D is disrupted at some random time, when a certain node m1 starts behaving maliciously by dropping the packets destined for D instead of forwarding them. Figure 3.4 below put into view this situation. Node *S* keeps sending out data packets, but node *m1* does not forward the packets to other intermediary nodes which lie in the path P to D. The detection / classifier component can then be useful in this case where there are undesirable activity within the network, for the purpose of identifying the actual causes of the broken links and to pick whether it is a result of malicious behavior.

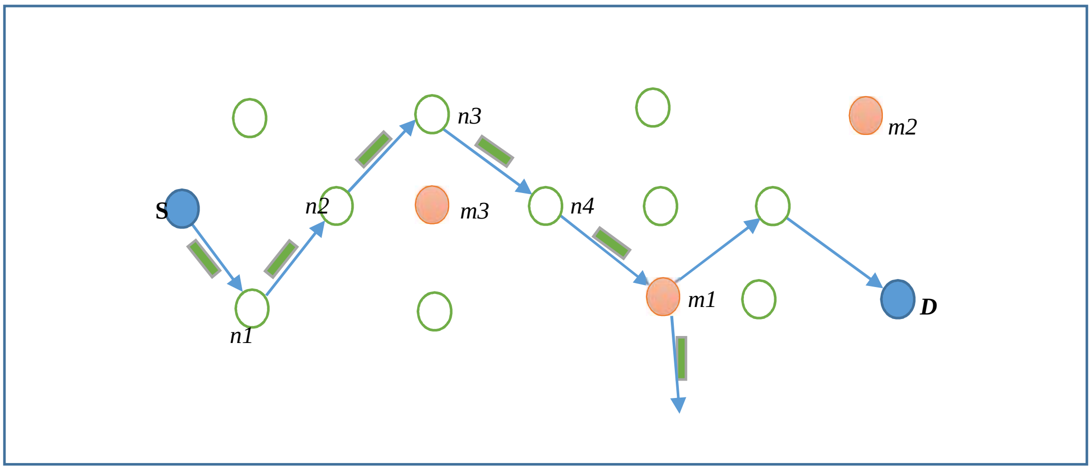


Figure 8: Need to identify *m1*

Our technique also aims at outsmarting some intelligent malicious nodes by focusing on dropped data packets rather than AODV route error message. Malicious nodes can at times send route error message to the source nodes for the packets they drop. In this case the source nodes will associate the TCP timeouts with the route error messages received, hence will fail to detect the malicious activity taking place in the network. We seek to also address malicious nodes that selectively drop data packets instead of dropping all of them. For example, if the detection /classifier component makes use of threshold value within a specified time interval, the malicious node could spread out its malicious drops so that the number of drops does not exceed the threshold within the time interval. Some malicious nodes collude with one another to launch sophisticated attacks without being detected. Colluding malicious nodes will not be taken into consideration in this current research, but will be studied as part of future research.

SIMULATION AND RESULTS

This section gives details of the tool used in conducting our research and experiments undertaken, we later on provide an analysis of the simulation results obtained. In the initial phase we implemented our technique in Network Simulator 2 (ns 2.35) [70] to simulate a MANET showing our problem as outlined in previous sections running the AODV routing protocol. Network Simulator version 2 is an object oriented, discrete-event driven network simulator written in C++ and OTcl. NS implements a variety of IP networks. NS is useful in implementation of network protocols such as TCP and UDP. Traffic source behavior such as FTP, Telnet, Web, CBR and VBR can also be implemented in NS. NS offers provisions for implementing router queue management mechanism such as Drop Tail, RED and CBQ, and routing algorithms such as reactive and proactive , and many more others [71], [72].

The present research work involves evaluation of the performance of our proposed system in detecting and classifying misbehaving nodes. We take into consideration cases where: (i) the misbehaving nodes continuously drop packets, and (ii) the misbehaving node selectively drop packets.

4.1 Simulation Environment

Numerous network simulators are available for use in various scenarios. Most popularly used for the purposes of network research include OMNeT++, Network Simulator 2 among a many as detailed as tabulated below.

Part of our work is simulated via Network Simulator 2.35. NS-2 has as its prime aim to support research in networking at various institutions undertaking networking research [70]. New protocols can be developed and traffic patterns can be studied in NS2.

NS-2 simulator is distributed as open source software, it was developed as a collaborative environment for networks. A considerable number of organizations and researchers throughout the world are making use of Network Simulator 2 for researching and experimental studies in networking ranging from protocols analysis to sophisticated technologies.

Simulator	NS-2	GloMoSim	OMNET
Online location	http://www.isi.edu/nsnam/ns/	http://pcl.cs.ucla.edu/projects/glomosim/	http://www.omnetpp.org/
Language	C++	C (PARSEC library)	C++
Authorization	Academic public permit	Academic public permit &	Academic public permit
Advantages	Extensive usage and support, also available at no cost	Accessible at no cost without restrictions	Accessible at no cost without restrictions
limitations	Challenging and tough to study for starters	Implements non object based.	Restricted Version

Table 2: Wireless Network Simulators

4.2 Overview of NS-2 simulator

Networking research has been improved through the use of simulation tools such as NS2. Network Simulator which is entity-based was coined and created at UC Berkeley using languages like (i) C++, (ii) OTcl that allow development of various prototypes in routing protocols for instance while resultantly providing trace support. Several protocols such as those running on TCP and UDP can be implemented. Various data generation techniques like file transfer protocol and constant bit rate are available for use in the simulator. Queue handling mechanisms for the router are supported for handling queuing needs of various devices in the simulation i.e. (i) Drop Tail, (ii) RED and (iii) CBQ. Other protocols for broadcasting and multicasting over wired and wireless networks are also implemented in NS.

NS began as a variation of the network simulator developed in 1989, since then a lot more revisions have been made during the past years. NS development was supported by the Defense Advanced Research Projects Agency (DARPA) under the auspices of the Virtual Inter Network Testbed (VINT) project in 1995, at Xerox Palo Alto Research Center (PARC) and at the Information Sciences Institute (USC/ISI) of the University of Southern California. Currently, NS-2 development is supported through DARPA with Simulation Augmented by Measurement and Analysis for Networks (SAMAN) and National Science Foundation (NSF) with Collaborative Simulation for Education and Research (CONSER), in collaboration with other researchers including the ICSI [115] (International Computer Science Institute) and Center for Internet Research (ICIR).

4.2.1 General Structure and Architecture of NS-2

NS-2 is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup (plumbing) module libraries. Creating network simulations and running them requires a user to write an OTcl script that initiates an event scheduler, sets up the network topology using the network objects and the plumbing functions in the library informing traffic sources to start and stop transmitting packets through the event scheduler.

An event scheduler is another major component of NS-2, whereas an event in NS has a packet ID that is unique for a packet .Each event is scheduled to take place at some given time, hence the event scheduler keeps track of simulation time and has a pointer to an object that handles the event. The event scheduler fires all the events in the queue of the event scheduler for the current time by triggering and initiating appropriate action associated with the packet pointed to by the event.

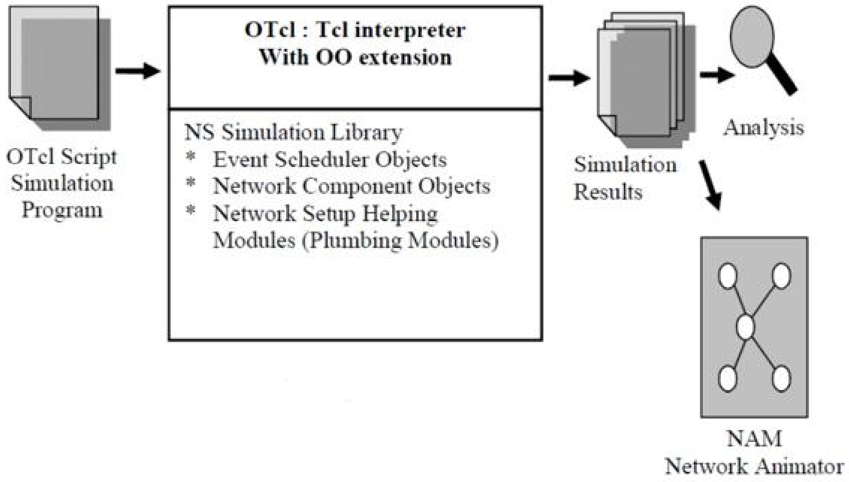

OTcl Script Simulation Program
OTcl : Tcl interpreter With OO extension
NS Simulation Library
* Event Scheduler Objects
* Network Component Objects
* Network Setup Helping Modules (Plumbing Modules)
Simulation Results
Analysis
NAM Network Animator

Figure 1: General Structure of NS

NS-2 is written not only in OTcl but in C++ also. They is separation of data path implementation from control path implementations in NS, this is so as to ensure efficient operation. Packet and event processing time is reduced in NS2 by way of writing and compiling the event scheduler and the basic network component objects in the data path in C++. The compiled objects are made available to the OTcl interpreter through an OTcl linkage that creates a matching OTcl object for each of the C++ objects and makes the control functions and the configurable variables specified by the C++ object acting as member functions and member variables of the corresponding OTcl object. In this way, the controls of the C++ objects are given to OTcl. It is also possible to add member functions and variables to a C++ linked OTcl object. The objects in C++ that do not need to be controlled in a simulation or internally used by another object do not need to be linked to OTcl. Similarly, an entity can be entirely implemented in OTcl as long as it is not in the data path. The figures below illustrates an object hierarchy in C++ and OTcl. Each C++ objects that have an OTcl linkage forming a hierarchy, there is a matching OTcl object hierarchy very similar to that of C++.

