

1 .Introduction

A mobile ad-hoc network (MANET) is a self-configuring network of mobile routers (and associated hosts) connected by wireless links - the union of which form a random topology as shown in figure 1.1. The routers are free to move randomly and organize themselves at random; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet.

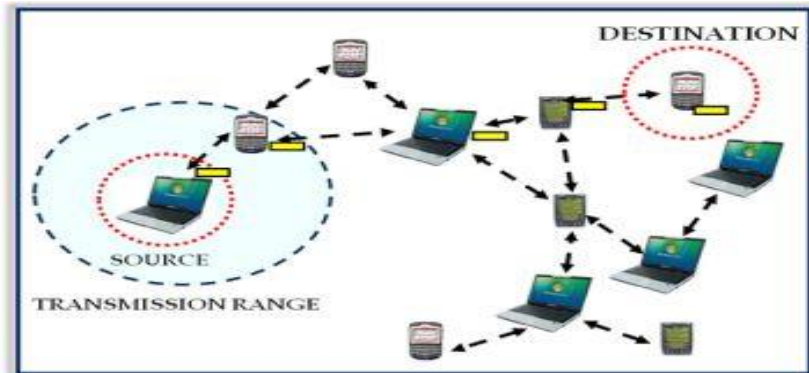


Figure 1.1: Scenario of MANET.

The main characteristics of MANETs are: autonomous in the behaviour of nodes, multi-hop radio relaying, decentralized control, rapid mobility of hosts, frequent dynamically varying network topology, shared broadcast radio channel, limited availability of resources, such as CPU processing capacity, memory power, battery power, and bandwidth, High user density, large level of user mobility, Nodal connectivity is intermittent, distributed operations, light weight terminals and shared physical medium.

The five different types of MANETs include:

- **InVANETs** – Intelligent vehicular ad hoc networks make use of artificial intelligence to tackle unexpected situations like vehicle collision and accidents.
- **Vehicular ad hoc networks (VANETs)** – Enables effective communication with another vehicle or helps to communicate with roadside equipments.
- **Internet Based Mobile Ad hoc Networks (iMANET)** – helps to link fixed as well as mobile nodes.
- **Smart phone ad hoc networks (SPAN):** (SPANs) leverage the existing hardware (primarily Bluetooth and Wi-Fi) in commercially available smart phones to create peer-to-peer networks without relying on cellular carrier networks, wireless access points, or traditional network infrastructure.
- **Military / Tactical MANETs** are used by military units with emphasis on security, range, and integration with existing systems.

MANETs are a kind of Wireless ad hoc network that usually has a routable networking environment on top of a Link Layer ad hoc network. MANETs consist of a peer-to-peer, self-forming, self-healing network. MANETs circa 2000-2015 typically communicate at radio frequencies (30 MHz - 5 GHz).

The advantages of an Ad-Hoc network include the following: They provide access to information and services regardless of geographic position, Self-configuring, Self-healing and Self organising network, less expensive as compared to wired network, Improved Flexibility, Robust and The network can be set up at any place and time.

The domain of applications for MANETs is diverse establishing survivable, efficient, dynamic communication for: network-centric military/battlefield environments, emergency/rescue operations, disaster relief operations, intelligent transportation systems, conferences, fault-tolerant mobile sensor grids, smart homes, patient monitoring, environment control, and other security sensitive applications. In civilian environments like taxicab networks, boats and small aircrafts.

The growth of wireless communication including mobile communication field is at very much high level during the last few decade. Currently second generation (2G) and third generation cellular systems have been reached probably at saturation level, which enables worldwide mobile connectivity. Now a day, Mobile users are using their smart phones to check email and browse the Internet. Recently, an increasing number of WLAN hot spots is rising, which will allow travellers with portable computers to surf the Internet from any feasible locations like airports, railway stations, hotels, school or college campus as well as other public locations. Presently, third generation (3G) provides higher data rates, location-based or person requirement based services.

A MANET environment has to overcome certain issues of limitation and inefficiency. It includes: Limited bandwidth, wireless link characteristics are time-varying in nature, Limited range of wireless transmission, Packet losses due to errors in transmission, Route changes due to mobility, frequent network partitions, hidden terminal problem and security threats.

MANET is more vulnerable than wired network. Some of the vulnerabilities are as follows: lack of centralized management, no predefined boundary, cooperativeness, limited power supply, adversary inside the network.

In MANET, all networking functions such as routing and packet forwarding, are performed by nodes themselves in a self-organizing manner. For these reasons, securing a mobile ad-hoc network is very challenging. Mobile wireless networks are generally more prone to physical security threats than are fixed, hardwired networks. Existing link-level security techniques (e.g. encryption) are often applied within wireless networks to reduce these threats.

The goals to evaluate if mobile ad-hoc network is secure or not are as follows: Availability, Confidentiality, Integrity, Authentication, Authorization, Resilience to attacks, and Freshness.

Further chapters of the books are organized as follows chapter 2 describes about the routing protocols in the MANETs, chapter 3 discuss about the AODV routing protocol in detail, chapter 4 describes the various security attacks in the MANETs, gives the in detail description of the black hole attack, and gives the in detail description of the gray hole attack, chapter 5 discusses the related work on detection and prevention of gray hole attack, chapter 6 gives a detailed description of proposed system i.e. Reputation based IDS where the malicious node is identified by the additive increment and exponential decrement of the reputation of a node, chapter 8 gives the implementation details, chapter 9 shows the simulation results, chapter 10 gives the performance analysis and chapter 11 concludes the proposed work.

2. Routing Protocols for MANETs

Routing is the process of information exchange from one host to the other host in a network. Routing is the mechanism of forwarding packet towards its destination using most efficient path. Efficiency of the path is measured in various metrics like Number of hops, Traffic; Security etc. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes.

The routing protocols meant for wired networks cannot be used for MANETs because routing in MANETs is nontrivial due to the highly dynamic nature of the mobile nodes. An Ad-hoc routing protocol is a convention or standard that controls how nodes come to agree which way to route packets between computing devices in a MANET. As shown in figure the classification of routing protocols are commonly divided into three main classes; they are Proactive, reactive and hybrid protocols as shown in figure 2.1.

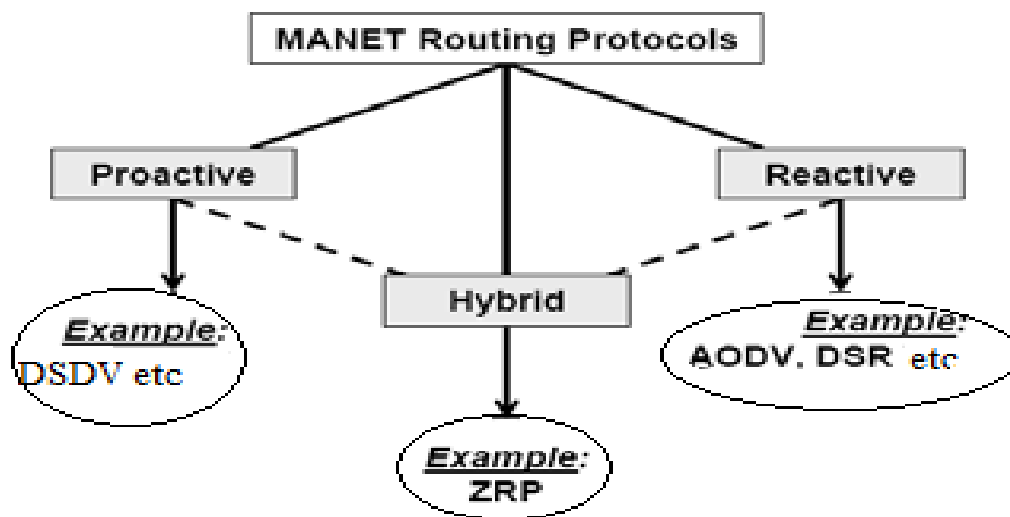


Figure 2.1: classification of routing protocols.

1) Proactive Protocols: Proactive or table-driven routing protocols. In proactive routing, each node has to maintain one or more tables to store routing information, and any changes in network topology need to be reflected by propagating updates throughout the network in order to maintain a consistent network view. Example Destination sequenced distance vector (DSDV).

2) Reactive Protocols: Reactive routing is also known as on-demand routing protocol since they do not maintain routing information at the network nodes if there is no communication. If a node wants to send a packet to another node then this protocol searches for the route in an on-demand manner and establishes the connection in order to transmit and receive the packet. The route discovery occurs by flooding the route request packets throughout the network. Examples Ad-hoc-On-Demand Distance Vector routing (AODV) and Dynamic Source Routing (DSR).

3) Hybrid Protocols: They introduce a hybrid model that combines reactive and proactive routing protocols. The Zone Routing Protocol (ZRP) is a hybrid routing protocol that divides the network into zones. ZRP provides a hierarchical architecture where each node has to maintain additional topological information requiring extra memory.

3. Ad-hoc On-Demand Distance Vector Routing (AODV)

The Ad-hoc On-Demand Distance Vector (AODV) routing protocol builds on the DSDV algorithm because it typically minimizes the number of required broadcasts by creating routes on an on-demand basis. Nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges.

The basic message set consists of: RREQ – Route request, RREP – Route reply, RERR – Route error, HELLO – For link status monitoring

RREQ Messages: A RREQ message is broadcasted when a node needs to discover a route to a destination. The RREQ also contains the most recent sequence number for the destination as shown in figure 3.1. A valid destination route must have a sequence number at least as great as that contained in the RREQ.

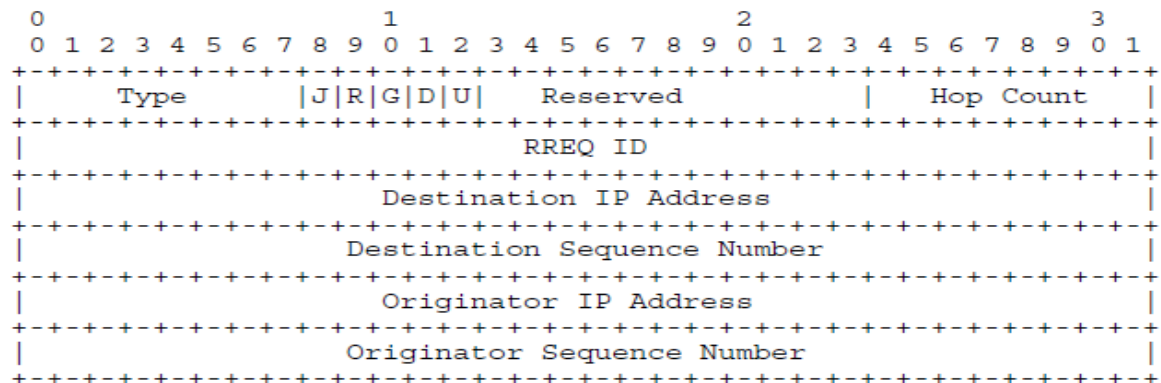


Figure 3.1: RREQ Message Format.

- Type: 1
- J: Join flag (reserved for multicast); R: Repair flag (for multicast).
- G: Gratuitous RREP flag; indicates whether a gratuitous RREP should be unicast to the node specified in the Destination IP Address field.
- Hop Count: The number of hops from the Source IP Address to the node handling the request
- Broadcast ID: A sequence number uniquely identifying the particular RREQ when taken in conjunction with the source node's IP address.
- Destination IP Address: The IP address of destination for which a route is desired.
- Destination Sequence Number: The last sequence number received in the past by the source for any route towards the destination.
- Source IP Address: The IP address of the node which originated the Route Request.
- Source Sequence Number: The current sequence number to be used for route entries pointing to (and generated by) the source of the route request.

Propagation of RREQ message:

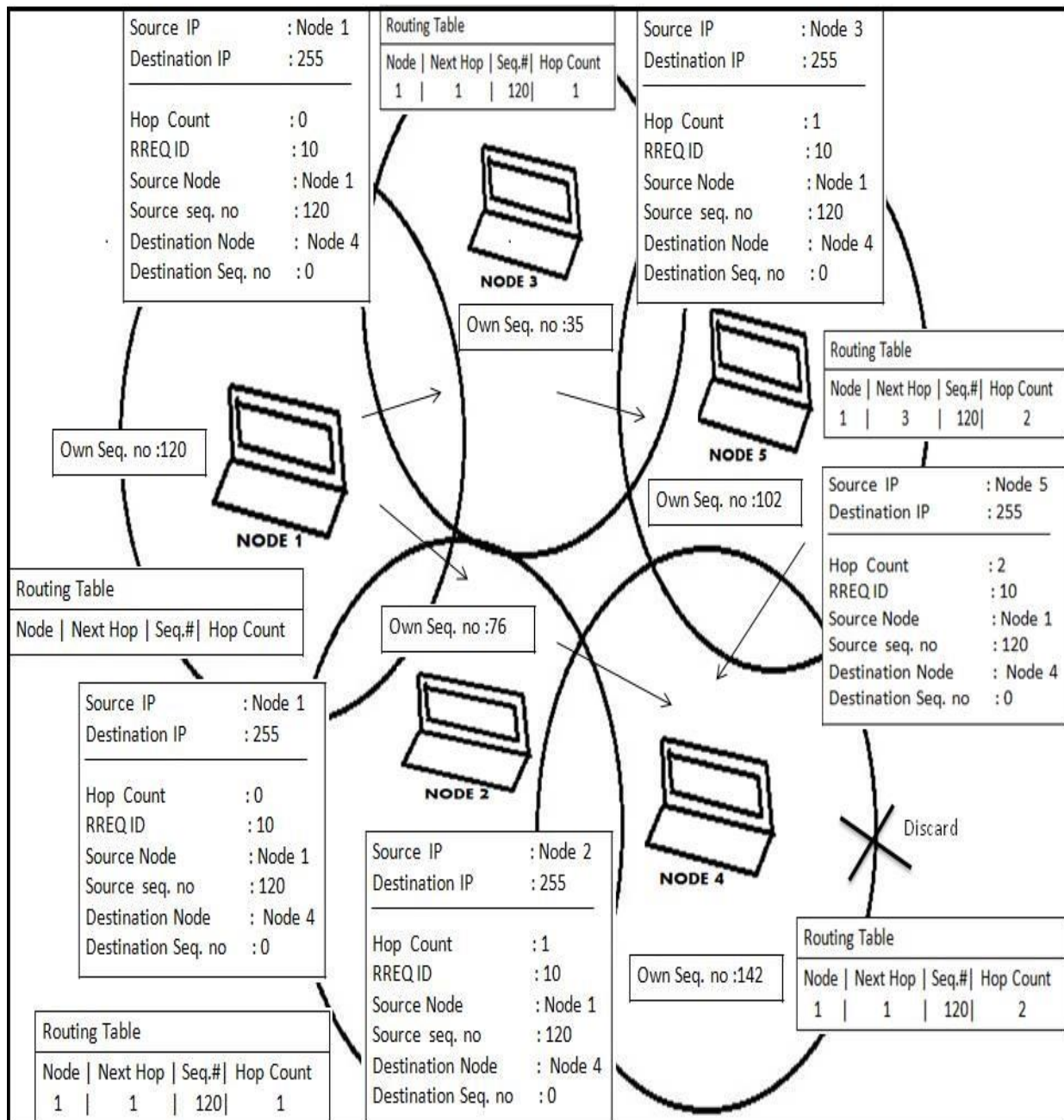


Figure 3.2: propagation of RREQ message

AODV makes sure the route to destination does not contain a loop and is the shortest path. Figure 3.2 shows how RREQ message is propagated in an ad-hoc network. Route Requests (RREQs), Route Reply (RREPs), Route Errors (RERRs) are control messages used for establishing a path to the destination sent by using UDP/IP Protocols. When the source node wants to make a connection with the destination node, it broadcast a RREQ message. This RREQ message is propagated from the source, received by neighbours (intermediate nodes) of the source node. The intermediate nodes broadcast the RREQ message to their neighbours. This process goes on until the packet is received by the destination node or an intermediate node that has a fresh enough route entry to the destination.

RREP Messages: When a RREQ reaches a destination node, the destination route is made available by unicasting a RREP back to the source route. A node generates a RREP if: It is itself the destination or it has an active route to the destination. As the RREP propagates back to the source node, intermediate nodes update their routing tables (in the direction of the destination node). The format of RREP message is as shown in figure 3.3.