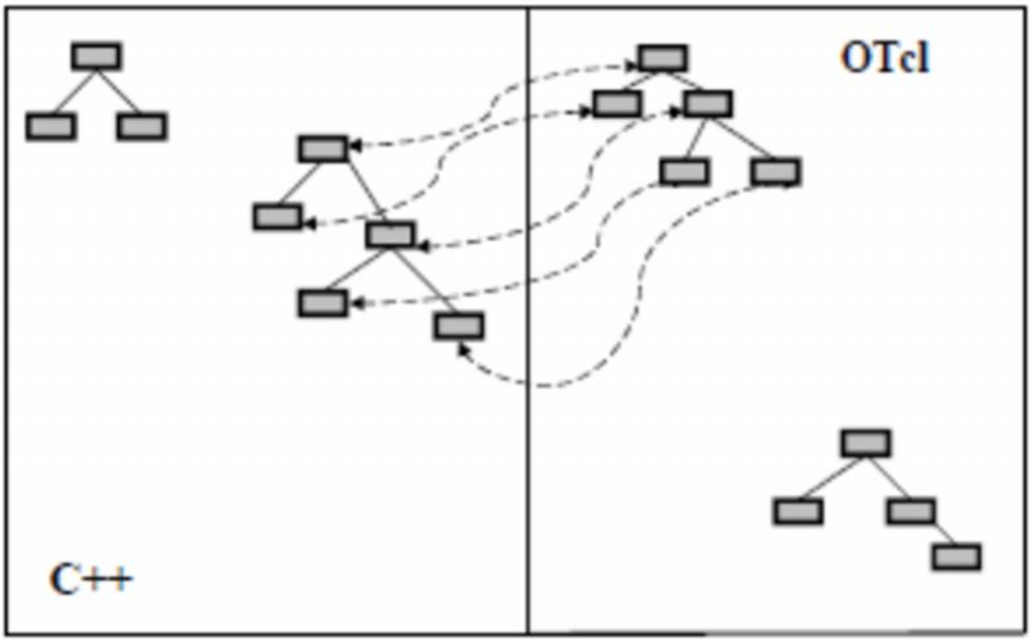


Figure 2: Architecture of NS-2

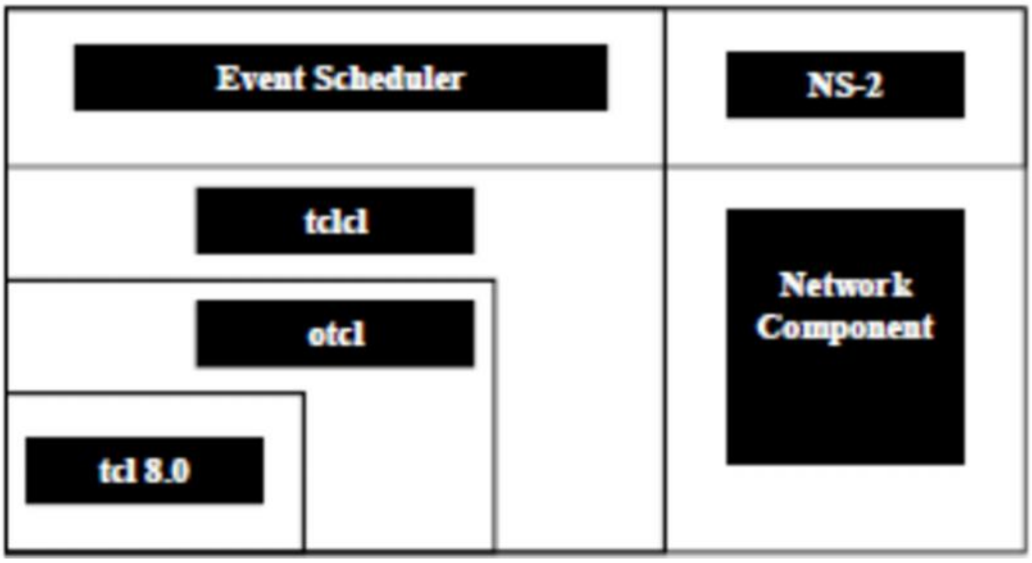


Figure 3: Architectural view of NS-2

4.3 Implementation: Phase 1

Our work is dived into two phases, in the first phases we model various attack scenarios including the normal expected scenario under a normal network operation in Network Simulator 2.In this phases we will obtain a number of parameters that shall serve as input to our second and final phase. Firstly we examine the simulation setup then go on to mobility models implemented as well as the traffic sources used.

4.3.1 Simulation setup

The simulation setup has a network with alternating number of nodes ranging from 25-50 nodes, in a flat topography of 670m X 670m in dimensions, adopted because of its popularity in usage in ad-hoc network researches and in most of the simulations done with ns-2. Simulation time of 200 seconds was allocated to run each of the simulations. The highest speed achieved by the nodes was given as 20 meters/second in our node movement model to simulate a highly mobile network. For the purpose of simulating low to medium mobility networks 5 meters/second was chosen as the speed. Pause times for the nodes were set to 5 seconds for highly mobile networks and 20 seconds to simulate low to medium mobility networks.

Our implementation will initially start by choosing of an appropriate configuration for our desired scenarios in the simulation, we further are required to configure the simulation with the appropriate parameters. The configuration used in this thesis for the purpose of our network simulation is as depicted below

4.3.2 Simulation Parameters

P.id	Parameters	Value
i.	Number of nodes	25-50
ii.	Simulation Time	200sec.
iii.	Area size	670*670m^2
iv.	Maximum Speed	20 m/s
v.	Traffic Source	CBR
vi.	Pause Time in seconds	5,10,20
vii.	Packet Size	512 Bytes
viii.	Rate of churning out packets	4 Packets/s
ix.	Maximum number of connections	10,20,30
x.	Bandwidth	10Mbps
xi.	Delay	10 ms
xii.	Mobility model used	Random way point

Table 1: Simulation Parameters

4.3.2 Mobility Model

Mobility models are a way to represent the movement of mobile nodes, and how their location, velocity and acceleration change over time. We adopt the random way model for our network simulation for the purposes of evaluating our technique.

In our simulations, the Random way-point mobility model was used to evaluate our detection technique. Random way-point model will enable the nodes participating in our network to move from one location to a new location by randomly choosing a direction and speed in which to travel, pausing between the changes in direction and/or speed. Random Way-Point is a memory less mobility pattern, it maintains no information regarding its past positions and speed values. In Random way-point model speed and direction of a node is independent of its past speed and direction. Sometimes the features inherent in the Random Way-Point can cause impractical movements such as abrupt stops and sharp turns [30].

Each node movement in our simulation is extracted from the movement scenario used to generate the node movement file. In each of the files storing these movement patterns used for each of our simulations, pause time is exclusively defined. In conducting our simulation we chose the movement patterns generated for different pause times so as to evaluate the usefulness of our technique under high, medium and low mobility patterns. We used pause time of 0-5 seconds in order to have corresponding continuous motion, and a pause time of just over 5 seconds to almost 15seconds as corresponding to medium mobility and lastly 20 seconds and over corresponds to low mobility and to almost no motion. In order to create a set of the movement scenario files for the different mobility models we used the ‘setdest’ program of NS-2, and in that we varied our pause time. Our node-movement files using the ‘Random Waypoint Algorithm’ were generated using the ‘setdest’ program.

4.3.4 Generation of Node Movement

In generating the node movements for our MANET we used a tool called ‘setdest’ developed by Carnegie Mellon University. It helped us to define the node movements, particularly specifying speed of the nodes towards a random or specified location within an area we defined. Upon arrival at the location the node could stop for a period of time depending on the pause time. After the brief stop the node will move to the next location. To randomly generate the movements of the mobile nodes in the simulation, we used (4.1) the below command:

```
. /setdest [-v version] [-n num_of_nodes] [-p pause_time] [-M max_speed] [-t sim_time] [-x max_ x] [-y max_y] > [outdir/ movement-file] > movement-file.
```

For the desired pause times of 5, 10 and 20 seconds respectively, the corresponding mobility files are generated in the directory “home/imap/ns-allinone/ns-2.35/indep-utils/cmu-scen-gen/setdest/” as follows-

Option	Interpretation
-v version	Simulator version
-n number	total number of nodes in the scenario
-p pause_time	Period a node stays motionless after it arrive a location
-M max_speed	Maximum moving speed of nodes.
-t sim_time	Simulation time.
-x max_x	Maximum length of the area.
-y max_x	Maximum width of the area.

Table 2: node movement options

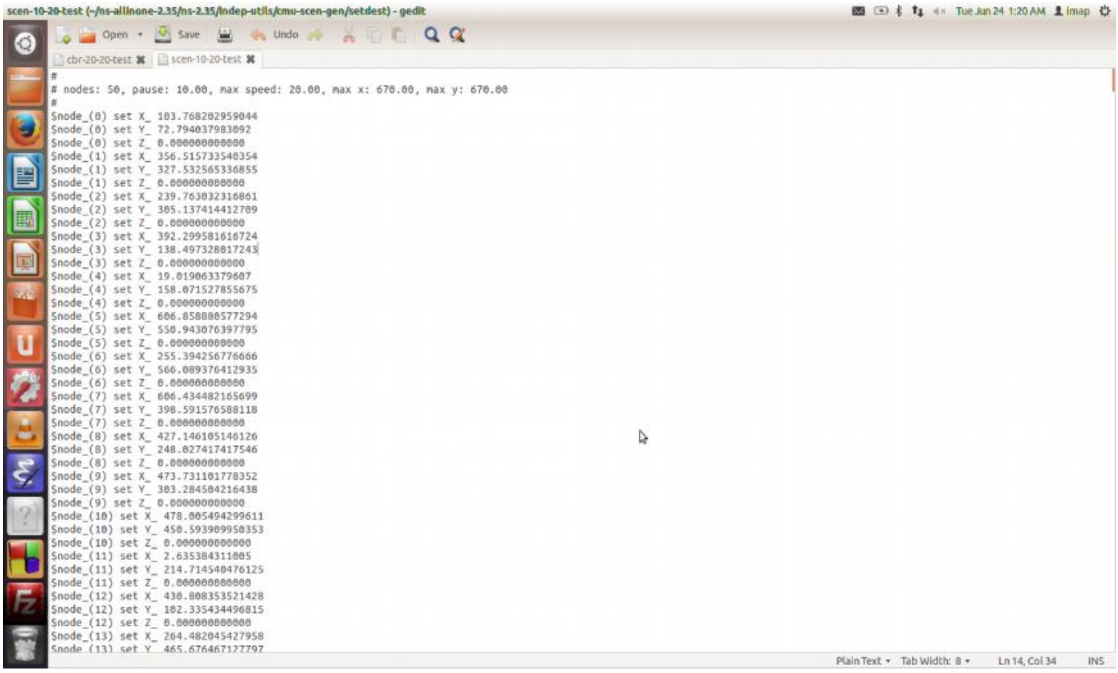
```
./setdest - v 1 -n 50 –p 5 –M 20 –t 200 –x 670 –y 670>scen-5-20-test
./setdest - v 1 -n 50 –p10 –M 20 –t 200 –x 670 –y 670>scen-10-20-test
./setdest - v 1 -n 50 –p20 –M 20 –t 200 –x 670 –y 670>scen-20-20-test
```

A sample set of file (4.2) with a pause time of 5 seconds and 50 nodes is given. For a different pause time a new file with the corresponding pause time is used .In this case three files representing pause time of 5, 10 and 20 seconds is used all working on the same routing protocol. Figure 4.4 below represent the mobility of pause time 5seconds.

4.3.5 Communication patterns

We chose to set up a random traffic connections making use of CBR between the mobile nodes using traffic-scenario generator script present in NS. We wrote a Tcl that invokes “cbrgen” to generate random flows of traffic. Cbrgen helped us to generate the traffic load, specifically using CBR. The following command was used along with the “cbrgen.tcl” program (4.2) with options shown in the Table 4.2 below

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate
rate] > traffic-file
```



```
#
# nodes: 50, pause: 10.00, max speed: 20.00, max x: 670.00, max y: 670.00
#
$node_(0) set X_ 103.768202959044
$node_(0) set Y_ 72.794037983092
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 356.515733540354
$node_(1) set Y_ 327.532565336855
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 239.763032316861
$node_(2) set Y_ 305.137414412709
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 392.299581616724
$node_(3) set Y_ 138.497328017243
$node_(3) set Z_ 0.000000000000
$node_(4) set X_ 19.019063379607
$node_(4) set Y_ 158.071527855675
$node_(4) set Z_ 0.000000000000
$node_(5) set X_ 606.858880577294
$node_(5) set Y_ 550.943076397795
$node_(5) set Z_ 0.000000000000
$node_(6) set X_ 255.394256776666
$node_(6) set Y_ 566.089376412935
$node_(6) set Z_ 0.000000000000
$node_(7) set X_ 606.434482165699
$node_(7) set Y_ 398.591576588118
$node_(7) set Z_ 0.000000000000
$node_(8) set X_ 427.146105146126
$node_(8) set Y_ 248.027417417546
$node_(8) set Z_ 0.000000000000
$node_(9) set X_ 473.731101778352
$node_(9) set Y_ 303.284504216438
$node_(9) set Z_ 0.000000000000
$node_(10) set X_ 478.005494299611
$node_(10) set Y_ 450.593909950353
$node_(10) set Z_ 0.000000000000
$node_(11) set X_ 2.635384311005
$node_(11) set Y_ 214.714540476125
$node_(11) set Z_ 0.000000000000
$node_(12) set X_ 430.808353521428
$node_(12) set Y_ 102.335434496815
$node_(12) set Z_ 0.000000000000
$node_(13) set X_ 264.482045427958
$node_(13) set Y_ 465.676467127797
```

Figure 4: Node movement file

It also worth to not that the data rate using CBR can be calculated using the formula given below:-

Data Rate (bits/second) = 512 bytes*8 bits/bytes * rate (packets/second defined in "cbrgen")

Command Options	Interpretation
-type cbr|tcp	Type of traffic, TCP or CBR
-nn nodes	Total nodes
-seed seed	Random seed value
-mc connections	maximum connections
-rate rate	packets per second

Table 3: traffic movement parameters

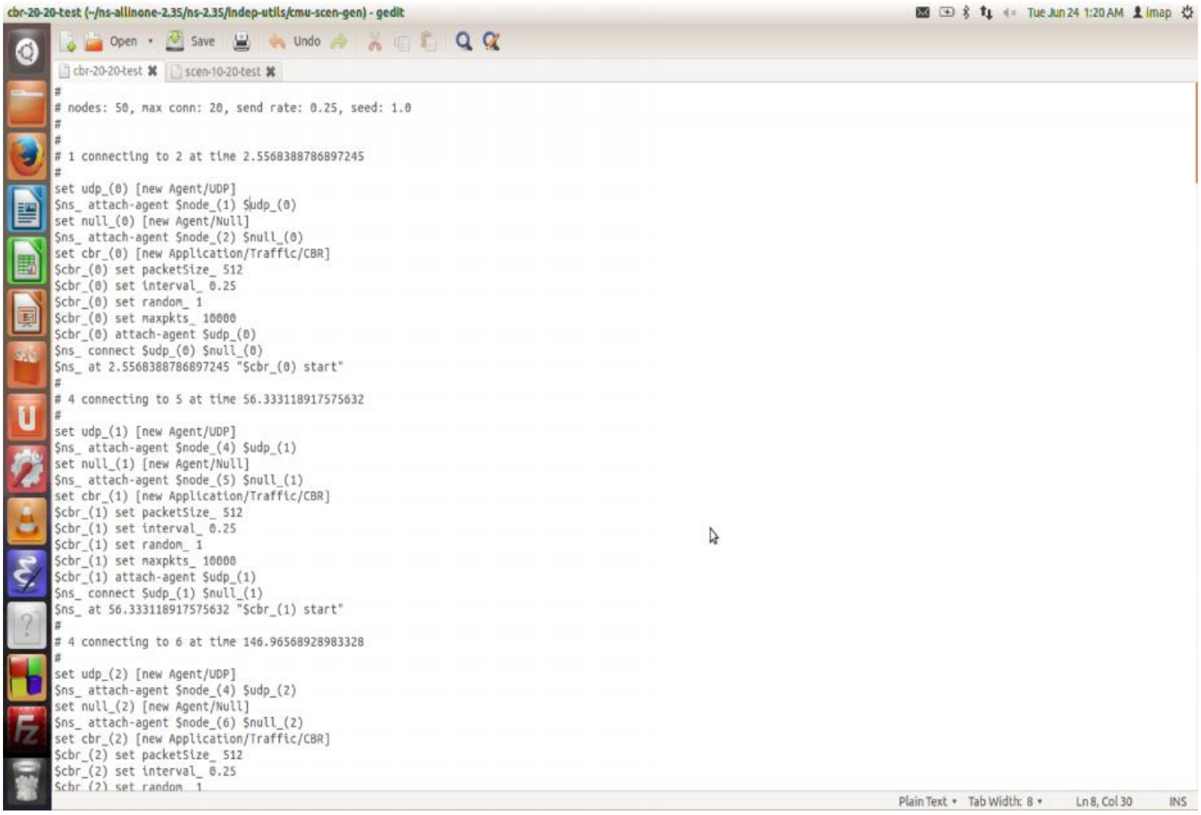
4.3.6 Generation of the traffic file

To create a CBR connection file amongst our simulated 50 nodes, having maximum of 10 connections, with a seed value of 1.0 and a rate of 4.0, we used:

ns cbrgen.tcl -type cbr -nn 50 -seed 1.0 -mc 10 -rate 4.0 > cbr-20-10-test

An extract from the generated cbr-20-10-test file is given below

A Traffic Load Generation File (cbr-20-10-test)