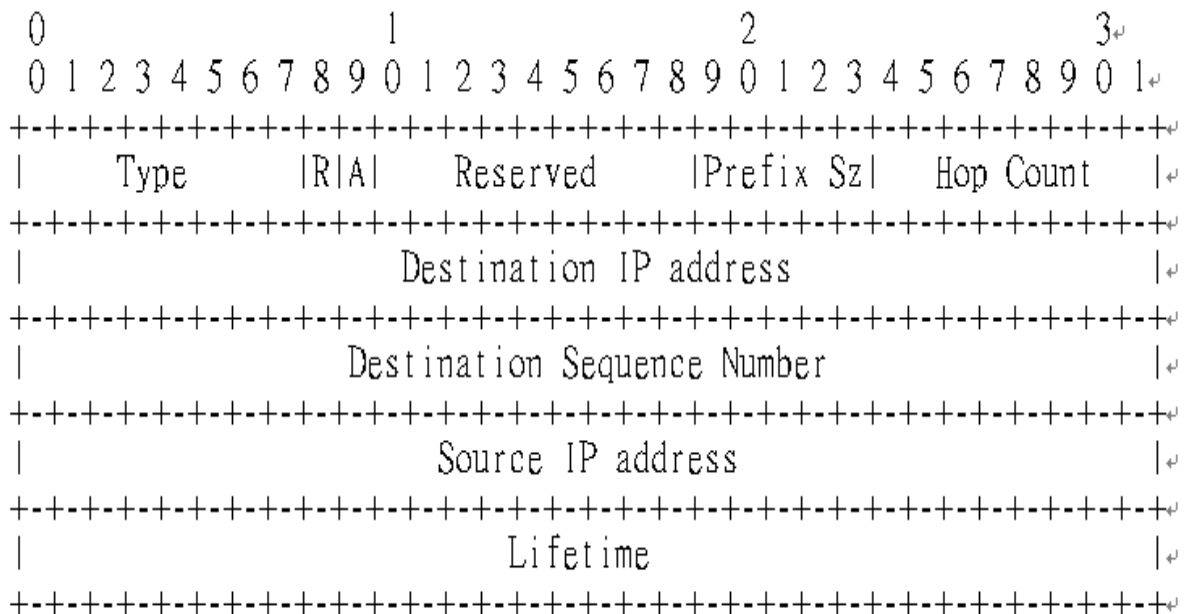


Figure 3.3: RREP message format

Hop Count: The number of hops from the Source IP Address to the node handling the request

Broadcast ID: A sequence number uniquely identifying the particular RREQ when taken in conjunction with the source node's IP address.

Destination IP Address: The IP address of destination for which a route is desired.

Destination Sequence Number: The last sequence number received in the past by the source for any route towards the destination.

Source IP Address: The IP address of the node which originated the Route Request.

Lifetime: The time for which nodes receiving the RREP consider the route to be valid.

Route Error Message: RERR are used mainly when nodes get moved around and connections are lost. If a node receives a RERR, it deletes all routes associated with the new error. Error messages are sent when a route becomes invalid, or if it cannot communicate with one of its neighbors.

HELLO Message: These are simple messages that nodes send at certain time intervals to all its neighbors to let them know that it is still there. If a node stops receiving hello messages from one of its neighbors, it knows that any routes through that node no longer exist.

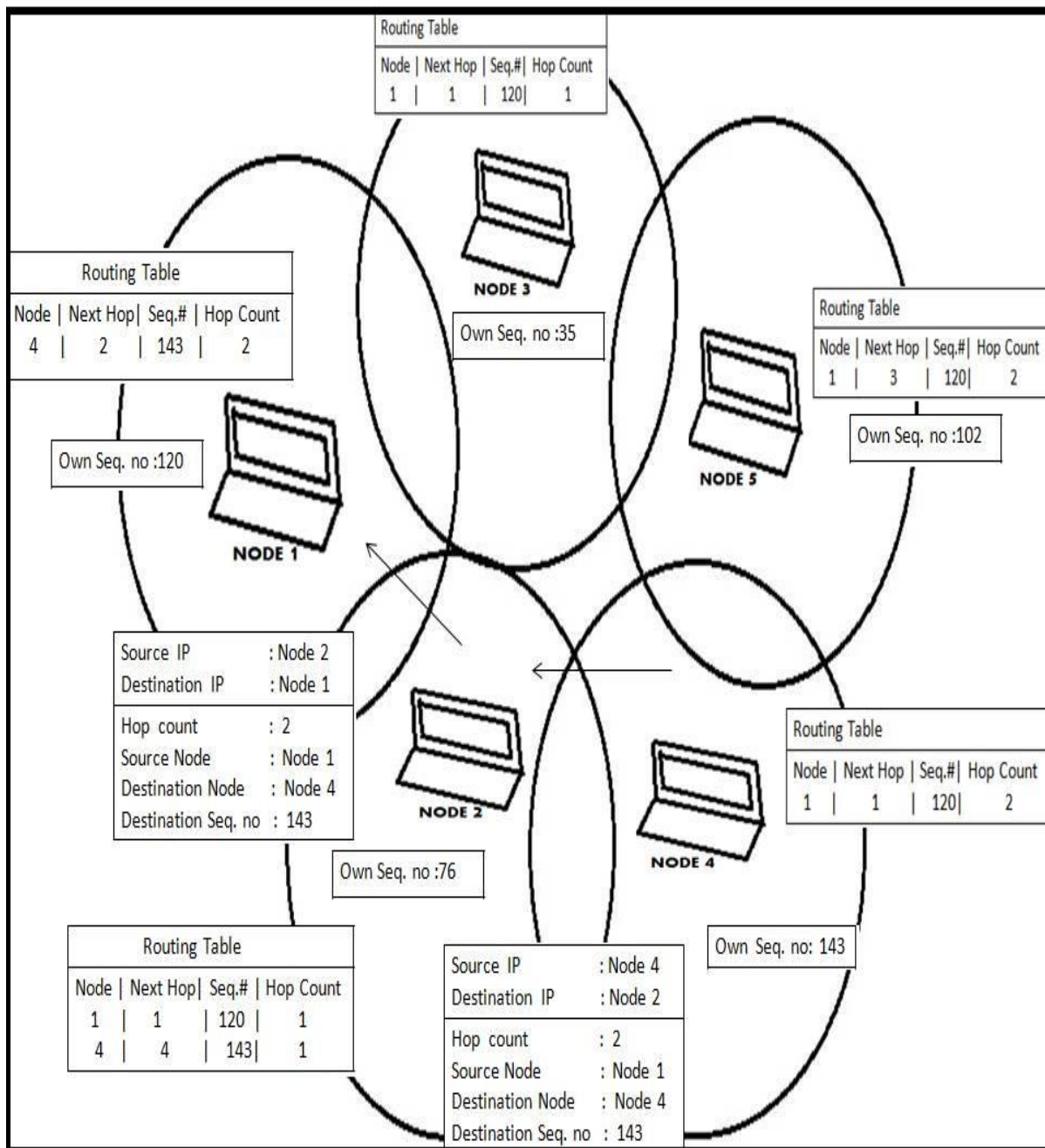
Unicasting the RREP message:

Figure 3.4: Unicasting the RREP message

They create a RREP message and update their routing tables with accumulated hop count and the sequence number of the destination node. Afterwards the RREP message is unicasted to the source node. The difference between the broadcasting RREQ and Unicasting can be seen in the Figure 3.2 and 3.4. While the RREQ and the RREP messages are forwarded by intermediate nodes, intermediate nodes update their routing tables and save the route entry. The node knows over which neighbour to reach at the destination. Figure 6 shows how the RREP message is unicasted and how route entries in the intermediate nodes are updated.

AODV Routing: AODV Routing works by using Route Request Messages (RREQ) and Route Reply Messages (RREP). When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a Path Discovery process to locate the other node. It broadcasts a route request (RREQ) packet to its neighbours, which then forward the request to their neighbours, and so on, until either the destination or an intermediate node with a "fresh enough" route to the destination is located.

AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. Each node maintains its own sequence number, as well as a broadcast ID. The broadcast ID is incremented for every RREQ the node initiates, and together with the node's IP address, uniquely identifies a RREQ. Along with its own sequence number and the broadcast ID, the source node includes in the RREQ the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ.

During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbour from which the first copy of the broadcast packet is received, thereby establishing a reverse path. If additional copies of the same RREQ are later received, these packets are discarded. Once the RREQ reaches the destination or an intermediate node with a fresh enough route, the destination/intermediate node responds by unicasting a route reply (RREP) packet back to the neighbour from which it first received the RREQ. As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their route tables which point to the node from which the RREP came. These forward route entries indicate the active forward route. Associated with each route entry is a route timer which will cause the deletion of the entry if it is not used within the specified lifetime. Because the RREP is forwarded along the path established by the RREQ, AODV only supports the use of symmetric links as shown in figure 3.5.

Routes are maintained as follows. If a source node moves, it is able to reinitiate the route discovery protocol to find a new route to the destination. If a node along the route moves, its upstream neighbour notices the move and propagates a link failure notification message (a RREP with infinite metric) to each of its active upstream neighbours to inform them of the erasure of that part of the route. These nodes in turn propagate the link failure notification to their upstream neighbours, and so on until the source node is reached. The source node may then choose to reinitiate route discovery for that destination if a route is still desired.

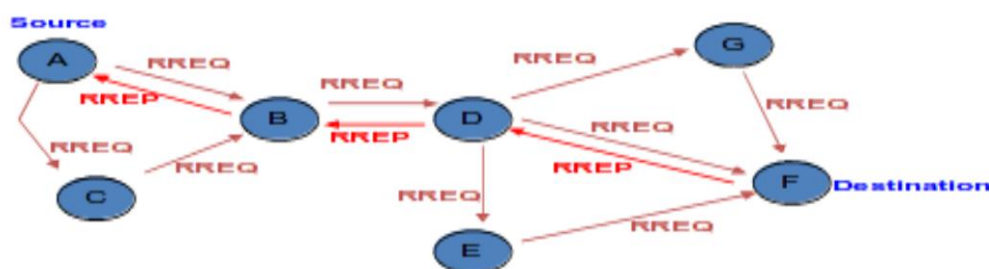


Figure 3.5: Message routing in AODV

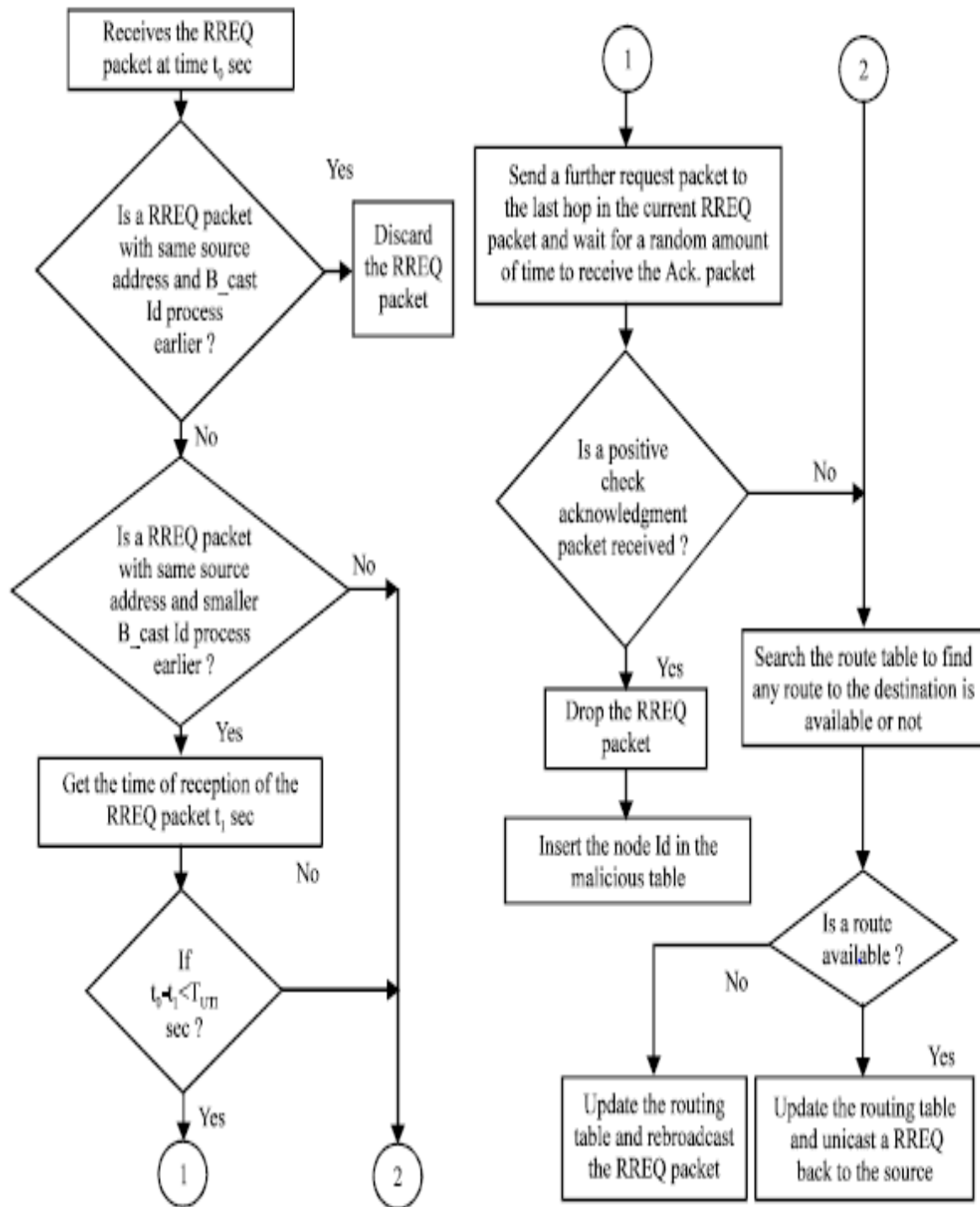
Flow chart representation of AODV:


Figure 3.6: Flow chart of AODV.

Limitations: AODV route discovery latency is high, AODV lacks an efficient route maintenance technique, and AODV lacks support for high throughput routing metrics.

4. Security Attacks in MANETs

A security attack is any action that compromises or bypasses the security of information in an unauthorized way. The attack may alter, release, or deny data. The attacks on the MANETs can be broadly classified into two categories: passive attacks and active attacks as shown in figure 4.1. Both passive and active attacks can be made on any layer of the network protocol stack.

Passive Attacks: A passive attack attempts to retrieve valuable information by listening to traffic channel without proper authorization, but does not affect system resources and the normal functioning of the network. The different types of passive attacks are eavesdropping (information leakage), traffic monitoring, and analysis. Passive attacks are very difficult to detect because they do not involve any alteration of the data.

Active Attacks: An active attack attempts to alter or destroy system resources and the data being exchanged in the network by injecting or modifying arbitrary packets, thus gain authentication and tries to affect or disrupt the normal functioning of the network services. An active attack involves information interruption, modification, or fabrication of the data. Active attacks can be either internal or external.

The characteristics of MANETs make them susceptible to many new attacks. These attacks can occur in different layers of the network protocol stack.

- **Attacks at Physical Layer:** Eavesdropping, Jamming and Active Interference.
- **Attacks at Data Link Layer:** Selfish Misbehaviour of Nodes, Malicious Behaviour Of nodes, Denial of Service, Misdirecting traffic.
- **Attacks at Network Layer:** Black hole Attack, Gray hole Attack, Rushing Attack, Wormhole Attack, Sinkhole Attack, Byzantine Attack.
- **Attacks at Transportation Layer:** Session Hijacking.
- **Attacks at Application Layer:** Repudiation Attack and Malicious code Attack.



Figure 4.1: Classification of Attacks.

4.1 Black Hole Attack:

In black hole attack, a malicious node uses its routing protocol in order to advertise itself for having the shortest path to the destination node or to the packet it wants to intercept. This hostile node advertises its availability of fresh routes irrespective of checking its routing table. In this way attacker node will always have the availability in replying to the route request and thus intercept the data packet and retain it. In protocol based on flooding, the malicious node reply will be received by the requesting node before the reception of reply from actual node; hence a malicious and forged route is created. When this route is establish, now it's up to the node whether to drop all the packets or forward it to the unknown address. The method how malicious node fits in the data routes varies.

The method how malicious node fits in the data routes varies. Figure.4.1.1 shows how black hole problem arises, here node "S" wants to send data packets to node "D" and initiate the route discovery process. So if node "M" is a malicious node then it will claim that it has active route to the specified destination as soon as it receives RREQ packets. It will then send the response to node "S" before any other node. In this way node "S" will think that this is the active route and thus active route discovery is complete. Node "S" will ignore all other replies and will start seeding data packets to node "M". In this way all the data packet will be lost consumed or lost.

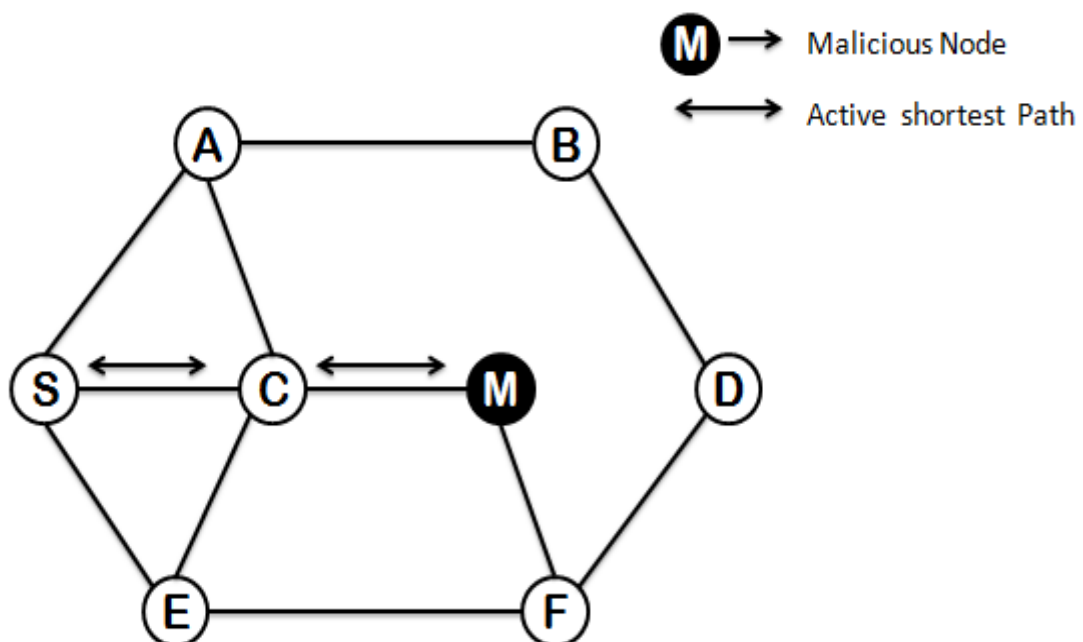


Figure: 4.1.1 Scenario of Black Hole Attack.