```
#
# nodes: 50, max conn: 20, send rate: 0.25, seed: 1.0
#
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.25
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
# 4 connecting to 5 at time 56.333118917575632
#
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 0.25
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 56.333118917575632 "$cbr_(1) start"
#
# 4 connecting to 6 at time 146.96568928983328
#
set udp_(2) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(2)
set null_(2) [new Agent/Null]
$ns_ attach-agent $node_(6) $null_(2)
set cbr_(2) [new Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set interval_ 0.25
$cbr_(2) set random_ 1
```

Figure 5: traffic movement file

4.3.7 Attack emulation representing misbehaving Nodes

In order to represent our adversary model, we modified the ns-2 simulator to enable particular node(s) to be configured as malicious. A fraction of the nodes within the network was configured to misbehave. The misbehaving nodes independently implemented a packet dropping strategy, either continuously or selectively. The packet dropping starts at some specified time at which nodes designated as malicious start behaving maliciously. At some scheduled time, the designated malicious nodes drops all the packets, these packets are exclusively non-control packets that are received at that node till the end of the simulation. Each network is planned to contain 3 malicious nodes reflecting misbehavior of 6% of the nodes .The malicious nodes will surely be part of the path P, an active path to the desired destination. To determine the effectiveness of our approach, the percentage of misbehaving nodes was varied from 0% to 6%.

The attack emulation was provisioned by a way of modifying AODV routing protocol, this provided us with a means to test the security aspect of the MANET system under study, running in NS. The interactions of each node with the packets results in the malicious node effecting the attack, thereby dropping the packets. Emulating the attacks for the purpose of investigating the effectiveness of our technique involved

a. Capturing and intercepting incoming and outgoing packets achieved by using the pcap library, which is part of NS for primarily capturing of network packets, including both data packets and routing messages.
b. Overhearing traffic in neighboring nodes – achieved by placing the wireless interface in promiscuous mode to monitor traffic in its proximity.

In the TCL script the respective malicious node are defined and work with the C++ file to operationalize the attack.

```
#==============================================================
#    A Set of misbehaving nodes as outlined in the tcl file
#==============================================================
$ns at 0.0 "[$n47 set mal_] m1"
$ns at 0.0 "[$n42 set mal_] m2"
$ns at 0.0 "[$n23 set mal _] m3"
$ns at 0.0 "[$n12 set mal _] m4"
$ns at 0.0 "[$n19 set mal _] m5"
$ns at 0.0 "[$n29 set mal _] m6"
```

4.3.8 Metrics Extraction

Initially we used OTcl script language to initiate an event scheduler, create and set up the network topology using the network objects and inform traffic sources to start and stop transmitting packets with the help of the event scheduler. Our OTcl script will be executed by NS-2, giving us simulation results in the form of text based output files and an input to a graphical simulation display tool called Network Animator (NAM).We will analyze the text based files that carry records and activities taking place in the network. We made use of “Perl” and “Awk” tools for evaluating and extracting some desired metrics as well as some needed results. First we show an extract of the trace file, which is a text based result file obtained at the simulation end, from which we will make analysis to obtain some desired metrics.

4.3.8.1 Network animator

The simulation results from running the script in NS-2 include one or more text based output files and an input to a graphical simulation display tool called Network Animator (NAM).

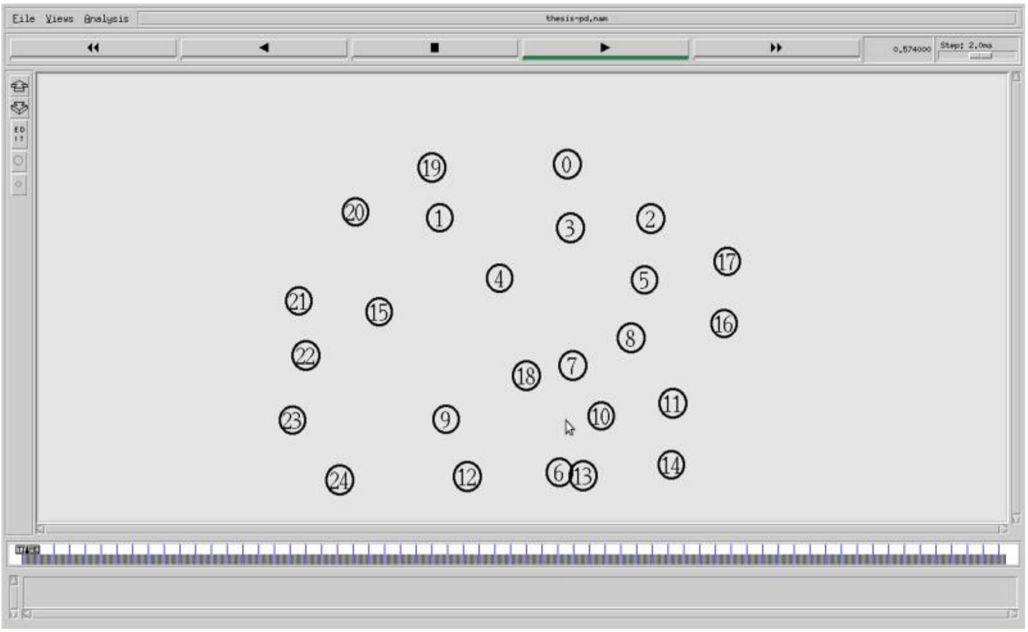


Figure 6: Network topology with initial 25 nodes

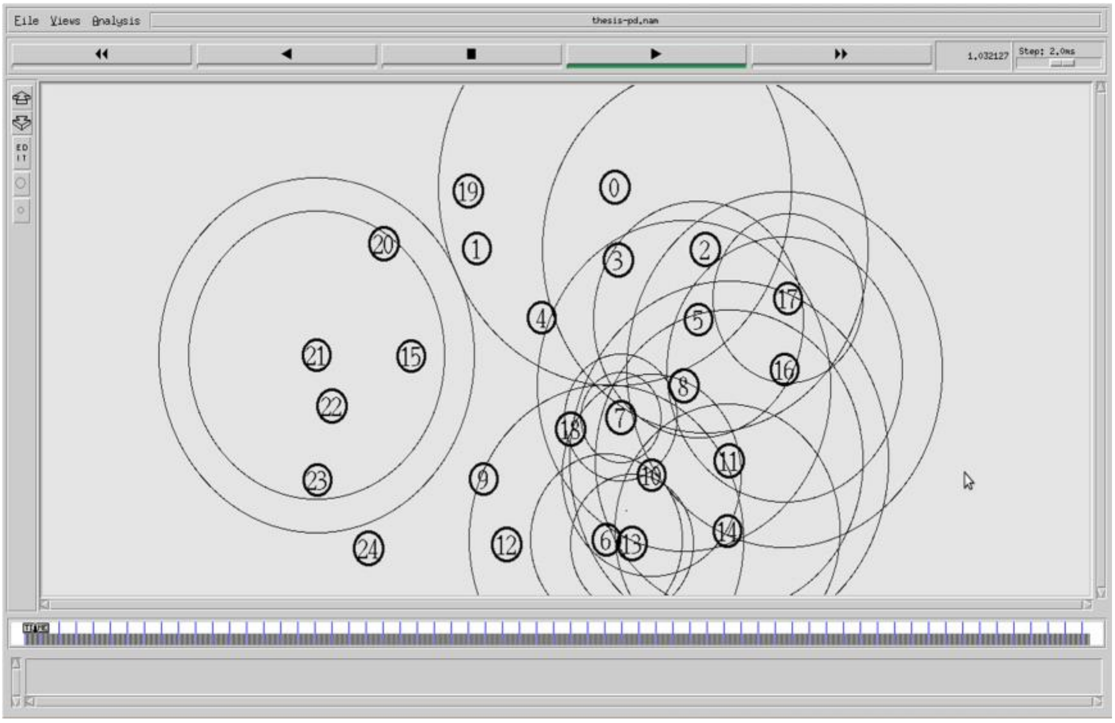


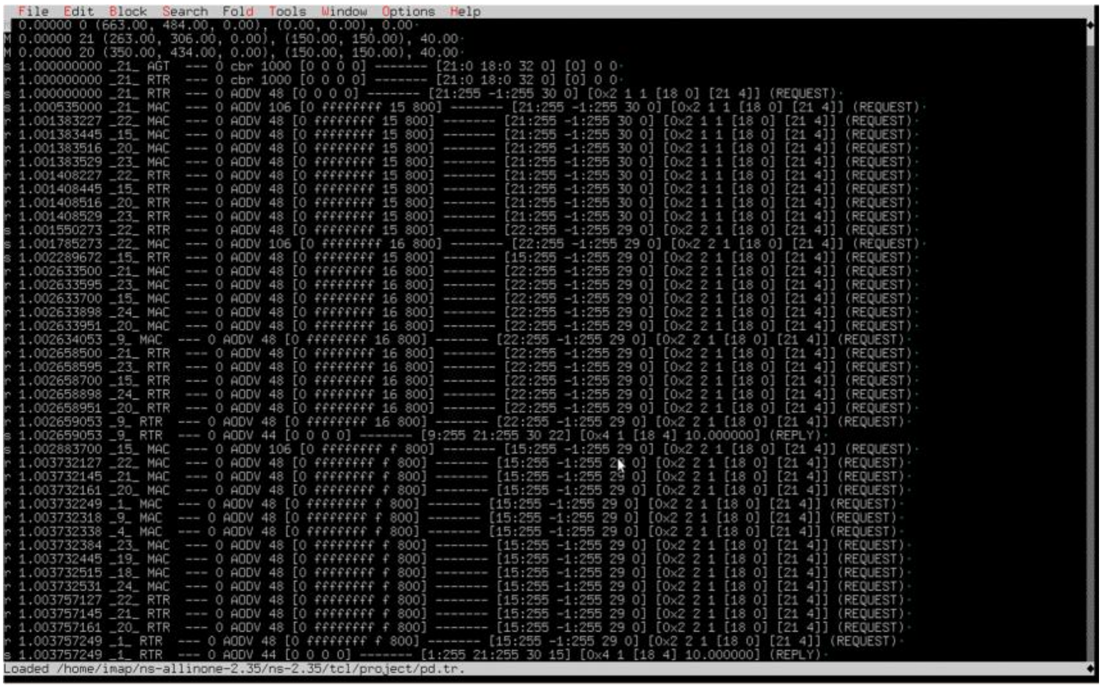
Figure 7: Nodes in transmission action with their hearing ranges

4.3.8.2 Trace File

Trace file is a text-based result file we get from the simulation. It contains a detailed record of actions and other important facts of the simulation. The trace file helps us obtain information pertaining to actions of different layers in the network. Included in the tracing are agent, router, MAC and movement activities. Figure 4.6 shows an example trace file in a screenshot below.

4.3.8.3 Performance evaluation parameters and metrics

i. RREQ Sent
ii. RREQ Replies
iii. Route Error Messages
iv. Packet Data Sent
v. Packet Data Received
vi. Packet Data Dropped