Sometimes a chain of Black Hole nodes act cooperatively and create attack in the MANET and this is called cooperative black hole attack.

4.2 Gray Hole Attack:

Gray hole is one of the attacks found in ad hoc network .Which act as a slow poison in the network side it means we cannot suppose how much data can be lost. In Gray hole Attack a malicious node trashes to precede certain packets and simply drops them. The attacker selectively drops the packets originating from a single IP address or a range of IP addresses and forwards the remaining packets. Gray Hole nodes in MANETs are very effective .the scenario of gray hole attack is shown in figure 4.2.1.

The Gray Hole attack has two significant phases:

In first phases, a malicious node exploits the AODV protocol to announce itself as having a valid route to destination node, with the intension of interjecting or humiliating packets, even though route is counterfeit. In the second phase, the node drops the intercepted packets with a certain probability.

This attack is more difficult to detect than the black hole attack where the malicious node drops the received data packets with certainty. A gray hole may exhibit its malicious behaviour in different ways. It may drop packets coming from (or destined to) certain specific node(s) in the network while forwarding all the packets for other nodes .Another type of gray hole node may behave maliciously for some time duration by dropping packets but may switch to normal behaviour later. A gray hole may also exhibit a behaviour which is a combination of the above two, thereby making its detection even more difficult.

Detection of Gray Hole attack is harder because nodes can drop packets partially not only due to its malicious nature but also due to congestion. Detection is difficult because the node's nature is not stable, it can't predicted that when node will be malicious and when it will turn to normal node.

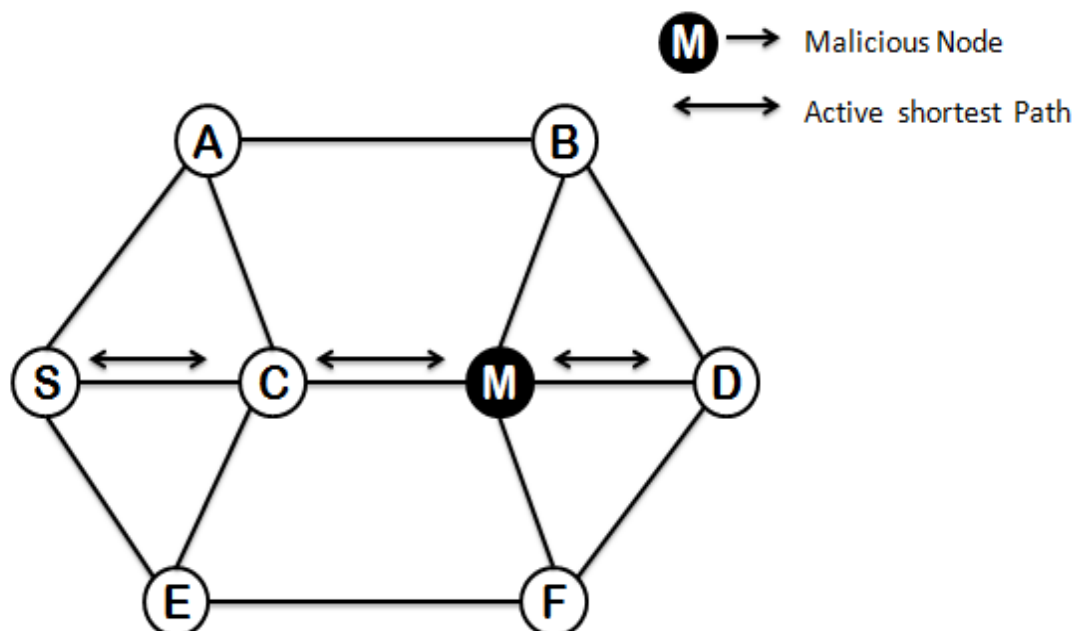


Figure 4.2.1: Scenario of Gray Hole attack.

5. Literature Survey

Amos J Paul et al. proposed system in this approach, when the source node (SN) wants to communicate with the destination node, it takes the help of Back Bone Nodes (BBN- which are trustworthy and powerful in terms of battery power and range). The source node requests the nearest BBN for a Restricted IP (RIP) which is a unique and unused IP address [1]. The BBN on receiving the request message replies with a unique IP address from a pool of unused IP addresses. Now the source node sends the RREQ to both destination and RIP at the same time.

If the source node gets the RREP only for the destination, it means that it is a normal case where the packets are delivered properly. So the source node reuses the RIP for a definite period of time for further data transmissions.

On the other hand, if the source node gets the RREP for the RIP, it means that there is possibility of presence of a malicious node in the active route. Now the source node starts the detection process. It alerts the neighbouring nodes to enter into promiscuous mode so that they listen to the packets that are destined to the specified destination node. Now the source node sends some dummy packets to the destination while the neighbouring nodes start monitoring the packet flow. These neighbouring nodes transmit the monitor message to the every next hop of dummy packet. At a certain point, the monitoring nodes find out that the loss of dummy packets is more than the normal expected loss in a network. Then it informs the source node about that particular Intermediate Node (IN). In this way the SN identifies the black/gray hole in the network. The black/gray hole attack removal process is done by analyzing the feedback sent by the alternate paths that reach the destination node.

Draw backs: Setting of BBN nodes and RIP in the network is complex and for the storage information of RIP it needs extra memory. During the detection process sending the dummy packets increases the traffic in the network and the usage of bandwidth for reliable packets will be less. Communication with all the neighbouring nodes results in the congestion in the network. The reliability of detection of gray hole attack is less in this proposed system.

Ping Yi et.al. proposed system in this detection algorithm is a **path based scheme**. In this, a node does not watch every node in the network, but only observes the next hop in the active route [2]. To implement the algorithm, every node should keep a FwdPktBuffer, which is a packet signature buffer.

The algorithm is divided into three steps:

- 1) When a packet is forwarded out, its signature is added into the FwdPktBuffer and the detecting node overhears.
- 2) Once the action that the next hop forwards the packet is overheard, the signature will be released from the FwdPktBuffer.
- 3) In a fixed period of time, the detecting node should calculate the overhear rate of its next hop and compare it with a threshold.

Overhear rate for Nth period of time is calculated using the formula

$$OR(N) = \text{total overhear packet number} / \text{total forward packet number}$$

If the forwarding rate is lower than the threshold, the detecting node will consider the next hop as a black or gray hole.

This algorithm has several advantages like each node depends only on itself, no need of encryption on the control packets to avoid further attacks, no need to watch all the neighbouring nodes' behaviour.

One problem of this detection method is that it suffers from a high false positive probability under high network overload if a constant threshold is used. The cause of high false positive probability is hidden node problem. A hidden node is a node which is beyond range of a packet sender but in the range of a packet receiver. These hidden nodes lead to higher collision probability. To avoid this problem caused by hidden node problem, a cross-layer mechanism was proposed.

For this, collisionPktNum and nonColPktNum are added to standard 802.11 protocol. If a collision occurs, collisionPktNum increases to 1; if a packet has received successfully, nonColPktNum increase to 1. The collision is defined as following.

$$RCR(N) = \text{collisionPktNum} / (\text{collisionPktNum} + \text{nonColPktNum})$$

In network layer, the accumulated collision rate is calculated. This mechanism is able to measure Poverhead using overhear rate $OR(N)$, and Pcollision using accumulated collision rate $ACR(N)$.

The dynamic detection threshold is as follows.

$$Td(N) = 1 - (1 - Tf)(1 - ACR(N))$$

Where Tf is the fixed detection threshold.

If a node drops packets in a probability higher than Tf , the detecting node can say it as a gray hole.

Drawbacks:

This method fails to detect cooperative black hole attack because sometimes the overhearing node may be act as a malicious node. Setting the range of fixed detection is also difficult because the performance of the network may vary and it depends on many factors.

Onkar V. Chandure et.al proposed system in this for the detection of gray hole in the network. This method starts with the initialization process, by first setting the waiting time for the source node to receive the RREQ coming from other nodes and then add the current time along with the waiting time. Then store all the RREQ Destination Sequence Number (DSN) and its Node Id in RR-Table until the computed time exceeds. Generally the first route reply will be from the malicious node with high destination sequence number, which is stored as the first entry in the RR-Table[3]. Then compare the first destination sequence number with the source node sequence number, if there exists much more differences between them, surely that node is the malicious node, immediately remove that entry from the RR-Table. This is how malicious node is identified and removed. Final process is selecting the next node id that has the higher destination sequence number. This is obtained by sorting the RR-Table according to the DSEQ-NO column, whose packet is sent to Receive Reply method in order to continue the default operations of AODV protocol.

Drawbacks:

This proposed system is unreliable to implement in a real time MANET. Here it is comparing the first node destination sequence number with the second one if the second node is also malicious then the difference will be less and it fails to identify the malicious node. Maintenance of RR table is an extra overhead. It also fails to address the behaviour of gray hole sometimes acting as normal node.

Meenu Chawla proposed system it is an algorithm which is a Destination based group gray hole attack detection in MANET through AODV. In this proposal, to identify the suspected node, the common neighbour of previous node and suspected node checks the two hop distance node to reach the destination[4]. To do so first it stores the RREP packet at previous node.

In AODV routing protocol, when the source node wants to communicate with the destination node it broadcasts the route request (RREQ) message to its neighbour first. If neighbour is a destination then it sends route reply (RREP) message reverse to source

node otherwise it forwards packet by updating their routing table. When RREP message replies to previous node it should also attach the one hop distance of suspected node otherwise previous node will reject the RREP message.

When there is no malicious node present in network, data packets successfully travel between source node and destination node but if there is malicious node present in MANET then it sends route reply (RREP) message to source by falsely replying that there is valid route.

The node which is common neighbour of suspected node (assume there is at least one common neighbour) and previous node check whether destination node is in one hop distance or in two-hop distance. Suppose the common neighbour is cooperative node with the attacker, it supports suspected node and sends back the route reply (RREP) to previous node that suspected node is valid. Therefore, source node follows that path and transfers the data packets, some data is lost at destination node due to co-operative gray hole effect.

Then finally, a fresh special route request (RREQ) message is sent through previous node, which does not follow any node that is one hop distance from suspected node. Now previous node checks routing table of destination node whether suspected node is in one hop distance of destination node, and if it is not in one hop distance then both suspected node and common neighbour will be added in black listed node table. In this way the gray hole is eliminated from data transmission path.

This proposed system is expected to boost up network performance, decrease end-to-end delay and routing overhead. The major factor is to increase the overall network throughput.

Draw backs:

As the nodes in the MANET are arbitrary identifying the neighbour node and two hop distance destination node of the suspected node every time in the real time systems is unreliable. Cross checking whether the destination node is one hop distance every will be an extra overhead and decrease the performance cycle.

Madhuri Gupta et.al proposed system in this the algorithm is divided into two phases:

1) Noticing Phase

2) Confirmation phase.

1) Noticing Phase: In the noticing phase, for communicating with the destination node, Source node (S) firstly want to find the route for the destination node. For this purpose it prepare a RREQ (Route Request) packet, in which it fills the address of the destination node (called as DSTO) and this packet is broadcasted to the neighboring nodes. Now, the source node waits for all the replies send by the replying nodes in terms of the RREP (Route Reply) packets and after getting all the replies from the replying nodes, it sorts these replies in terms of the Decreasing order of the destination sequence numbers (DSN) into its own Route Record (RR)[5]. Means, a RREP contains highest Destination Sequence Number stored in the top of the Route Reply table. Now, the source node compares the Destination Sequence Number of the first entry from the R-R table with the Threshold value (TV), which is average of the all the Destination Sequence Numbers of the replying nodes. Now, If Destination Sequence Number of the first node is much greater than Threshold value the source node notes this node as attacker node and called the second phase.

2) Confirmation Phase: In the Confirmation phase, Source node sends a new RREQ packet for a new destination, known as Virtual Destination (DSTV) and waits for the reply coming from the replying nodes containing the paths from the source node to this virtual node. And stores the replies in terms of the Destination Sequence Numbers, and picks the first entry from the Route Reply table and compare it with the Threshold value and if it is much greater than the Threshold Value and also that node which is already

considered as the noticing node in the previous phase then confirm it as Gray hole attacker node. And after confirming the gray hole attacker node it broadcast the information about this node to all other nodes and then they remove the entry of this gray hole node from their route cache.

Draw backs:

Depending on the destination sequence number is unreliable .This algorithm fails if the virtual destination is malicious and it also fails to explain the gray hole property where it may not act as malicious node during confirmation phase.

Sukla Banerjee proposed system in this system, the behaviour of each node in the route is monitored by all the neighbours of that node. The main idea is dividing the traffic volume into set of data blocks so that the malicious nodes can be captured in between the transmission of two such blocks. Firstly, it divides the data packets to be sent into equal parts. It sends the prelude message containing number of data blocks to destination node [6]. Broadcast monitor message to all its neighbours, Instructing neighbours to monitor next node in the route. Then it starts transmitting data packets from data block to destination. It sets time for the receipt of the postlude message containing count of no of data packets received by Destination .If time is not expired and the postlude message is received continue sending data packets else Broadcast query message to all its neighbours and detected as malicious then terminates data sending and its action.

Draw backs:

This algorithm is not effective because the postlude message may not be received within the time limit due to traffic or congestion in the network we cannot say the node is malicious only with is constraint. Broadcasting number of data packets to all the nodes in the network is an extra overhead.

V.T. Gaikwad et.al proposed system in this security procedure is invoked by a node when it identifies a suspicious node by examining its DRI table. We call the node that initiates the suspected node recognition procedure as the Initiator Node (IN)[7]. The IN first chooses a Cooperative Node (CN) in its neighbourhood based on its DRI records and broadcasts a RREQ message to its 1-hop neighbours requesting for a route to the CN. In reply to this RREQ message the IN will receive a number of RREP messages from its neighbouring nodes. It will certainly receive a RREP message from the Suspected Node (SN) if the latter is really a gray hole (since the gray holes always send RREP messages but drop data packets probabilistically). After receiving the RREP from the SN, the IN sends a probe packet to the CN through the SN. After the time to live (TTL) value of the probe packet is over, the IN enquires the CN whether it has received the probe packet. If the reply to this query is affirmative, (i.e., the probe packet is really received by the CN) then the IN updates its DRI table by making an entry '1' under the column Check Bit against the node ID of the SN. However, if the probe packet is found to have not reached the CN, the IN increases its level of suspicion about the SN and activates the suspected node recognition, once a node is recognized to be really malicious or suspected, the scheme has a notification mechanism for sending messages to all the nodes that are not yet suspected to be malicious, so that the spiteful node can be separated and not allowed to use any network resources.

Draw backs:

In their proposed mechanism if the cooperative node acts as malicious node then the probability of detecting the gray hole will be unreliable. Every time collecting the information from the neighboring nodes and two phase detection of malicious node will be traffic overhead and computational overload in the network.

Ashok M Kanthe et.al proposed system the proposed mechanism is to detect gray hole attack and eliminate the normal nodes with higher sequence number to enter in the black list .In this technique, detection of malicious nodes (m_node) [8] is done during route discovery process. But the m_node is not black listed during first attempt of malicious activity. Whenever a malicious activity is detected by receiving node, it increases a false reply count for replying node in its local black list buffer (recv node). In this approach it makes 3 attempts of false reply to add a m_node in the black list. The attempts of false reply can be incremented as per scenario and elapsed time, node density. Black list is local for each node. Each node maintains its own black list buffer. Information of the list is never broadcasted to any other nodes. Hence any m_node will not broadcast false alarm packet pretending that particular node is malicious node (even it is normal) to other nodes in the network. M_node is detected and black listed when receiving source node detects malicious activity from replying nodes.

Drawbacks:

The proposed system doesn't have a clear algorithm of how it identifies the malicious node. Maintains a black list buffer at every node is an extra overhead in the network.

Hizbullah Khattak et.al proposed system the proposed mechanism consists of three parts:

1. In the first one, slight changes are made for AODV. This system uses the second shortest path for the transmission of the data packets. According to this system if the data is transmitted through the first path, the black/gray hole tries to be in the first shortest path, as AODV uses the first path, due to the minimum hop count. Hence the system uses the second shortest path. The source node can then send data packets safely on this route to the destination node because the malicious node will not be able to know through which route the data will come. Malicious node will need to monitor the entire network which is not an easy task in dense networks.
2. If the receiver has only a single neighbour, the receiver stores the first RREQ to specific threshold time value. If it does not receive any other RREQ within this threshold time value, it means that the receiver has only a single neighbour. In this case, the receiver sends the RREP on the same path.
3. It is also possible that malicious node can be a part of second shortest path. To solve this problem this system proposed to use a hash function, on the message that has to be sent to get a unique message digest (MD)[9].Source node sends this MD with the first data packets to the receiver. The receiver node stores this MD with itself. When the receiver node gets all the data packets, it applies a hash function on this message to get a message digest. It then compares this message digest with the stored one. If both these are equal it means that message has been received safely and there is no such attack node in the route. But if they are not equal it means some data packets have been dropped in the data transmission and there is some malicious node working in the route. In such case, the receiver broadcast Data Packets Received Error (DPRE) to all the nodes in the network.