```
M 0.00000 0 (663.00, 484.00, 0.00), (0.00, 0.00), 0.00
M 0.00000 21 (263.00, 306.00, 0.00), (150.00, 150.00), 40.00
M 0.00000 20 (350.00, 434.00, 0.00), (150.00, 150.00), 40.00
s 1.000000000 _21_ AGT  --- 0 cbr 1000 [0 0 0 0] ------- [21:0 18:0 32 0] [0] 0 0
r 1.000000000 _21_ RTR  --- 0 cbr 1000 [0 0 0 0] ------- [21:0 18:0 32 0] [0] 0 0
s 1.000000000 _21_ RTR  --- 0 AODV 48 [0 0 0 0] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
s 1.000535000 _21_ MAC  --- 0 AODV 106 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
r 1.001383227 _22_ MAC  --- 0 AODV 48 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
r 1.001383445 _15_ MAC  --- 0 AODV 48 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
r 1.001383516 _20_ MAC  --- 0 AODV 48 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
r 1.001383529 _23_ MAC  --- 0 AODV 48 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
r 1.001408227 _22_ RTR  --- 0 AODV 48 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
r 1.001408445 _15_ RTR  --- 0 AODV 48 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
r 1.001408516 _20_ RTR  --- 0 AODV 48 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
r 1.001408529 _23_ RTR  --- 0 AODV 48 [0 ffffffff 15 800] ------- [21:255 -1:255 30 0] [0x2 1 1 [18 0] [21 4]] (REQUEST)
s 1.001550273 _22_ RTR  --- 0 AODV 48 [0 ffffffff 15 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
s 1.001785273 _22_ MAC  --- 0 AODV 106 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
s 1.002289672 _15_ RTR  --- 0 AODV 48 [0 ffffffff 15 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002633500 _21_ MAC  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002633595 _23_ MAC  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002633700 _15_ MAC  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002633898 _24_ MAC  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002633951 _20_ MAC  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002634053 _9_ MAC  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002658500 _21_ RTR  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002658595 _23_ RTR  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002658700 _15_ RTR  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002658898 _24_ RTR  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002658951 _20_ RTR  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.002659053 _9_ RTR  --- 0 AODV 48 [0 ffffffff 16 800] ------- [22:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
s 1.002659053 _9_ RTR  --- 0 AODV 44 [0 0 0 0] ------- [9:255 21:255 30 22] [0x4 1 [18 4] 10.000000] (REPLY)
s 1.002883700 _15_ MAC  --- 0 AODV 106 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732127 _22_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732145 _21_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732161 _20_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732249 _1_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732318 _9_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732338 _4_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732384 _23_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732445 _19_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732515 _18_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003732531 _24_ MAC  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003757127 _22_ RTR  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003757145 _21_ RTR  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003757161 _20_ RTR  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
r 1.003757249 _1_ RTR  --- 0 AODV 48 [0 ffffffff f 800] ------- [15:255 -1:255 29 0] [0x2 2 1 [18 0] [21 4]] (REQUEST)
s 1.003757249 _1_ RTR  --- 0 AODV 44 [0 0 0 0] ------- [1:255 21:255 30 15] [0x4 1 [18 4] 10.000000] (REPLY)
Loaded /home/imap/ns-allinone-2.35/ns-2.35/tcl/project/pd.tr.
```

Figure 8: Trace file after the initial simulation in the network

So to extract the following important information from a trace file like the one given above we wrote a program using ‘Perl’ in order to help our analysis component extract the desired information.

4.4 Implementation: Phase 2

4.4.1 Training and Validation Method

This section details an implementation of a Multilayer layer Perceptron-NN done in MATLAB^{TM} using the Neural Network Toolbox [16], for the purpose of detection and classification. We constructed the MLP-NN with the desired neurons per each layer, with a desired activation function also. The number of iterations for conducting each training set that yield were able to specify the number of layers, number of neurons in each layer, activation functions of neurons in different layers, and number of training epochs. Then the training data (feature vectors) and the corresponding target or desired outputs can be fed to the neural network to begin our training. The implemented neural networks had 6 input neurons (equal to the dimension of the feature vector) and three output neurons equal to the number of classes desired. Number of the hidden layers and neurons in each were parameters used for the optimization of the architecture of the neural network. Error back-propagation algorithm was used for training.

The efficiency of the NN depends on the training data. The collecting of data for training is a critical problem. This can be obtained several ways as including by using real traffic and by using simulated traffic. In our case we used simulated traffic to arrive at our data sets, which we then divided into three subsets. The first subset is the training set, which is used for training and updating the ANN parameters. The second subset is the validation set.

Sample Collected data for a Normal network scenario from our simulation (each row shows the value of one of the selected parameters) can be shown below

RREQ Sent	224	196	206	219	190	202	170	188	240	196
RREQ Replies	36	51	49	61	34	32	36	48	60	62
Route Error Messages	0	2	2	1	0	0	1	2	2	1
Packet Data Sent	1241	1241	1241	1241	1241	1241	1241	1241	1241	1241
Packet Data Received	1240	1192	992	1020	900	962	980	1021	1011	960
Packet Data Dropped	0	56	249	221	139	341	279	261	220	230

Table 4: Sample Data

In this phase we will make use of the parameters we obtained in phase one above as our intrusion detection evaluation data set with an extract shown table above. The sample version of the dataset included 6000 records. A subset of the data that contained the desired attack types and a reasonable number of normal events were selected manually. The final dataset used in this study included 2500 records. These parameters will be trained using a feed forward neural network, which shall then work as our detection and classification engine.

The inaccuracy on the validation set is monitored during the training process. The validation error will normally decrease during the initial phase of training similar to the training set error. However, when the ANN begins to over-fit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights that produced the minimum error on the validation set are retrieved [12]. In the present study, this training-validation strategy was used in order to maximize the generalization capability of the ANN.

There are at least four different chosen different categories of packet dropping attacks in our MANET environment including selfish, sleep deprivation attack, wormhole attack and rushing attacks among a many other attacks. Table 1 shows detailed information about the number of records from normal and three attack types included in training, validation, and testing sets. The implemented intrusion detector was a three layer MLP (two hidden layers with 6 neurons in each).This structure is referred to as: {6 6 6 3}

Record types	Training set	Validation set	Test set
Desired / Normal	2500	500	1000
Selfish packet droppers	2500	500	1000
Blackhole	2500	500	1000
Sleep deprivation	2500	500	1000

Table 5: Distribution of data vectors

This research was aimed to detect the malicious packet droppers and solve the multi class problem brought about by several attacks. Our scenario has a categorization which detail a set of different attacks, nevertheless open for adding more classes. Our output layer gives a variety of outputs which we shall term outputs states representing various classes, a good example will be to suppose that state [0] represents normal desired scenario, while [1] represents a malicious attack known as sleep deprivation and another state [2] representing packet dropper.

Figure below shows the training performance after 5 epochs using a Multi-layer perceptron with 2 layers, where both our layers are configured to use **PURELIN** activation function.

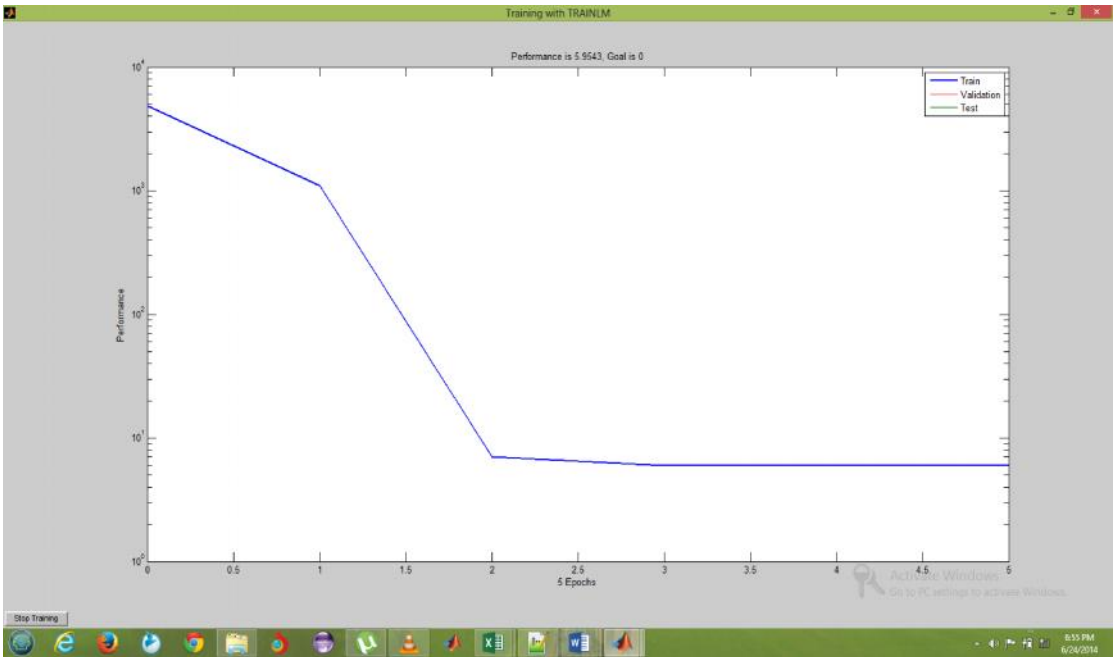


Figure 9: Training performance

To be able to give a credible analysis and reach an informed true judgment, we make use of analytical and investigative models that seeks to check performance of our technique with regard to the aims of this research. Our technique which seeks to provision a security service by detecting malicious behavior within a MANET comes with a cost, but as at this time our major concern is to ascertain its effectiveness towards detecting malicious packet droppers within the network. This will describe how effective our technique can be in decreasing the amount of loss incurred by an organization because of a certain attack.

4.4.2 Detection Effectiveness and Accuracy

Percentage of Dropped Packets

We compute the ratio of dropped packets calculated over all source/destination pairs every 100 epochs given as

$$PD = \frac{\#\, packets\ dropped}{\#\, packets\ sent\ by\ sources} \times 100\%$$

Where $\#\,packets\ dropped = \#packets\ sent - \#packets\ recieved$

We measured the percentage of dropped packets when misbehaving nodes were dropping the packets. We also varied the number of nodes misbehaving in the network with capabilities to drop packets haphazardly. We went on to evaluate the performance of our detection engine as measured by the detection effectiveness derived as:

$$DE = \frac{\#\,malicious\ nodes\ detected}{\#\,malicious\ nodes\ present} \times 100\%$$

As can be observed in the diagram below were we can have a visual of our actual detected malicious nodes versus the malicious nodes present in the network at one given time.

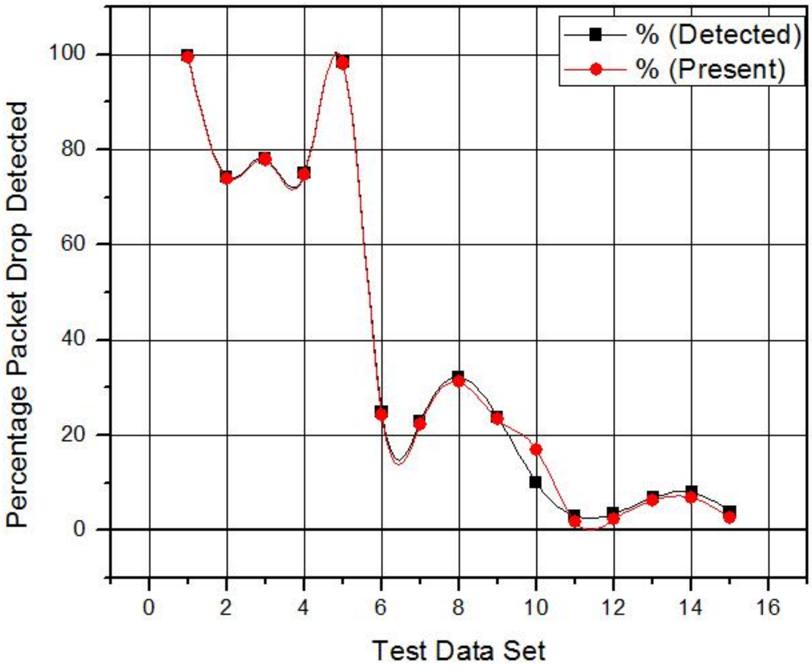


Figure 10: Detection effectiveness

4.4.3 Classification

For the purpose of classification tasks in our system, we used the metrics **true positives (TP)**, **true negatives (TN)**, **false positives (FP)**, and **false negative (FN)** in helping us to compare how effectively our classifier classified the results under test against our expected output. Our detection /classifier engine will be evaluated according to the estimation it will do, to evaluate whether it is close enough to the expected solution or not. Based on the prediction outcome a judgment can be arrived at whether it is TP, FP, FN or TN. For example if we define our investigation from **the above cases** for an approximate situation. Our desired outputs can be better represented by a 2×2 *confusion matrix*, given below:

Test Outcome (T.O)	T.O Positive	TP	FP	Precision = $\frac{\sum true\ posittive}{\sum Test\ outcome\ ppositive}$
		FN	TN	predictive rule (Negative) = $\frac{\sum true\ negaitive}{\sum Test\ outcome\ negative}$
	T.O Negative	Sensitivity = ty = Specificity = $\frac{\sum true\ positive}{\sum Condition\ positive}$ $\frac{sitive}{positive}$ $\frac{\sum true\ negative}{\sum Condition\ negati}$	Specificity = ty = Accuracy = $\frac{\sum true\ negative}{\sum Condition\ negativenegative}$ $\frac{gative}{}$ $\frac{\sum true\ positive + \sum tru}{\sum Condition\ nega}$	Accuracy = :y = $\frac{\sum true\ positive + \sum tru\ true\ negative}{\sum Condition\ negcnegative}$

Table 6: Confusion matrix

Where our Precision can be defined as

$$precison = \frac{tp}{tp + tf}$$

While our Recall can be stated and presented as:

$$recall = \frac{tp}{tp + fn}$$

Recall represents true positive value in this analysis. Precision can be termed as the **positive predictive value** (PPV).Some of the formulas that have been used to arrive at our analysis include these given below.

$$true\ negative\ rate = \frac{tn}{tp + tf}$$

$$true\ positive\ value = \frac{tp}{tp + fn}$$

$$true\ positive\ value = \frac{fn}{tp + fn}$$

$$Accuracy = \frac{tp + tn}{tp + tn + tp + fn}$$

False-alarm rates that is, (i) false-positive (FP) and (ii) false-negative (FN) will also be considered to give a comprehensive analysis of the effectiveness of our detection technique. A false-positive is defined as a normal node mistakenly classified as a malicious node, and a false-negative means that an actual misbehaving node fails to be detected. Recall that the detection score is in the interval [0, 1]. The detection threshold was set to 0.5 in our evaluation to strike a balance between FP and FN rates, and this parameter is configurable. There is always a tradeoff between FP and FN rates. A lower threshold can be set if FNs are a concern, while a higher threshold is required if FPs are less desirable .We tried to ascertain the detection effectiveness of our technique in light of predicting correctly the presence . The diagram below detail the observed FPR and FNR in light to our detection technique

Scenario	**TPR**	**TNR**	**FPR**	**FNR**	**Accuracy**	**Precision**
Normal	1	0.76	0	0.2221	1	1
12% Selfish packet droppers	1	0.74	0	0.111	1	1
24% Selfish packet dropper	0.56	0.8	0.0047	0	1	0.857
Sleep deprivation	1	0.716	0.0878	0	0.92	1

Table 7: FPR

It can be noted from the diagram below that our classification engine on the test data provided yielded good results, with all desired classification falling in their desired classes. Differences can be observed from our graph, but the system was able to classify the attacks as objectively.

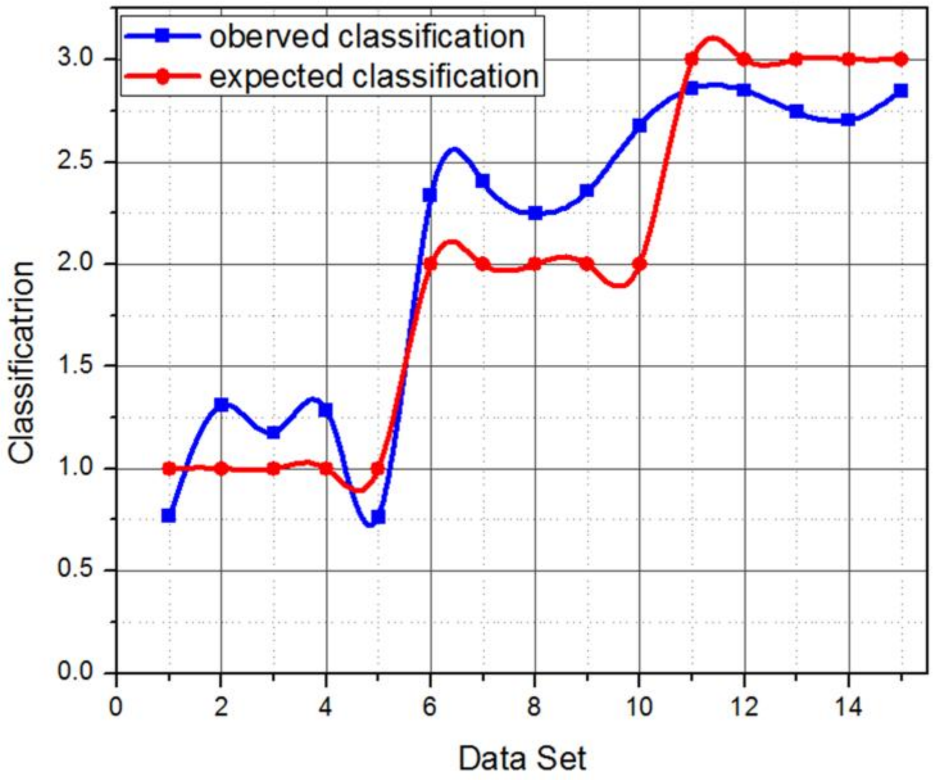


Figure 11: Classification effectiveness

CONCLUSION AND FUTUREWORK

This chapter summarizes our research work, analyses our contribution, and deliberates on our future work.

5.1 Conclusion

Some features inherent in Mobile ad hoc networks (MANET) makes them susceptible to several attacks such as free access for everyone to the wireless medium, an ever-changing topology, distributed collaboration, and limitations in their abilities like memory and power. MANET’s security can be aided by Intrusion Detection Systems (IDS) which can act as a second line of defense that is crucial to the overall security of the Network. Nevertheless, designing an IDS in MANET is a daunting task. Most wired IDS in the market today lack an approach that is distributive. Core to our research problem is the design of a scheme that is a scalable approach of intrusion detection for a MANET. This research utilise MANET as the core assessment platform.