Draw backs:

Using the second shortest path as an active path in the network is not effective and the probability that it will not have a malicious node is also unpredictable. Performing

message digest at receiver node is an extra overhead and reduces the performance cycle and it complicates the detection process.

Shrishti Jain et.al proposed system in this system IDS node is defined which continuously watch the all neighbour nodes and if they found any intermediate node dropping the data packet through gray hole attack behaviour, then IDS node send the unicast message to the source node about the presence gray hole, this is behavioural analysis mechanism. The other is each node performance check. Here the incoming and outgoing packets performance is watched and if the performance is zero that means the node is receiving packets and is unable to forward them. Such node is detected as a gray hole and IDS send the unicast message to sender to change the route .The IDS node check the intermediate node behaviour and performance when intermediate node is in the range of the IDS node[10]. If the intermediate node is have the gray hole behaviour then the IDS node verifies the performance, if the intermediate node is receiving packets and is not forwarding then the IDS node sends the unicast message to the source node to change the route.

Draw backs:

In this system the malicious node is detected by an intrusion node identifying the intrusion node in the network is complex and if the intrusion node itself is malicious then it will be a failure. False positive rate is high as the performance of a node may be varying due to some other reasons.

Mozmin Ahmed et.al proposed mechanism in this proposed system Data Transfer Quality (DTQ) and Stability Model behaviour (STB) are considered. $DTQ > THRESHOLD$.In this system if a source node detects any node with the DTQ value below the THRESHOLD and then it broadcasts the request to trigger a vote. On receiving this request, the nodes in the MANET check their respective tables for the DTQ values of Node which is detected by source and responds with a positive or a negative vote. The Source Node keeps the count of number of votes it receives from the neighbouring nodes. It accepts only one vote from each node[11]. There is a time limit set for receiving the votes from the neighbouring nodes. Only the votes received within the stipulated time is accounted for aggregation. All the neighbouring nodes that receive the voting request attempts to join the voting process. If suspected node is blacklisted, a message is sent to all the nodes about this information. All the nodes add suspected Node in their blacklist details. If the suspected node is acquitted after the voting, all the nodes treat suspected node as a normal node.

Draw backs:

Fixing a certain threshold value is complex. Voting from all the neighbouring nodes is unreliable and an extra over head to the network. A neighbouring node may not sent the vote within a stipulated time due to traffic and congestion it results in an increase false positive rate.

Sameh R.Zakhary et.al proposed a mechanism which is based on reputation based protocol and the reputation of each node in a MANET collects reputation information, through direct observation of its neighbours (subjective observation) and gathers indirect (second hand) reputations from other nodes. In addition to using historical observations, this protocol uses reputation discounting to ensure that old reputations will fade away giving more chance for nodes to reclaim their reputation by consistently behaving in a cooperative manner. It employs

two kinds of Centrality: Eigen vector and degree centrality in order to elect the most influential nodes to assist in the role of helping other nodes to build their trust into other less popular nodes in the network and act as community leaders. Nodes with higher centrality have higher probability of getting in contact with many other nodes than nodes with low centrality[12]. Both centrality of the reporting nodes and indirect-reputation are key to quick isolation of the malicious nodes and convergence of reputation across all the

nodes. It resolves node's reputation as a function of its centrality characteristics, classify it as high, medium or low centrality. This technique allows the network to evolve into a multiple clusters of different trustworthiness levels. These different levels of trustworthiness allow higher layer applications to limit their interaction only to one selected zone vs. any other zone. Reputation Management is the main entity responsible for storing and retrieving all the node's neighbours' reputation records. Neighbour Reputation Record is the entity representing reputation observation for one of the neighbours. Reputation Broadcast is the entity responsible for receiving indirect reputation from neighbours. It performs a selective deviation test to ensure the unity of view with the receiving node point of view. Resolver is responsible for doing the actual calculation of the neighbour final reputation (called resolved reputation) by combining direct and indirect reputation and performing Reputation Noise Cancellation. Route Maintenance is being called when the Resolver detect that a certain neighbour reputation has fallen below a certain threshold.

Draw backs:

This algorithm is completely based on theoretical calculation of reputation at each node and trusting it based on reputation it may leads to false positive rate as the gray hole attack behaviour is unpredictable.

Marti et.al proposed a mechanism in which a malicious node is traced by using watchdog/ pathrater [13]. In watchdog when a node forwards a packet, the node's watchdog verifies that the next node in the path also forwards the packet by promiscuously listening to the next node's transmissions. If the watchdog finds the next node does not forward the packet during a predefined threshold time, the watchdog will accuse the next node as a malicious node to the source node. In pathrater algorithm each node uses the *watchdog's* monitored results to rate its one-hop neighbours. Further the nodes exchange their ratings, so that the pathrater can rate the paths and choose a path with highest rating for routing.

Draw backs:

The proposal has two shortcomings:

- 1) To monitor the behaviour of nodes two or more hops away, one node has to trust the information from other nodes, which introduces the vulnerability that good nodes may be bypassed by malicious accusation.
- 2) The watchdog cannot differentiate the misbehaviour from the ambiguous collisions, receiver collisions, controlled transmission power, collusion, false misbehaviour and partial dropping.
- 3) Shortcoming of this algorithm is that the idea of exchanging ratings genuinely opens door for blackmail attack.

Ramswamy et.al proposed system it claims to prevent the cooperative black hole attacks in ad-hoc network. In this algorithm each node maintains an additional Data Routing Information (DRI) table. Whenever a node (say IN) responded to a RREQ it send the id of its next hop neighbour (NHN) and DRI entry for NHN to the source[14]. If IN is not a trustable node for source then source sends a further route request (FRQ) to NHN. NHN in turn responds with (FRP) message including DRI entry for IN, the next hop node of current NHN, and the DRI entry for the current NHN's next hop. If NHN is trusted node then source checks whether IN is a black hole or not using the DRI entry for IN replied by NHN. If NHN is not trustable node then the same cross checking will be continued with the next hop node of NHN. This cross checking loop will be continued until a trusted node is found. Moreover, in the case when the network is not under the attack, the algorithm takes more time to complete.

Draw backs:

This algorithm is based on a trust relationship between the nodes, and hence it cannot tackle gray hole attacks.

P.Agrawal et.al proposed System a technique for detecting chain of cooperating malicious nodes (black and gray hole nodes) in ad hoc network. In this technique initially

a backbone network of strong nodes (capable of tuning its antenna to short (normal) as well as to long ranges) is established over the ad hoc network. Each strong node is assumed to be a trustful one. These trustful strong nodes detect the regular nodes (having low power antenna) if they act maliciously. With the assistance of the backbone network of strong nodes, the source and the destination nodes carry out an end-to-end checking to determine whether the data packets have reached the destination or not. If the checking results in a failure then the backbone network initiates a protocol for detecting the malicious nodes. For detecting malicious node strong node associated with source node broadcast a find chain message to the network containing the id of the node replied to RREQ[15]. On receiving find chain message strong node associated with destination node Initialize a list Gray Hole Chain to contain the id of the node replied to RREQ. It then instructs all the neighbours of that node to vote for the next node to which it is forwarding packets. If the next node id is null then the node is a black hole node. Then the gray hole removal process is terminated and a broadcast message is sent across the network to alert all other nodes about the nodes in Gray Hole Chain to be considered as malicious. Else strong node will elect the next node to which replied to RREQ is forwarding the packets based on reported reference counts. Then again broadcast the find chain message containing the id of the elected node.

Draw Backs:

The main disadvantages of this algorithm are the difference between the regular node and backbone node in the network in terms of power, antenna range which makes it unsuitable for all types of mobile ad hoc network. Also it is not proved that backbone network is optimal in terms of minimality and coverage. Algorithm will fail if the intruder attacks strong nodes because it violates the assumption that strong nodes are always trusted node.

G. Xiaopeng et.al proposed system in this the detection scheme against gray hole attack. It consists of three algorithms which are creating proof algorithm, the check up algorithm and the diagnosis algorithm [16]. In creating proof algorithm, the source nodes are creating proof which is based on aggregate signature algorithm for received message. In check up algorithm, the source node suspects the malicious node. In diagnosis algorithm, the evidences are getting from the check up algorithm, it finds the malicious node.

Draw backs:

This mechanism is not detecting all malicious nodes and false positive rate is also high.

Tamilselvan proposed system. It is a solution of wait and check strategy for preventing multiple malicious nodes. The authors proposed that source chooses a secure path by repeated next hope node using "wait and check" strategy after collecting route reply (RREPs) message from neighbour nodes [17]. Source node assumes route to be safe and secure if it finds any repeated nodes in the receiving replies. If source does not find any repeated node, it chooses a path randomly for data packets transmission.

Draw backs:

The wait strategy causes additional processing delay and receiving replies from different nodes create additional delay.

Sun proposed system it is a general solution for detection of black hole attack they developed a neighbourhood-based solution for detection of the attacker and a routing recovery protocol for establishing a safe path to destination[18]. They proposed a neighbour collection of a node within the range of radio transmission of a node. For sharing neighbour set among nodes, two kinds of control packets are introduced. If any node receives two different sets of neighbour at the same time then it can be said that these two neighbours' sets belong to two different nodes.

Draw backs:

The drawback of this solution is that there is a need of public key infrastructure. The detection of the malicious node is still vulnerable.

Deng proposed system he has proposed solution of modifying AODV routing protocol for preventing black hole attacks. In this approach, every intermediate node appends in route reply (RREP) packets the address of the next hop node for identification of the existence of the advertised route of black hole [19]. After receiving the route reply (RREP) packet from intermediate node, source node takes out and finds information of the next hop node and sends supplementary request to the next hop node for verification of routing metric value with the next hope node. For confirming the route information next hop node of neighbour sends back the supplementary reply packet to the sender. In case the source does not get back this supplementary reply, it specifies that the route contains the malicious nodes. This route is removed from the routing table and an alarm message is sent to other nodes in the network to isolate malicious nodes.

Draw backs:

The drawback of this approach is that cooperative black hole attacks can be launched on it. Furthermore, this solution causes additional routing overhead due to supplementary request and supplementary reply for verification.

Awerbuch proposed a system to detect malicious nodes by using acknowledgements sent by destination node. This scheme was consisted of three related algorithms: 1) the route discovery with fault avoidance [20]. By using flooding, cryptography algorithms and weight list, the source nodes could discover route that will deliver packets; 2) the Byzantine fault detection. Based on binary search algorithm and the input path, the source node could detect malicious nodes with Byzantine behaviour; 3) the link weight management. This algorithm is used to update the link weight.

Draw backs:

The proposal has three shortcomings:

- 1) the bandwidth overhead is significant, as the destination node will send an acknowledgement whenever it receives a packet;
- 2) it is a challenging work to make sure that the source node has a shared key with each node in the network;
- 3) the probe packet is easily to be distinguished from other general packet, as the probe packet contains a probe list.

5.1 Problem statement:

Numerous attempt can be found in the literature survey for the detection and prevention of gray hole attack like path based scheme, hash based scheme, cryptographic techniques, identification of the gray hole with the help of neighbouring nodes cooperatively, some imply wait and request strategies, some made the changes to the AODV routing protocol. However all these are inefficient to address the gray hole attack because of its characteristics like selective drop of packets from a specific node or a range of node and misbehaving nature of the gray hole sometimes as malicious node and sometimes as normal nodes. The existing strategies are not efficiently addressing these issues and they are implying extra computational overload and traffic load to the network.

6. PROPOSED SYSTEM

In this section we first mention some practical assumptions that have been made for formulating the network model and then present the proposed mechanism in detail.

Network Model:

We consider a MANET consisting of similar types of nodes. Each node may freely roam, or remain stationary in a location for an arbitrary period of time. In addition, each node may join or leave the network, or fail at any time. The nodes perform peer-to-peer communication over shared, bandwidth-constrained, error-prone and multi-hop wireless channel. For the purpose of differentiation, we assume that each node has a unique nonzero ID. All the links in the network are assumed to be bi-directional. The packets used for the propagation in this network model are user datagram packets (UDP) with a constant bit rate (CBR). Constant bit rate packets are used as they generate a constant traffic during the simulation.

Reputation based IDS:

The proposed mechanism Reputation based IDS involves both node independent mechanism and previous hop observation to identify any malicious gray hole node in the network. Once a node is detected to be really malicious the scheme has a notification mechanism for sending messages to all the nodes that are not yet suspected to be malicious, so that the malicious node can be isolated and not allowed to use any network resources.

Node independent mechanism:

In general there is node independent and node dependent mechanisms for the detection of the malicious node we are making use of node independent mechanism for the reliable detection of the attack. In node independent mechanism every node act as an independent node in the network it does not share its information with the other nodes in the network until the detection process is completed because there may be a chance that the node with which the node share the information may be malicious and it leads to the poor detection of the malicious node. Another advantage of this technique there will be less traffic in the network as there is no sharing of information between nodes.

Reputation:

In this project we present a new routing metric known as reputation to identify a gray hole node in the network. For the identification of whether the node is malicious or not reputation is the more simple and practical approach. Reputation is nothing but a constant value assigned to the nodes. By observing its behaviour the reputation value is additively increased and exponentially decreased. When the reputation value becomes less than zero we will be able to find the node as the malicious node. Reputation metric helps in less mathematical calculation for the identification of the node which results in the less computational cost in the network.

6.1 Prerequisites for Reputation based IDS:

The two main prerequisites for the Reputation based IDS are:

- 1) Average value
- 2) Count

Average value:

It is the minimal drop rate allowed in the network.

Due to various reasons like network congestion, insufficient bandwidth, multi-path fading, faulty network drivers and due to collisions packets drop in a conventional network. To differentiate these packet drops from the intentional dropping of packets of a malicious node we are identifying the minimal drop rate that can occur in a network.

We are calculating the minimal drop rate in the network by the following formula:

$$\text{Collision} = \text{colpktNum} / (\text{colpktNum} + \text{nonColPktNum})$$

Where

ColpktNum = if a collision occurs colpktNum increases to 1.

nonColPktNum = if a collision do not occur and the packet has received successfully nonColPktNum increases to 1.

Count:

It is the number of packets dropped at a certain node in a periodic time interval.

Procedure of Reputation based IDS:

In this mechanism the source initiates the route request for an path to the destination in the network. The malicious node sends a route reply to the source assuring that it has a route to the destination with a minimum hop count and higher destination sequence number. The source believes it as the active path and sends the data packets through this path which includes malicious node. The route reply of a malicious node is propagated without any delay to the source so the probability of malicious node includes in the active path is high.

After the identification of active path. We will identify the nodes that include in the active path and assign a reputation value to those nodes. We are assigning reputation to the nodes which are in active path only not to the all the nodes in the network because we will make use of previous node reputation in the path. It helps in reducing the unnecessary memory load in the network. Here every node stores the reputation value of the previous node.

Every node in the active path observes the previous node in the active path and identifies the drop rate at each node with the help of count variable in a periodic time interval. Here two conditions arises they are if the count i.e. drop rate is less than the average value i.e. the minimal drop rate allowed in the network then the reputation value is additively incremented. If the count is greater than the average value it means the drop of packets is more than the minimal drop rate then the reputation value is exponentially decremented. Here we are decreasing the reputation exponentially for faster detection of the malicious node.

At a certain instant of time if the reputation value becomes less than zero we declare it as the malicious node and broadcast a error message to all other nodes in the network and isolates it. The node which is malicious and the next hop in the active are stored in a file and this file is maintained at every node. Next time whenever it receives a route reply it cross checks whether the route reply is from the malicious node by cross checking the data within the file. If the route reply is from malicious node it free the packet thus the malicious node is isolated from the network.

6.2 Algorithm for Reputation based IDS:

Step 1: Source broad cast the route request for Destination in the MANET.

Step 2: If node is Malicious node, then re-broadcast the route request with decremented hop_count and higher sequence number.

Step 3: Else node is re-broadcast the re-broadcast the route request with increment the hop_count by one and no change in the sequence number.

Step 3: Upon the receiving route request Destination sends the route reply to the least hop count to the request node. In this case malicious node has higher probability to be part of active route.

Step 5: An active shortest path is established from the source to destination.

Step 6: The nodes which are in the active path are identified.

Step 7: For each previous node Reputation is assigned and maintained by next node in the active path.

Step 8: when data packets are received by the malicious node, it drops the packets for a certain time interval.

Step 9: next node count the number packets dropped at a specific node by the help of counter variable.

Step 10: if $\text{avgvalue} \geq \text{count}$

Step 10.1: $\text{reputation} = \text{reputation} + 1;$

Step 11: else

Step 11.1: $\text{reputation} = \text{reputation} - \text{pow}(2, k);$

Step 11.2: $k++;$

Step 12: if ($\text{reputation} < 0$)

Step 12.1: print node is malicious;

Step 12.2: broadcast error message to the source.

Step 12.3: store the index of the node which is malicious and the next hop in a text file.

Step 13: Every next node in the active path repeat step 9-12 for every periodic time interval the node.

Step 14: Next time whenever a node receives a route request it cross checks the index of the node with the index in the text file. If the request is from a malicious node it free the packets and isolate node from the network.