In principle our research provide solutions to some of the most sort questions like the ones below:

i. What type of attacks have to be taken into account? A study carried out looked for intrusions that led to various attacks which were appropriate for detection approaches.

ii. How to design the adversarial model to represent the required attacks? For that programming in C++ and OTCL was done.

iii. What parameters and metrics are ideal in constructing a model close enough to the problem under study? In our methodology, we focused on network characteristics that that were affected by the presence of an attack and extracted those as our input before and after the adversarial model was effected in the system.

iv. How to extract the required parameters and metrics? We made use of a several tools including Perl scripts and Awk scripts.

v. How to construct a detection and classification engine for the purpose of identifying misbehavior or attacks in the network? We combined the capabilities found in NS2.35 and Matlab platforms

To evaluate the detection and classification effectiveness of our technique we carried out a number of simulations. The detection / classification technique was evaluated on the basis of verifying false positives, accuracy and false negatives. Our work is done in two phases, firstly with the major part carried in NS2.35 while the remaining part is based on the Matlab simulation platform.

5.2 Future Work

Ongoing researches have not dedicated much time to this area of securing MANETs by a way of intrusion detection systems. The research undertaken in this thesis detail our preliminary work on IDS in MANETs. Many notable exciting, thought-provoking impending directions are promising in this research area:

- Our focus has been on the network layer were we used AODV routing protocol, we will however seek to extend our technique to other layers such as Medium Access Control (MAC) layers or application layers.
- Explore how to design more detection strategies especially at the hands of Zero day attacks and other sophisticated attacks.
- Broaden the scope beyond MANETs, to incorporate other technologies like wireless sensor networks.

ABBREVIATIONS

MANET	: Mobile Ad hoc NETwork
IDS	: Intrusion Detection System
AODV	: Ad hoc On-demand Distance Vector
RERROR	: Route Error
RREQUEST	: Route Request
WLAN	: Wireless Local Area Network
WSN	: Wireless Sensor Network
AID	: Anomaly based Intrusion Detection
DARPA	: Defense Advanced Research Project Agency
RREPLY	: Route Reply
MAC	: Medium Access Control Layer
DoS	: Denial of Service
DSR	: Dynamic Source Routing
EM	**:** Expectation Maximization Algorithm
HIDS	: Host based Intrusion Detection System
IDWG	: Intrusion Detection Work Group
IEEE	: Institute of Electrical and Electronic Engineers
IETF	: Internet Engineering Task Force
MBC	: Model Based Clustering
MLE	: Maximum Likelihood Estimate
NEC	: Normalized Entropy Criterion
NIDS	: Network based Intrusion Detection System
PDA	: Personal Digital Assistance
RIDAN	: Real-time Intrusion Detection System
SID	: Signature based Intrusion Detection
SSL	: Secure Sockets Layer
ARP	: Address Resolution Protocol
CA	: Certificate Authority

REFERENCES

[1] S. Buchegger and J.-Y. L. Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP'02), pages 403{410, 2002.

[2] M. Jakobsson, J.-P. Hubaux, and L. Buttyan. A micropayment scheme encouraging collaboration in multi-hop cellular networks. In Proceedings of Financial Crypto 2003, 2003.

[3] K. Liu, J. Deng, P. Varshney, and K. Balakrishnan. An acknowledgment based approach for the detection of routing misbehaviour in Manets. IEEE Transactions on Mobile Computing, 6(5):536{550, May 2007.

[4] Y. Liu and Y. R. Yang. Reputation propagation and agreement in mobile ad-hoc networks. In Proc. of IEEE Wireless Communication and Networking Conference (WCNC'03), pages 1510{1515, March 2003.

[5] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehaviour in mobile ad hoc networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), pages 255{265, 2000.

[6] S. Buchegger and J.-Y. L. Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP'02), pages 403{410, 2002

[7] M. Jakobsson, J.-P. Hubaux, and L. Buttyan. A micropayment scheme encouraging collaboration in multi-hop cellular networks. In Proceedings of Financial Crypto 2003, 2003.

[8] B. Culpepper, H. Tseng, N. Center, and C. Monett Field. Sinkhole intrusion indicators in DSR MANETs. In Proceedings of Broadband Networks. First International Conference on, pages 681{688, 2004.

[9] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In Network and Distributed System Security Symposium (NDSS), pages 131{141, 2004.

[10] Y. Hu, A. Perrig, and D. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In Proceedings of the 2nd ACM workshop on Wireless security, pages 30{40, 2003.

[11] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. Chang. Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach. In Proceedings of IEEE Wireless Communications and Networking Conference, volume 2, 2005.

[12] K. Liu, J. Deng, P. Varshney, and K. Balakrishnan. An acknowledgment based approach for the detection of routing misbehaviour in Manets. IEEE Transactions on Mobile Computing, 6(5):536{550, May 2007.

[13] Y. Liu and Y. R. Yang. Reputation propagation and agreement in mobile ad-hoc networks. In Proc. of IEEE Wireless Communication and Networking Conference (WCNC'03), pages 1510{1515, March 2003.

[14] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehaviour in mobile ad hoc networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000), pages 255{265, 2000.

[15] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In Proceedings of the Sixth IFIP Conference on Security Communications and Multimedia (CMS02), 2002.

[16] E. Ngai, J. Liu, and M. Lyu. On the intruder detection for sinkhole attack in wireless sensor networks. In Proceedings of the IEEE International Conference on Communication (ICC), 2006.

[17] A. Pirzada and C. McDonald. Circumventing sinkholes and wormholes in wireless sensor networks. In Proceedings of the International Conference on Wireless Ad Hoc Networks (IWWAN), 2005.

[18] R. Poovendran and L. Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. Wireless Networks, 13(1):27{59, 2007.

[19] B. Culpepper, H. Tseng, N. Center, and C. Monett Field. Sinkhole intrusion indicators in DSR MANETs. In Proceedings of Broadband Networks. First International Conference on, pages 681{688, 2004.

[20] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto. Detecting blackhole attack on AODV-based mobile ad hoc networks by dynamic learning method. International Journal of Network Security, (3):338{346, 2007.

[21] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In Network and Distributed System Security Symposium (NDSS), pages 131{141, 2004.

[22] Y. Hu, A. Perrig, and D. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In Proceedings of the 2nd ACM workshop on Wireless security, pages 30{40, 2003.