6.3 Flow chart representation of Reputation based IDS:

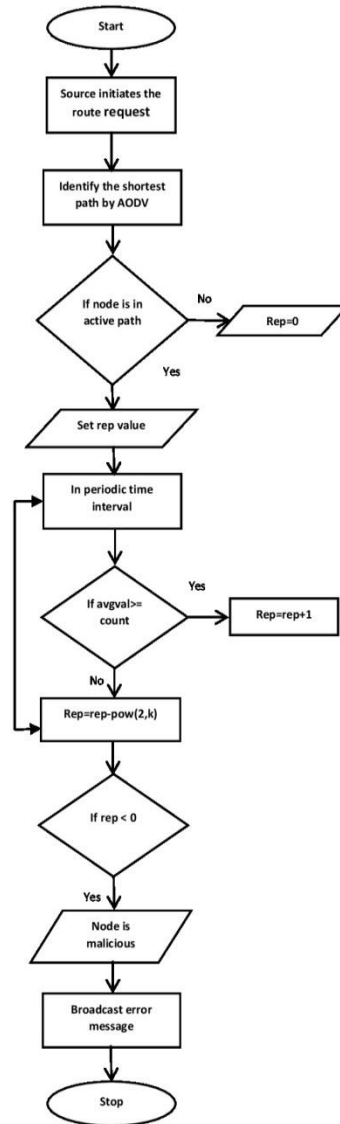


Figure 6.3.1: Flow chart of Reputation based IDS

6.4 Working of Reputation based IDS with an example scenario:

Consider an example MANET with 8 mobile node with S as the source and the D as destination as shown in the figure 6.4.1. Among those 8 mobile nodes c act as malicious node. The communication among those nodes taken place with the help of UDP packets with a constant bit rate.

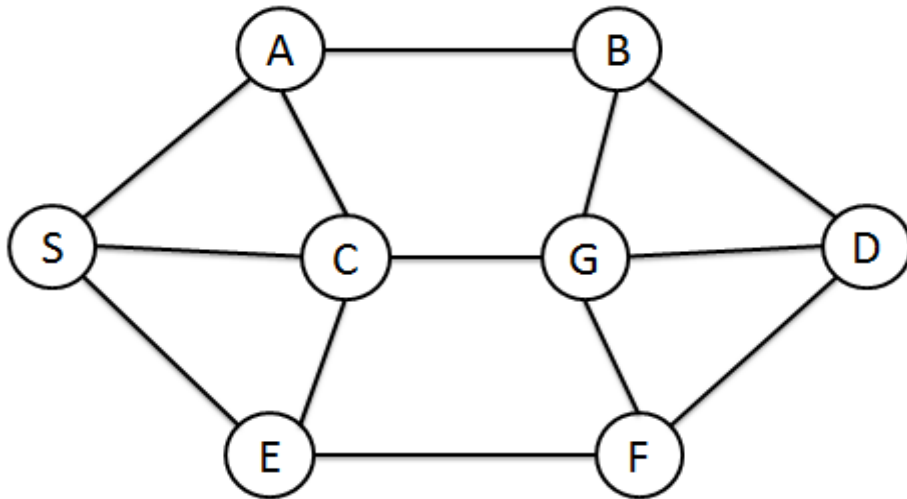


Figure 6.4.1: network topology.

Source S initiates a route request and sends RREQ packets to the neighbouring nodes for a route to the destination D. Since C is the gray hole node it generates a route reply with less hop count and high destination sequence number and it sends the route entry packets without any delay. As the source receives the route reply from the malicious node first than any other node with less hop count it believes that the c has a route to the destination and establish an active path to the destination D which includes c in its active path as shown in figure 6.4.2.

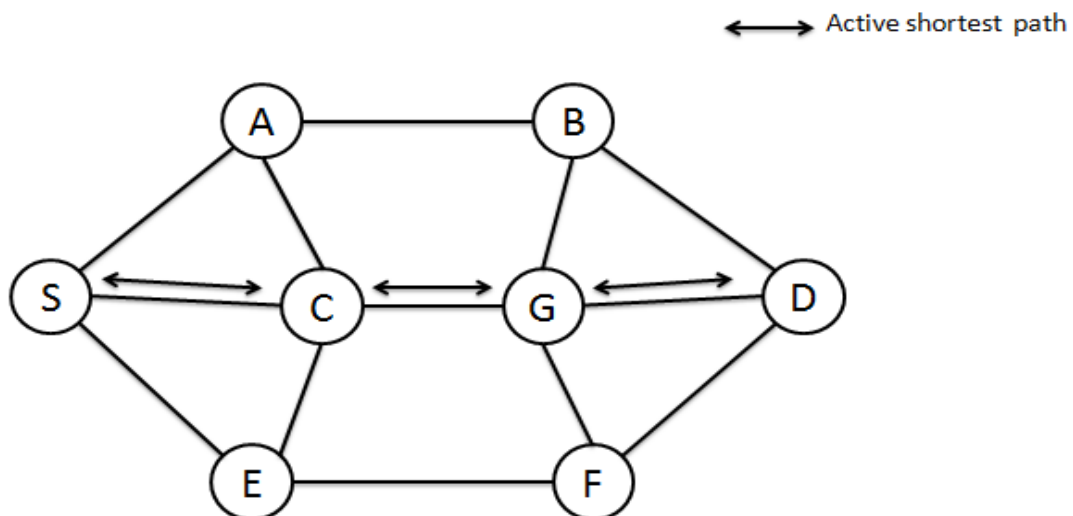


Figure 6.4.2: depiction of an active path

Reputation is assigned to the nodes which were in the active path. Every node maintains its previous node reputation as shown in figure 6.4.3. Here the reputation is assigned to the nodes which were in the active path not to the all the nodes in the network so it implies less memory constraints to the network.

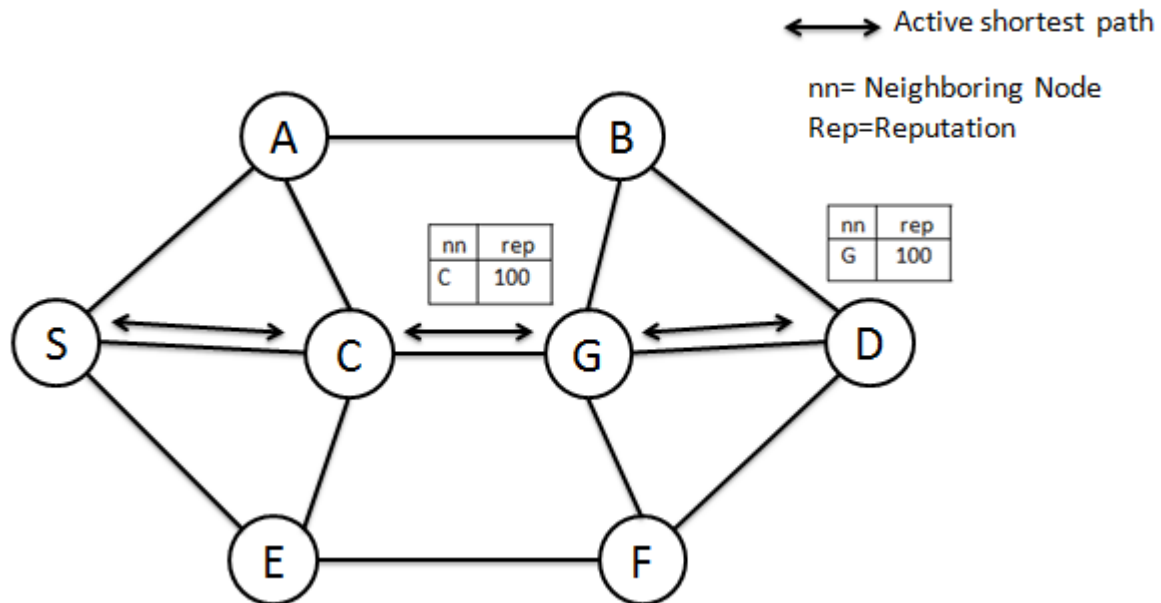


Figure 6.4.3: Assignment of Reputation.

In a periodic time interval every node in the active path observes the previous node and check if the drop rate is greater than the average value if the drop rate is greater than the average value reputation is exponentially decremented else reputation vale is additively incremented. Here as c is the malicious node G observes the C and reduces the reputation every time when it drop the packets at a certain instant of time the reputation value becomes less than zero as shown in the figure 6.4.4. we declare the node as malicious node as isolate it from the network.

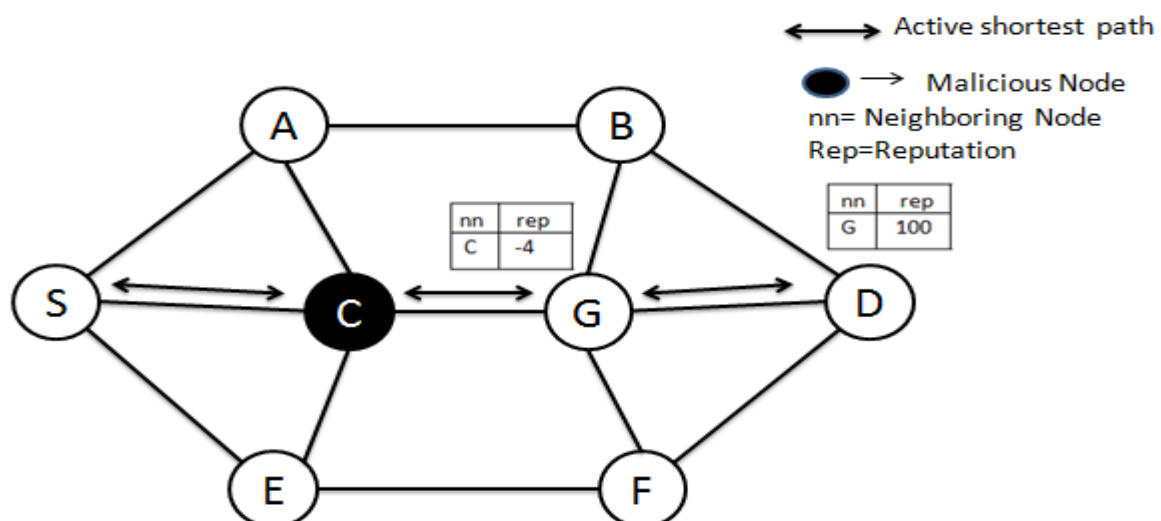


Figure 6.4.4: Detection of malicious node.

7. SYSTEM SPECIFICATIONS

Software requirements:

Simulator	:	NS-2(v-2.33)
MAC Protocol	:	IEEE 802-11
Routing Protocol	:	AODV
Languages	:	Perl, TCL, C++
Operating system	:	Windows XP/7/8/UBUNTU
Environment	:	Cygwin (To create Linux-like Environment in windows)

Minimal Hardware requirements:

Processor	:	Pentium IV
Hard Disk	:	100 GB
RAM	:	2 GB

8. IMPLEMENTATION

The following changes are made in AODV.cc file in route resolve module:

```
void
AODV::rt_resolve(Packet *p) {

struct hdr_cmn *ch = HDR_CMN(p);

struct hdr_ip *ih = HDR_IP(p);

aodv_rt_entry *rt;

ch->xmit_failure_ = aodv_rt_failed_callback;

ch->xmit_failure_data_ = (void*) this;

rt = rtable.rt_lookup(ih->daddr());

if(rt == 0) {

rt = rtable.rt_add(ih->daddr());

}

Packet *rerr1 = Packet::alloc();

struct hdr_aodv_error *re1 = HDR_AODV_ERROR(rerr1);

assert (rt->rt_flags == RTF_DOWN);

re1->DestCount = 0;

re1->unreachable_dst[re1->DestCount] = rt->rt_dst;

re1->unreachable_dst_seqno[re1->DestCount] = rt->rt_seqno;

re1->DestCount += 1;

#ifdef DEBUG

fprintf(stderr, "%s: sending RERR...\n", __FUNCTION__);

#endif

if(rt->rt_flags == RTF_UP) {

assert(rt->rt_hops != INFINITY2);

if((ch->pptype()!=PT_AODV) && (malicious==1000))

{

if(t < CURRENT_TIME)

{

t=t+.1;

drop(p, DROP_RTR_NO_ROUTE);

count++;

if(t1==(int)CURRENT_TIME)

{
```

```

if(avgvalue>=count)
{
printf("%d\n",count);
printf("%d\n",rep);
rep=rep+1;
}
else
{
printf("count1 %d rep1 %d \n",count,rep);
rep=rep-pow(2,k);
k++;
}
count=0;

t1=t1+2;
if(rep<0)
{
FILE *fp;
fp=fopen("reputation1.txt","w");

printf("\nnode is malicious");

fprintf(fp,"malicious %d next node %d\n", index, rt->rt_nexthop);
sendError(rerr1, false);
fclose(fp);
}
}

else
forward(rt, p, 0.8);
}
else
forward(rt, p, NO_DELAY)
}
}

```

8.2 Tcl file:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#=====
#   Simulation parameters setup
#=====
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 7 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 711 ;# X dimension of topography
set val(y) 442 ;# Y dimension of topography
set val(stop) 100.0 ;# time of simulation end

#=====
#   Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#=====
#   Mobile node parameter setup
#=====
$ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channel $chan \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON

#=====
#   Nodes Definition
#=====
#Create 7 nodes
set n0 [$ns node]
$n0 set X_ 344
$n0 set Y_ 244
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 415
```

```

$n1 set Y_ 340
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 528
$n2 set Y_ 342
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 482
$n3 set Y_ 240
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 440
$n4 set Y_ 165
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 549
$n5 set Y_ 163
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 611
$n6 set Y_ 258
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20
#$ns at 0.0 "[$n5 set ragent_] blackhole1"
$ns at 0.0 "[$n1 set ragent_] blackhole2"
$ns at 0.0 "[$n4 set ragent_] blackhole3"
#=====
#      Agents Definition
#=====
#Setup a UDP connection
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set null1 [new Agent/Null]
$ns attach-agent $n6 $null1
$ns connect $udp0 $null1
$udp0 set packetSize_ 1500
#=====
#      Applications Definition
#=====
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 5.0Mb
$cbr0 set random_ null
$ns at 1.0 "$cbr0 start"
$ns at 100.0 "$cbr0 stop"
#=====
#      Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "\"$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run.

```

Simulation Results

Simulation parameters:

PARAMETERS	USED IN SIMULATION
Simulator	NS-2.33
Dos Attack	Gray Hole Attack
Channel Type	Channel / wireless channel
Antenna Type	Antenna / Omni Antenna
Interface Queue Type	PriQueue
Mac Type	MAC/802.11
Protocols Studied	AODV
Simulation Time	100 sec
Simulation Area	1500 * 1500
Traffic Type	CBR(UDP)
Data Payload	1000 Bytes / packet
Number of Malicious Nodes	1
Speed	5 m /sec
Number of Nodes	7

Table 1: Simulation parameters.

9.2 Metrics used for Simulation:

To analyze the performance of reputation based IDS various contexts are created by varying number of nodes, nodes mobility and nodes pause time. The metrics used to evaluate the performance of these contexts are given below:

Good put: Good put is the application level throughput .i.e. the number of useful information bits delivered by the network to a certain destination per unit of time.

Note: the greater the Good put value means the better performance.

Packet Delivery Ratio: The ratio of number packets originated by the "application layer" CBR sources and the number of packets received by the CBR sink at the final destination. This illustrates the level of delivered data to the destination.

Packet Delivery Ratio= Σ Number of packets Receive / Σ Number of packets send.

Note: The greater the packet delivery ratio means the better performance.

End-to-end Delay: The average time taken by a data packet to arrive in the destination. It also includes the delay caused by the route discovery process and the queue in the data packet transmission. Only the data packets that successfully delivered to the destination that counted.

End-to-end Delay= Σ (Arrive Time – send time) / Σ Number of connections.

Note: The lower the value of end to end delay means better performance.

Packet Loss Ratio: Packet loss occurs when one or more packets of data travelling across a computer network fail to reach the destination. Packet loss can be caused by number of factors including signal degradation over the network medium due to multi path fading, packet loss because of channel congestion corrupted packets rejected in transit and faulty networking hardware.

Packet loss= Number of packets send - Number of packets Receive.

The lower the packet loss means the better the performance.

Detection Probability: The ratio of number of detected malicious nodes and the total number of malicious nodes. This metric directly reflects the performance of our detection algorithm.

False positive probability: The ratio of number of honest nodes mistakenly detected as malicious node and the total number of honest nodes.

Note: In a detection algorithm, the false positive ratio should be low.

False negative probability: The ratio of number of malicious nodes mistakenly detected as honest nodes and the total number of malicious nodes.

Note: In a detection algorithm, the false negative ratio should be low.

9.3 TEST CASE:

9.3.1 Network Topology:

Network Topology is created with the help of NS Executable Jar File. It generates a Tcl file with which we will be able to run the network in a network simulator. As shown in figure 9.3.1.1, a network Topology is created with seven mobile nodes with n0 and n6 as source and destination respectively and n1 as the malicious node. The network set up is a wireless channel with all the simulation parameters mentioned in the table 1. The communication between the nodes in the network topology is done with the help of User data gram (UDP) packets with a Constant bit rate (CBR).

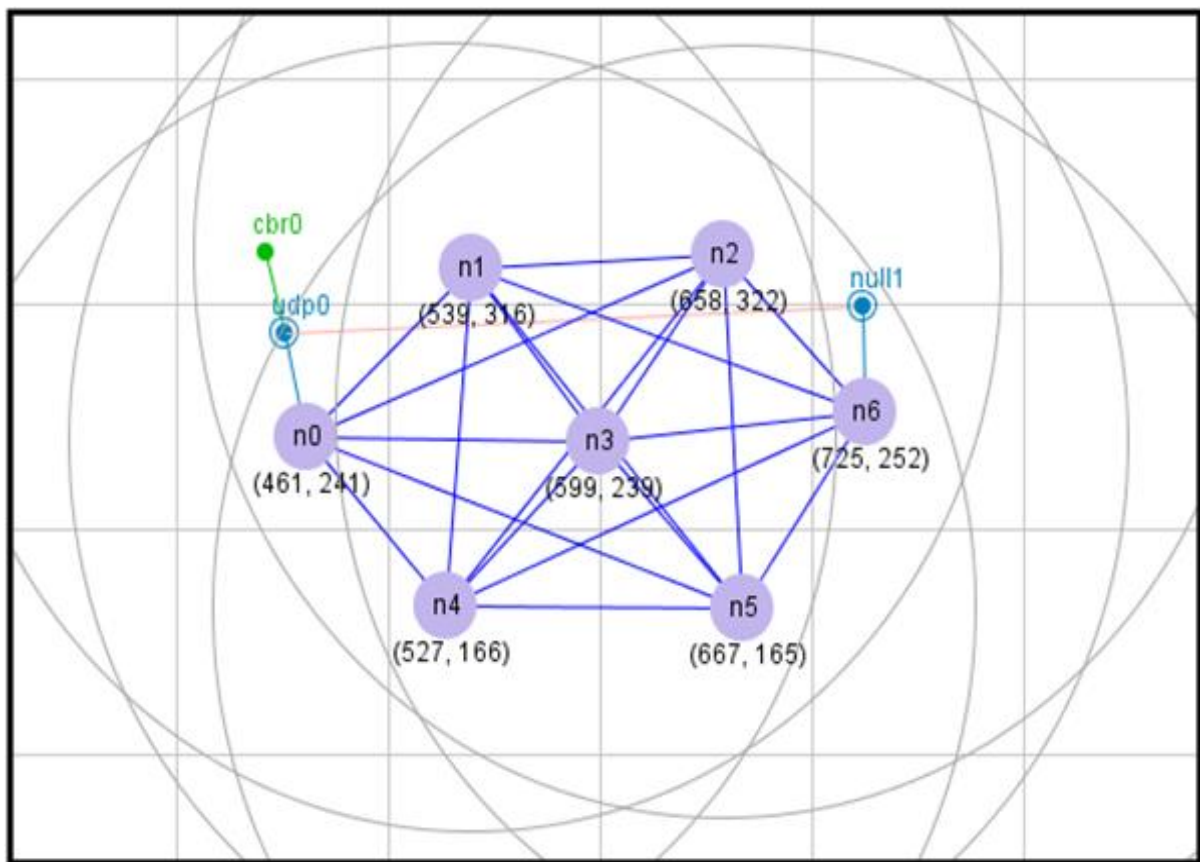


Figure 9.3.1.1: An example scenario of MANET

Here, the generated topology of network is a wireless scenario and an UDP agent is connected to the source node and the simulation starts at 1st second and the simulation stops at 100th sec. Here, the packet size is 1000 sec.

All the nodes in the network Topology make use of Ad-hoc On Demand Distance Vector Routing Protocol.

The MAC Layer Protocol implemented in this network Topology is 802.11 and improved version of "Random Way Point" is used as the mobility node. The link Layer type is Logical Link (LL). The Queue type is Drop Tail / priority queue with maximum number of packets 50.

9.3.2 Gray hole attack in NS-2:

In the first scenario, where there is no malicious node in the network topology a connection is established between the source n0 to destination n6 with n2 and n5 as intermediate nodes. The active path is as shown in the figure 9.3.2.1. In this the source node n0 sends the data packets to destination node n6 via node n2 and node n5 which is the shortest path.

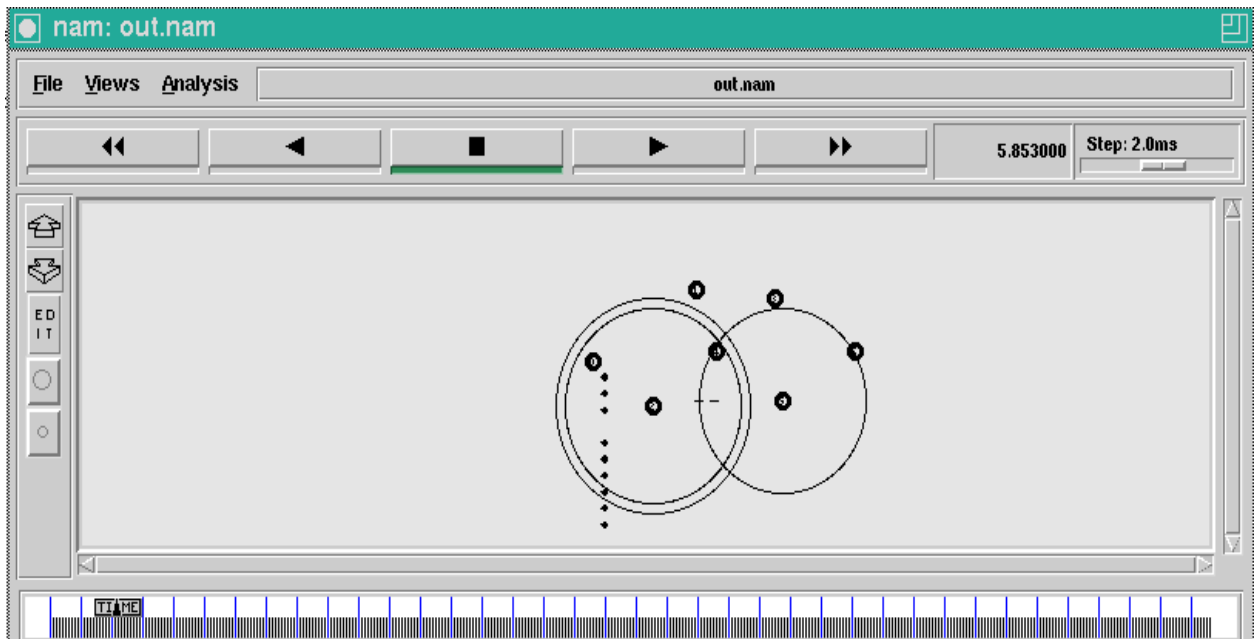


Figure 9.3.2.1: Establishment of active path

In the below scenario 9.3.2.2 we can see the pictorial representation of data packets delivery from node n0 to n6 via intermediate nodes n2 and n5.

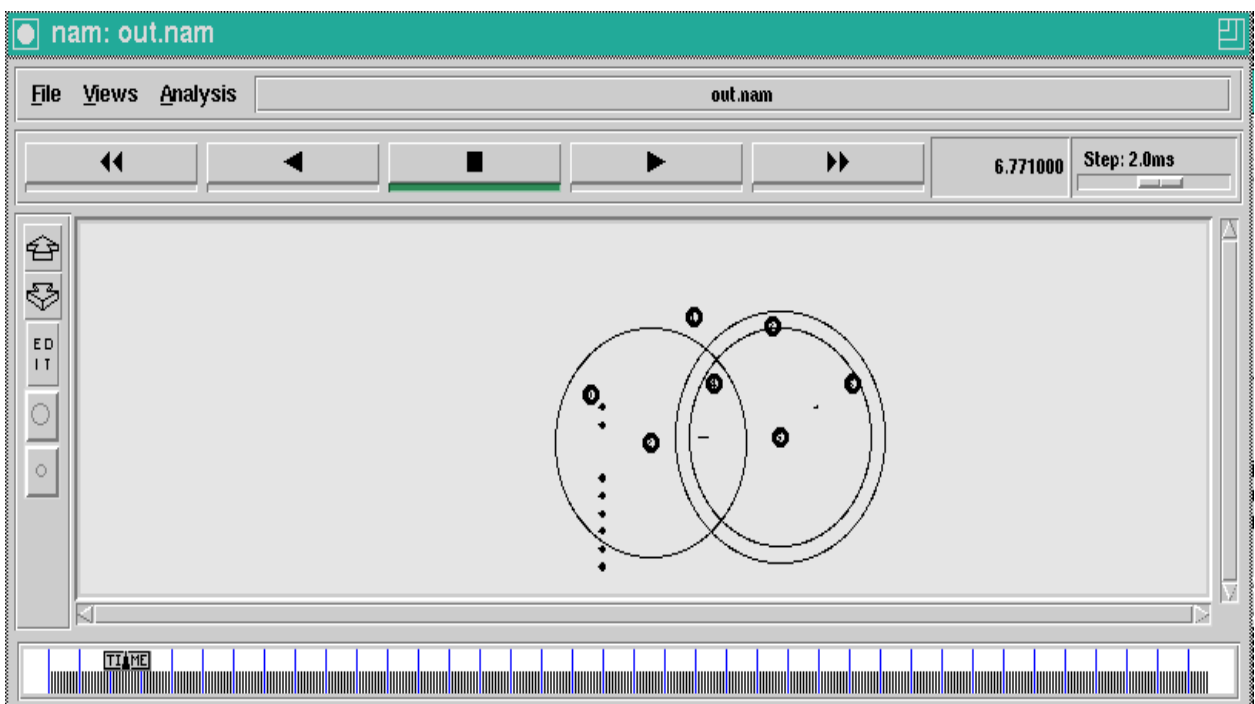


Figure 9.3.2.2: Data flow between nodes

In the second scenario, we add the malicious node to the tcl script by adding the following example block.

```
$ns at 0.0 "[$n5 set ragent_] blackhole1"
$ns at 0.0 "[$n1 set ragent_] blackhole2"
$ns at 0.0 "[$n4 set ragent_] blackhole3"
```

The scenario of the network with malicious node added to the topology can be seen below. In this, we can absorb the gray hole attack behaviour.

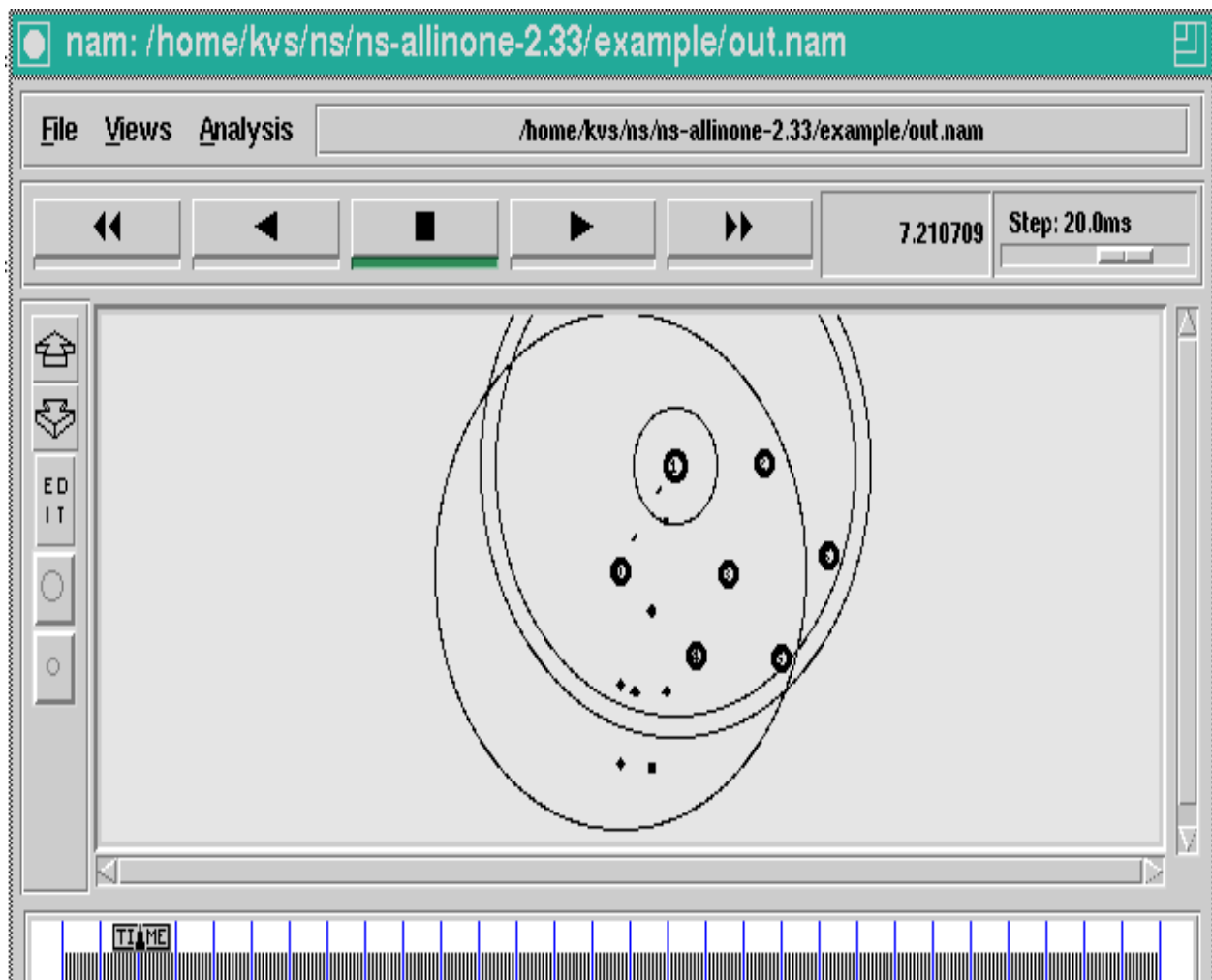


Figure 9.3.2.3: Active path with malicious node

Node 1 being the gray hole node will be able to establish an active path between `n0` and `n6` by acting as the intermediate node. Source `n0` sends the data packets to the `n1` to deliver the data at the destination. But, `n1` being the malicious node, drops the packets intentionally without sending them to destination as shown in the above figure 9.2.2.3 and the gray hole attack is created in the network at node `n1`.

As the Characteristics of the gray hole is it drops subset of the packets and sends some of the packets to the destination we can observe it in the below figure 9.3.2.4 where the node n1 drops a subset of packets and send some of the packets to the destination.

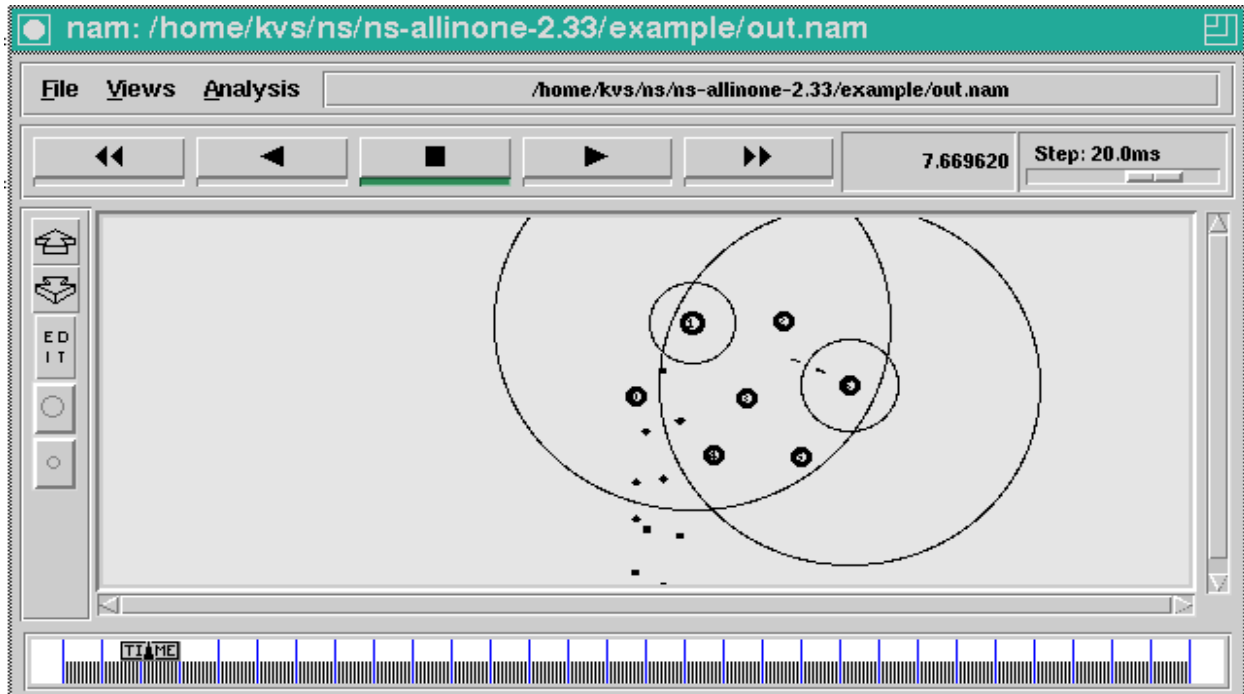


Figure 9.3.2.4: Packet drop at malicious node

We get simulation results in a output trace file which has .tr extension. Trace files include all events in the simulation such as when packets are sent, which node generated them, which node has received, which type of packet is sent, if it is dropped why it is dropped etc. As the node n1 drops the packet intentionally by specifying NOROUTE we can observe it in the below figure 9.3.2.5

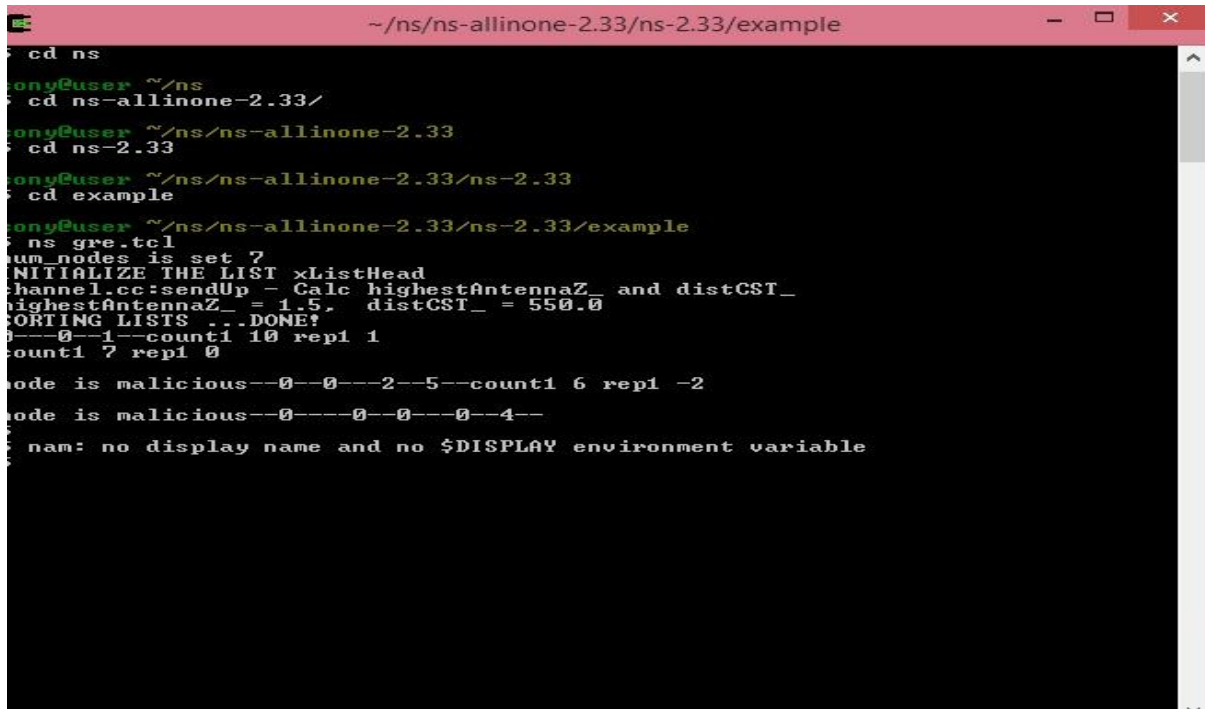
```
s 1.000000000 0 AGT --- 0 cbr 1000 [0 0 0 0] ----- [0:0 6:0 32 0] [0] 0 0
r 1.000000000 0 RTR --- 0 cbr 1000 [0 0 0 0] ----- [0:0 6:0 32 0] [0] 0 0
s 1.000000000 0 RTR --- 0 AODV 48 [0 0 0 0] ----- [0:255 -1:255 30 0] [0x2 1 1 [6 0] [0 4]] (REQUEST)
s 1.000535000 0 MAC --- 0 AODV 106 [0 ffffffff 0 800] ----- [0:255 -1:255 30 0] [0x2 1 1 [6 0] [0 4]] (REQUEST)
s 1.002358106 6 RTR --- 0 AODV 44 [0 0 0 0] ----- [6:255 0:255 30 1] [0x4 1 [6 4] 10.000000] (REPLY)
r 1.006066749 1 MAC --- 0 ARP 28 [0 ffffffff 6 806] ----- [REQUEST 6/6 0/1]
s 1.006176749 1 MAC --- 0 RTS 44 [52e 6 1 0]
s 1.006539457 6 MAC --- 0 CTS 38 [3f4 1 0 0]
s 1.008000000 0 AGT --- 1 cbr 1000 [0 0 0 0] ----- [0:0 6:0 32 0] [1] 0 0
r 1.008000000 0 RTR --- 1 cbr 1000 [0 0 0 0] ----- [0:0 6:0 32 0] [1] 0 0
D 5.635463842 1 RTR NRTE 474 cbr 1020 [13a 1 0 800] ----- [0:0 6:0 29 1] [474] 1 0
D 100.000000000 0 IFQ END 12329 cbr 1020 [0 1 0 800] ----- [0:0 6:0 30 1] [12329] 0 0
D 100.000000000 0 IFQ END 12332 cbr 1020 [0 1 0 800] ----- [0:0 6:0 30 1] [12332] 0 0
```

Figure 9.3.2.5: Sample trace file

9.3.3 Detection of malicious node with the help of reputation based IDs.

The proposed mechanism will identify the malicious node by observing its previous hop in the active path in periodic time interval and if the drop rate is greater than the actual drop rate, it reduces the reputation exponentially. In a certain instant of time, if the reputation becomes less than 0, it declares the node as malicious.

Here, node n2 observes node n1 and reduces the reputation whenever the drop rate is greater than the average drop rate. When the reputation becomes less than 0, it will display node is malicious as shown in the figure 9.3.3.1.