[23] S. Buchegger and J.-Y. L. Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing (EUROMICRO-PDP'02), pages 403{410, 2002.

[24] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM 2003), San Francisco, CA, USA, April 2003.

[25] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In Proceedings of the 2003 ACM Workshop on Wireless Security, pages 30–40, San Diego, CA, USA, 2003. ACM Press.

[26] L. Tamilselvan and V. Sankaranarayanan. Prevention of Blackhole Attack in MANET. In Proceedings of the 2nd International Conference on Wireless Broadband and Ultra Wideband Communications, pages 21{21, 2007.

[27] R. Griswold and S. Medidi. Malicious node detection in ad-hoc wireless networks. In Proceedings of SPIE AeroSense, Digital Wireless Communications V, April 2003.

[28] S. Medidi, M. Medidi, S. Gavini, and R. Griswold. Detecting packet mishandling in Manets. In Security and Management, pages 159–162, 2004.

[29] S. R. Medidi, M. Medidi, and S. Gavini. Detecting packet-dropping faults in mobile ad-hoc networks. In Proceedings of The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, pages 1708–1712, November 2003.

[30] M. Gerla, Ad Hoc Networks: Technologies and Protocols. Springer, 2005, Ch. 1, pp.1 {22.

[31] I. F. Akyildiz, X. Wang, and W. Wang, \Wireless mesh networks: A survey," Computer Networks Journal (Elsevier), vol. 47, no. 6, pp. 445{487, March 2005.

[32] S. Buchegger and J. Le Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In Proceedings of the Parallel, Distributed and Network-based Processing, pages 403–410, January 2002.

[33] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehaviour in mobile ad hoc networks. In Proceedings of the Mobile Computing and Networking, pages 255–265, 2000.

[40] L. Buttyan and J.-P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," in Mobile Networks and Applications, 4483, pp. 5752{5522.

[41] H. Miranda and L. Rodrigues, "Friends and foes: Preventing selfishness in open mobile ad hoc networks," in ICDCSW'03, 4483.

[42] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in Proceedings of IEEE INFOCOM '03, San Francesco, CA, and April 4483.

[43] B. Raghavan and A. C. Snoeren, "Priority forwarding in ad hoc networks with self- interested parties," in Workshop on Economics of Peer to Peer, June 4483.

[44] F. K. J. Crowcroft, R. Gibbens and S. Ostring, "Modelling incentives for collaboration in mobile ad hoc networks," in Proceedings of Workshop on Modelling and Optimization in Mobile, Ad Hoc and Wireless Networks , Ad Hoc, and Wireless Networks (WiOpt'03),France, March 4483.

[45] A. Mok, E. C. Bina Mistry, and B. Li, "Fair: Fee arbitrated incentive architecture in wireless ad hoc networks," in 48th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04), 4484, p. 38.

[46] Y. Zhang, W. Lou, and Y. Fang, "Sip: a secure incentive protocol against selfishness in mobile ad hoc networks," in IEEE Wireless Communications and Networking Conference (WCNC'04), March 4484.

[47] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," in Journal of the Operational Research Society, vol. 452, 152528. [Online]. Available: citeseer.ist.psu.edu/kelly528rate.html

[48] S. Marti, T. Giuli, K. Lai, and M. Baker, Mitigating routing misbehaviour in mobile ad hoc networks," in The 6th ACM International Conference on Mobile Computing and Networking, 2000.

[49] P. Michiardi and R. Molva, "CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in CMS'2002, Communication and Multimedia Security 2002 Conference, September 26-50, 2002, Portoroz, Slovenia / Also published in the book : Advanced Communications and Multimedia Security ,Borka Jerman-Blazic & Tomaz Klobucar, editors, Kluwer Academic Publishers, ISBN 1-4020-7206-6, August 2002 , 320 pp, August 2002.

[50] M. Hauspie and I. Simplot-Ryl, "Cooperation in ad hoc networks: Enhancing the virtual currency based models," in Proceedings of the 1st ACM International Conference on Integrated Internet Ad hoc and Sensor Networks InterSense 2006), Nice, France, May 2006.

[51] E. Huang, J. Crowcroft, and I. Wassell, "Rethinking incentives for mobile ad hoc networks," in Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems, Portland, USA, 2004, pp. 1521{1526.

[52] F. Kargl, A. Klenk, S. Schlott, and M.Weber, "Advanced detection of selfish or malicious nodes in ad hoc networks," August 2004.

[53] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks)," in MOBIHOC'02, 2002.

[54] University of Southern California Information Sciences Institute (USC/ISI). The network simulator -ns-2. Computer software. Available from http://www.isi.edu/nsnam/ns/.

[55] WPI. NS by Example. Online. Accessed from http://nile.wpi.edu/NS/.

[56] M. Greis. Tutorial for the Network Simulator "ns". Online. Accessed from http://www.isi.edu/nsnam/ns/tutorial/.

[57] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In Wireless Communication & Mobile Computing, pages 483–502, 2002.

[58] D. Johnson, D. Maltz, and J. Broch, "The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," Ad Hoc Networking, Addison-Wesley, 2001.

[59] C.E. Perkins and E.M. Royer, "Ad-Hoc on-Demand Distance Vector Routing," Proc. Second IEEE Workshop Mobile Computing Systems and Applications (WMCSA '99), pp. 90-100, 1999.

[60] D. Ganesan, B. Krishnamurthy, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An Empirical Study of Epidemic Algorithms in Large Scale Multihop Wireless Networks," Technical Report Intel IRPTR-02-003, Intel Research, Mar. 2002.

[61] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,"ACM SIGCOMM Computer Comm. Rev., vol. 24, pp. 234-244, 1994.

[62] "The Network Simulator - ns-2," http://www.isi.edu/nsnam/ns,2011.

[63] S. Bagchi, S. Hariharan, and N.B. Shroff "Secure Neighbor Discovery in Wireless Sensor Networks," Technical Report ECE 07-19, Purdue Univ, http://docs.lib.purdue.edu/ecetr/360, 2007.

[64] R. Muraleedharan and L.A. Osadciw, "Jamming Attack Detection and Countermeasures in Wireless Sensor Network Using Ant System," Proc. Wireless Sensing and Processing, vol. 6248, p. 62480G, 2006.

[65] S. Buchegger and J.L. Boudec, "Robust Reputation System for P2P and Mobile Ad-Hoc Networks," Proc. Workshop Economics of Peerto-Peer Systems, 2004.

[66] I. Khalil, S. Bagchi, and N. Shroff, "SLAM: Sleep-Wake Aware Local Monitoring in Sensor Networks," Proc. 37th Ann. IEEE/IFIP Int'l Conf. Dependable Systems and Networks (DSN '07), pp. 565-574, June 2007.

[67] N. Sastry, U. Shankar, and D. Wagner, "Secure Verification of Location Claims," Proc. ACM Workshop Wireless Security (WiSe '03), pp. 1-10, 2003.

[68] L. Hu and D. Evans, "Using Directional Antennas to Prevent Wormhole Attacks," Proc. Network and Distributed System Security Symp. (NDSS '04), pp. 131-141, 2004.

[69] S. Hariharan, N. Shroff, and S. Bagchi, "Secure Neighbor Discovery in Wireless Sensor Networks," Technical Report ECE 07-19, Purdue Univ., 2007.

[70] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proc. Sixth Ann. Int'l Conf. Mobile Computing and Networks, pp. 255-265, 2000.

[71] R. de Oliveira and T. Braun, "A Dynamic Adaptive Acknowledgment Strategy for TCP over Multihop Wireless Networks," Proc. IEEE INFOCOM, pp. 1863-1874, 2005.

[72] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing Exposed Terminals in Wireless Networks," Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI '08), pp. 59-72, 2008.