```

~/ns/ns-allinone-2.33/ns-2.33/example
cd ns
ony@user ~/ns
cd ns-allinone-2.33/
ony@user ~/ns/ns-allinone-2.33
cd ns-2.33
ony@user ~/ns/ns-allinone-2.33/ns-2.33
cd example
ony@user ~/ns/ns-allinone-2.33/ns-2.33/example
ns gre.tcl
num_nodes is set 7
INITIALIZE THE LIST xListHead
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
PORTING LISTS ...DONE!
0--0--1--count1 10 repl 1
count1 7 repl 0
node is malicious--0--0--2--5--count1 6 repl -2
node is malicious--0--0--0--0--4--
nam: no display name and no $DISPLAY environment variable

```

Figure 9.3.3.1: Display of malicious node

The index of the node which is malicious and the next hop is stored in a text file and every node maintains the text file. An sample text file is shown in below figure 9.3.3.2:

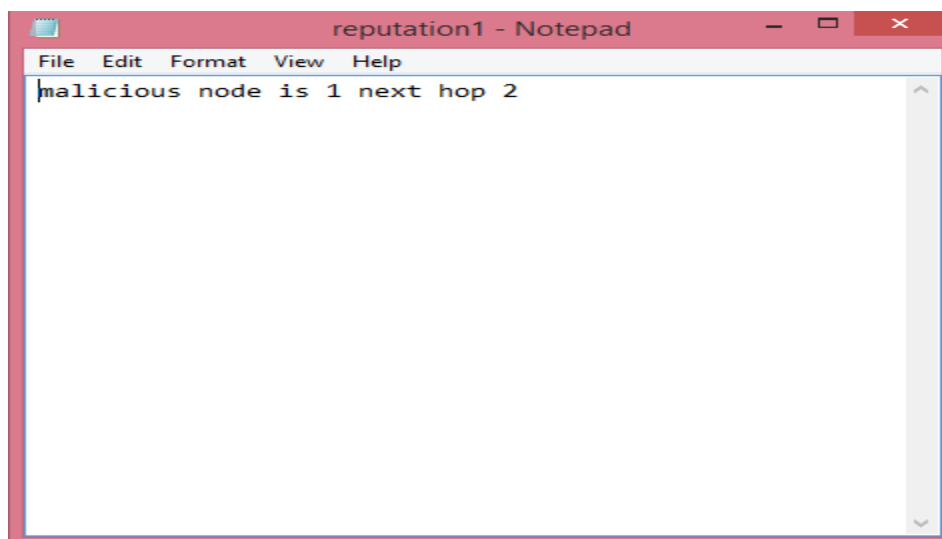


Figure 9.3.3.2: Sample Text File

9.4 RESULT OF THE TEST CASE:

The following table2 gives the result of test case.

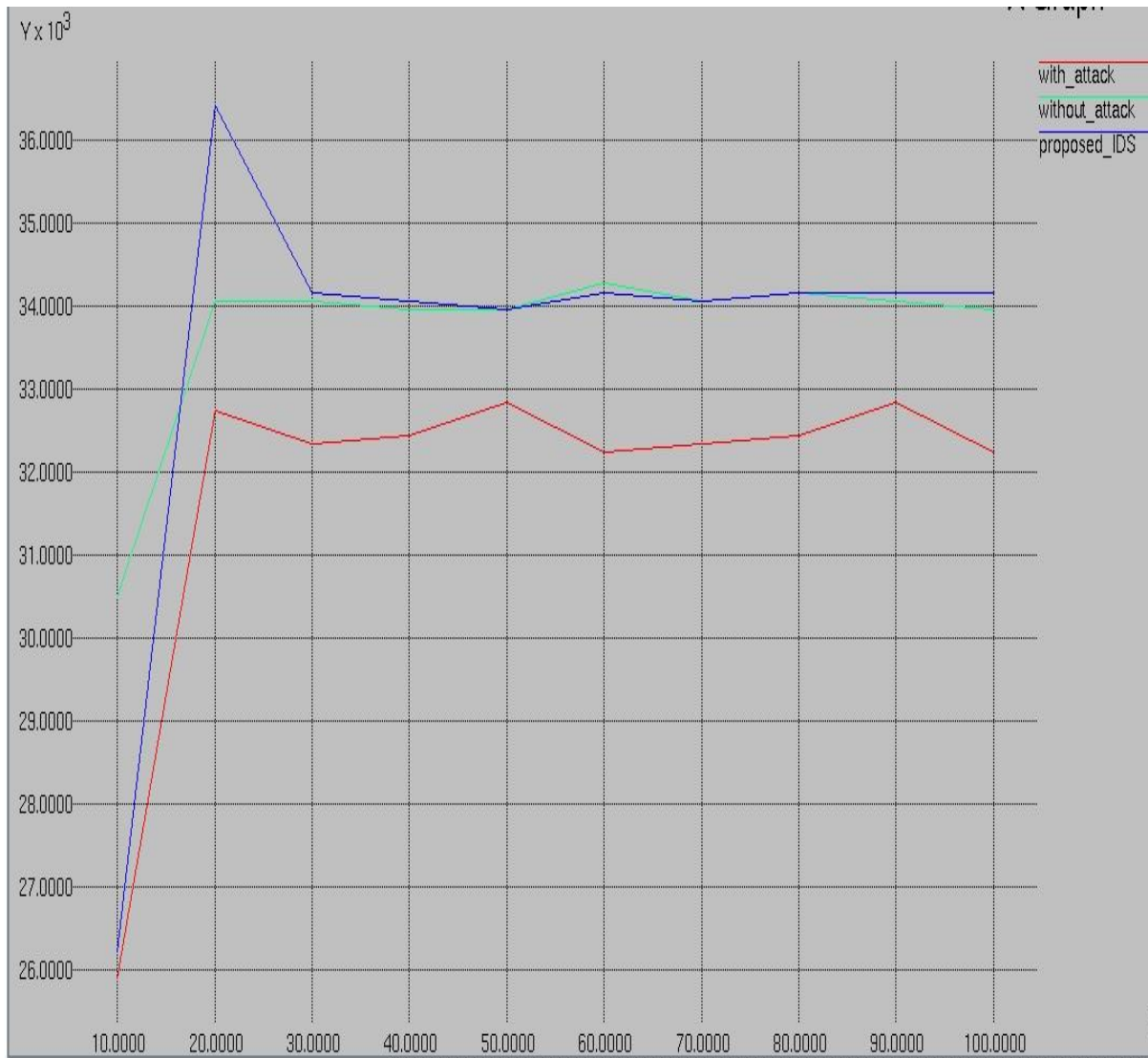
No of packets forwarded	3310
No of packets received	3276
No of packets dropped due to no route	30
No of packets dropped due to collision	4
Simulation time	100 sec

Table2: Result of test case when node n1 is malicious.

10. PERFORMANCE ANALYSIS:

10.1 GOODPUT:

Figure 10.1.1 shows the good put X graph with attack, without attack and proposed IDS.



X-axis: Time
Y-axis: Bit Rate

Figure 10.1.1: Graph for good put.

In the above graph we can observe the sudden raise in the proposed IDS it was due to the buffer packets which reached the destination through the legitimate path when the node is identified as the malicious. After a certain instant of time it will show the same behaviour as of without attack scenario.

The good put is calculated with the help of trace file and perl file as mentioned in Appendix.

10.2 PACKET DELIVERY RATIO:

Figure 10.2.1 shows the packet delivery ratio with attack, without attack and proposed IDS.

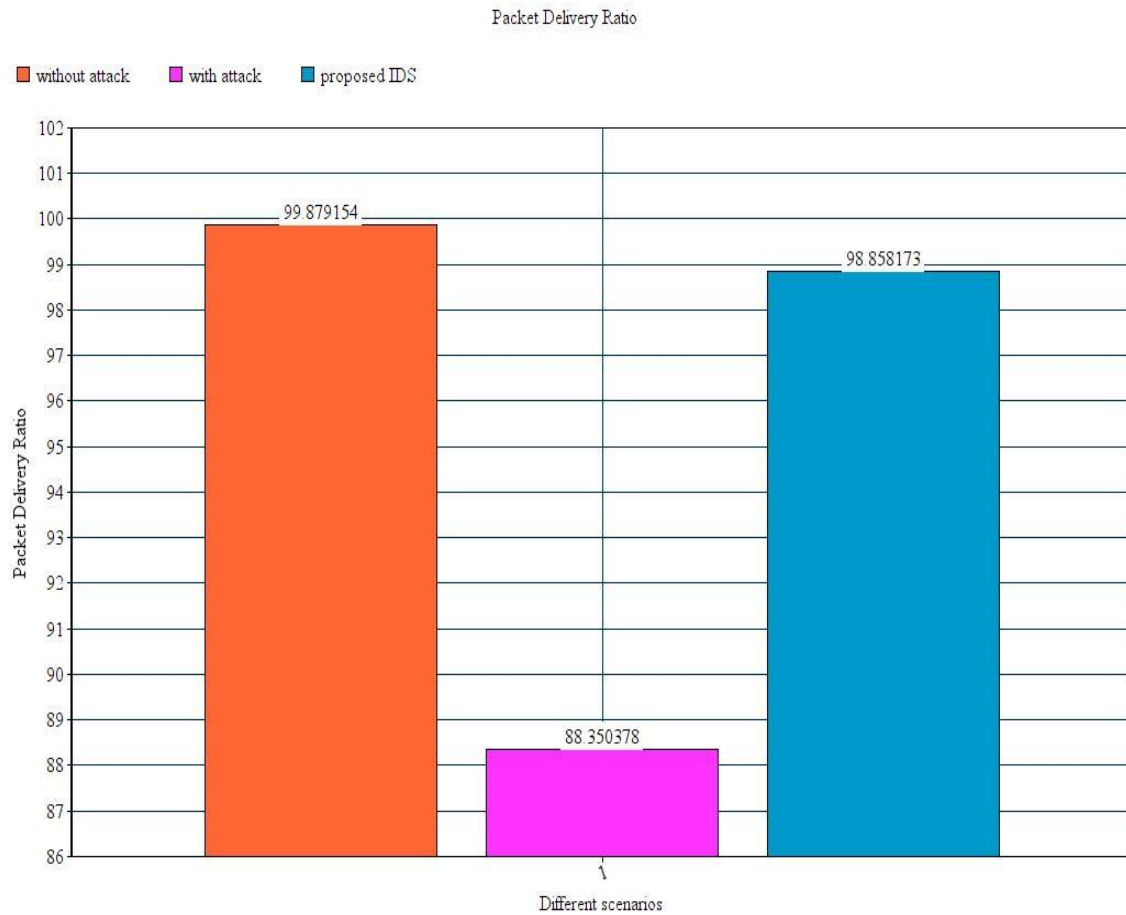


Figure 10.2.1: graph for packet Delivery Ratio.

In the above graph we can observe the packet deliver ratio at the proposed IDS is 98% which almost equal to without attack. Difference of 1% is also due to the packet loss during the time of malicious node detection.

The packet delivery ratio is calculated with the help of trace file and perl file as mentioned in Appendix.

10.3 DETECTION RATE Vs NODE MOBILITY:

Figure 10.3.1 shows the detection rate with varying node mobility.

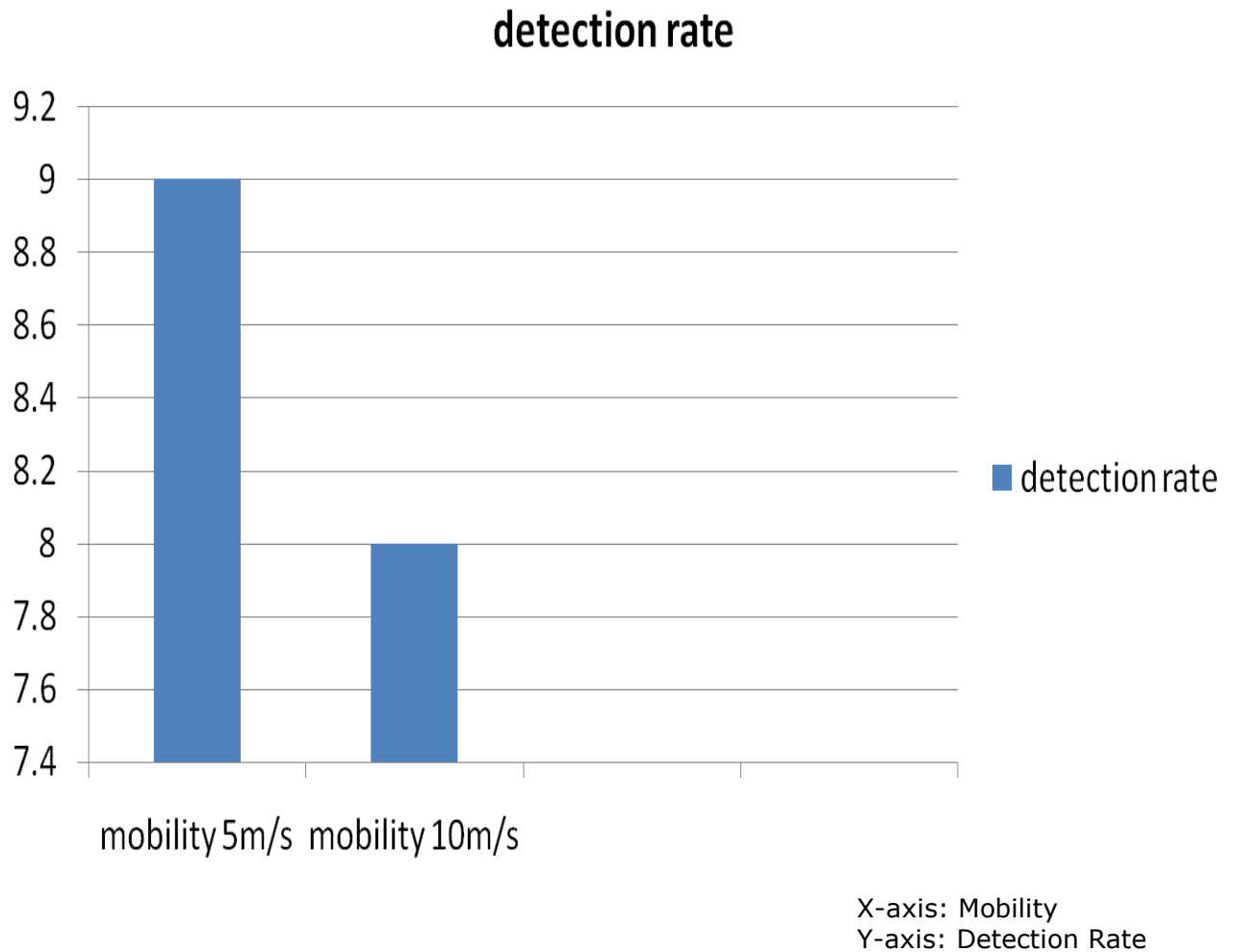


Figure 10.3.1: Graph for Node Mobility Vs Detection Rate.

- When the Node Mobility 5 m/s the detection rate that node is malicious is 90%.
- When the Node Mobility 10 m/s the detection rate that node is malicious is 80%.

11. CONCLUSIONS

In this project we have studied the various routing attacks and their countermeasures in MANETs. Based on the study, we have identified that gray hole attack more difficult to detect and also existing countermeasures have their own limitations to detect gray hole attack in MANETs. We have proposed a node independent reputation based IDS to detect the gray hole attack in active path. The proposed IDS mainly works on two major functionalities such as average number of collisions and additive increase/ exponential decrease of node reputation value. These functionalities help us to detect the gray hole attack more faster and able to keep the high detection rate. The simulation results show that the proposed IDS has 80 percent detection rate 10m/sec node mobility and 90 percent detection rate 5m/sec node mobility, and effective packet delivery ratio 98.8 percent.

12. BIBLIOGRAPHY

1. Vishnu, K., and Amos J. Paul. "Detection and removal of cooperative black/gray hole attack in mobile ad hoc networks." *International Journal of Computer Applications* 1.22 (2010): 38-42.
2. Yi, Ping, et al. "Cross-layer Detection for Blackhole Attack in Wireless Network." *Journal of Computational Information Systems* 8.10 (2012): 4101-4109.
3. Chandure, Onkar V., et al. "Simulation of secure AODV in GRAY hole attack for mobile ad-hoc network." *International Journal of Advances in Engineering & Technology, ISSN* (2012): 2231-1963.
4. Kumar, Avenash, and Meenu Chawla. "Destination based group Gray hole attack detection in MANET through AODV." *IJCSI, ISSN (Online)* (2012): 1694-0814.
5. Gupta, Madhuri, and Krishna Kumar Joshi. "An Innovative Approach to Detect the Gray-Hole Attack in AODV based MANET." *International Journal of Computer Applications* 84.8 (2013).
6. Banerjee, Sukla. "Detection/removal of cooperative black and gray hole attack in mobile ad-hoc networks." *proceedings of the world congress on engineering and computer science*. 2008.6
7. Chandure, Onkar V., and V. T. Gaikwad. "Detection & Prevention of Gray Hole Attack in Mobile Ad-Hoc Network using AODV Routing Protocol." *International Journal of Computer Applications* 41.5 (2012).
8. Kanthe, Ashok M., Dina Simunic, and Ramjee Prasad. "A Mechanism for Gray hole Attack Detection in Mobile Ad-Hoc Networks." *International Journal of Computer Applications* 53.16 (2012).
9. Khattak, Hizbullah, N. Nizamuddin, and Fahad Khurshid. "Preventing black and gray hole attacks in AODV using optimal path routing and hash." *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*. IEEE, 2013.
10. Jain, Sonal, and Sandeep K. Raghuwanshi. "Behavioural and node performance based Grayhole attack Detection and Amputation in AODV protocol." *Advances in Engineering and Technology Research (ICAETR), 2014 International Conference on*. IEEE, 2014.
11. Ahmed, Mariwan, and Muhammad Awais Hussain. "Performance of an IDS in an Adhoc Network under Black Hole and Gray Hole attacks." *Electronics, Communication and Instrumentation (ICECI), 2014 International Conference on*. IEEE, 2014.
12. Zakhary, Sameh R., and Milena Radenkovic. "Reputation-based security protocol for MANETs in highly mobile disconnection-prone environments." *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*. IEEE, 2010.
13. Marti, Sergio, et al. "Mitigating routing misbehavior in mobile ad hoc networks." *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000.
14. Ramaswamy, Sanjay, et al. "Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks." *International conference on wireless networks*. Vol. 2003. 2003.
15. Agrawal, Piyush, Ratan K. Ghosh, and Sajal K. Das. "Cooperative black and gray hole attacks in mobile ad hoc networks." *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. ACM, 2008.
16. Wei, Chen, et al. "A new solution for resisting gray hole attack in mobile ad-hoc networks." *Communications and Networking in China, 2007. CHINACOM'07. Second International Conference on*. IEEE, 2007.
17. Tamilselvan, Latha, and V. Sankaranarayanan. "Prevention of blackhole attack in MANET." *Wireless Broadband and Ultra Wideband Communications, 2007. AusWireless 2007. The 2nd International Conference on*. IEEE, 2007.
18. Sun, Bo, et al. "Detecting black-hole attack in mobile ad hoc networks." *Personal Mobile Communications Conference, 2003. 5th European (Conf. Publ. No. 492)*. IET, 2003.

19. Deng, Hongmei, Wei Li, and Dharma P. Agrawal. "Routing security in wireless ad hoc networks." *Communications Magazine, IEEE* 40.10 (2002): 70-75.
20. Awerbuch, Baruch, et al. "An on-demand secure routing protocol resilient to byzantine failures." *Proceedings of the 1st ACM workshop on Wireless security*. ACM, 2002.
21. <http://ns2codeforblackholeattack.blogspot.in/>
22. <http://www.eexploria.com/manet-mobile-ad-hoc-network-characteristics-and-features/>
23. <https://www.ietf.org/rfc/rfc2501.txt>
24. <http://user.it.uu.se/~erikn/files/DK2-adhoc.pdf>
25. <https://www.ukessays.com/essays/computer-science/the-characteristics-and-applications-of-manets-computer-science-essay.php>
26. <http://acikarsiv.atilim.edu.tr/browse/160/172.pdf>

13. APPENDIX

NETWORK SIMULATOR (NS):

In this work, we have tried to evaluate the effects of Gray hole attack in the wireless ad-hoc Networks. To create a virtual platform for network simulation we make use of NS Network Simulator Program.

NS is an event driven network simulator program, developed at the University of California Berkley, which includes many network objects such as protocols, applications and traffic source behavior. The NS is a part of software of the VINT Project that is supported by DARPA since 1995.

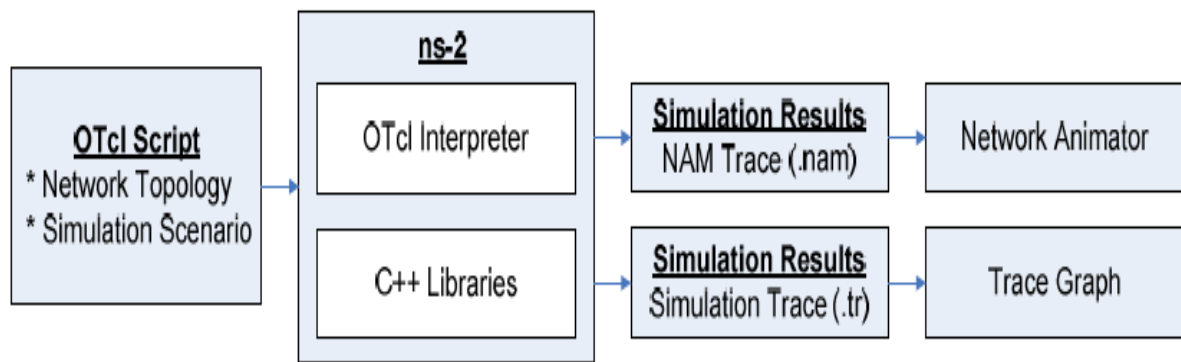


Figure 10.1: NS2 Schema.

At the simulation layer NS uses OTcl (Object Oriented Tool Command Language) programming language to interpret user simulation scripts. OTcl Language is in fact an object oriented extension of the Tcl Language. The Tcl Language is fully compatible with the c++ programming Language. At the top layer, NS is an interpreter of Tcl scripts of the users, they work together with c++ codes.

OTcl script written by a user is interpreted by NS. While OTcl script is being interpreted, NS creates two main analysis reports simultaneously. One of them is NAM(Network Animator) object that shows the visual animation of the simulation. The other is the trace object that consists of the behavior of all objects in the simulation. Both of them are created as a file by NS. Former is .nam file used by NAM software that comes along with NS. Latter is a ".tr" file that includes all simulation traces in the text format.

NS project is normally distributed along with various packages (ns, nam, tcl etc) named as "all-in-one package", but they can also be found and downloaded separately. In this study, we have used version ns-2.33 of ns all-in-one package and installed the package in the windows environment using cygwin. we have written .tcl files in the text editor and analyzed the results in the .tr file. The implementation phase of gray hole behavior to the AODV protocol is written using c++ language.

TCL FILE GENERATION:

Now, we will see in detail about the generation of tcl file in NS Executable Jar File.

TCL Language in NS2:

Short for the Tool Command Language, TCL is a powerful interpreted programming Language developed by John Ousterhout at a University of California, Berkley. TCL is very powerful and dynamic programming language. It has a wide range of usage, including web and desktop applications, networking, administration, Testing etc. Tcl is a truly cross platform, easily deployed and highly extensible.

The significant advantage of Tcl Language is that it is fully compatible with the C programming Language and Tcl Libraries can be interoperated directly into C programs. For generation of Tcl scripts, we have used NS executable jar file.



Figure: Wireless scenario selection

In this, we can have the wired scenario and wireless scenario, we have choose the new wireless scenario of the networks. In this, we can select of UDP as it does not send Acknowledgement to the network. CBR (constant bit rate) is used as the application for the data packets. In this we can have the *start time, stop time, packet size, rate and interval*.

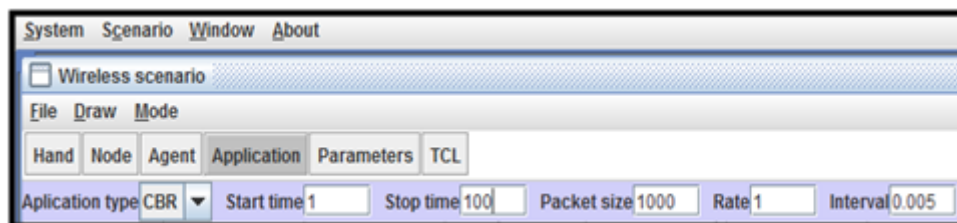


Figure: CBR Application set up

From this we can generate the topology of the network with required nodes. Here we have taken a topology having the application of CBR, and the UDP agent connected to the source and the null node connected to the destination and set the start time=1sec stop time=100sec packet size=1000 Here, we connected six nodes and the topology is created using the parameters of wireless network using AODV Routing protocol.

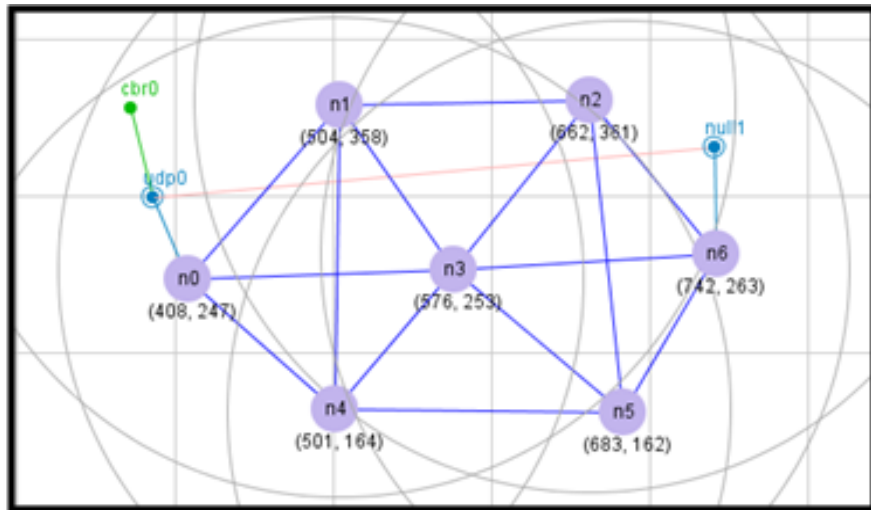


Figure: Network Topology

In the simulation parameter set up we can have Wireless Channel, MAC protocol of 802.11. Maximum packets in the queue 50. AODV routing protocol is used for the shortest path from source to destination and Agent Trace, Router Trace, Mac Trace, Movement Trace set to be on.

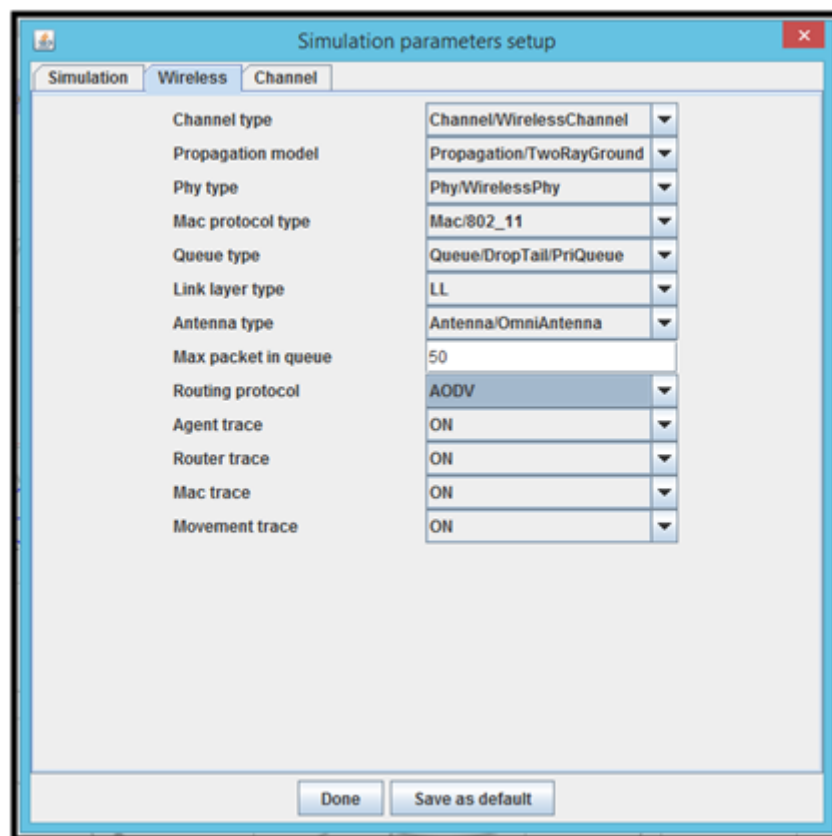


Figure: Simulation Parameter set up

Good put Perl Script:

To calculate the good put of network, we used the Perl script which has ".pl" extension. Perl is a programming language which can be used for a large variety of tasks. A typical simple use of Perl would be for extracting information from a text file and printing out a report or for converting a text into another form. Perl is implemented as an interpreted language. Thus, the execution of a Perl script tends to use more CPU time than a corresponding c program for instance. Computer tends to get faster and faster, and writing something in Perl which can save time. The following is the Perl script which we have implemented to calculate good put.

```
# type: perl throughput.pl <trace file> <required node> <granlarity> >      output file
$infile=$ARGV[0];
$granularity=$ARGV[1];
#we compute how many bytes were transmitted during time interval specified
#by granularity parameter in seconds
$sum=0;
$clock=0;
open (DATA,"<$infile")
|| die "Can't open $infile $!";
while (<DATA>) {
    @x = split(' ');
    #column 1 is time
    if ($x[1]-$clock <= $granularity)
    {
        #checking if the event corresponds to a reception
        if ($x[0] eq 'D')
        {
            if ($x[3] eq 'RTR')
            {
                #checking if the destination corresponds to arg 1
                if ($x[2] ne "_0_") #change according to your source node
                {
                    #checking if the packet type is CBR
                    if ($x[6] eq 'cbr')
                    {
                        $sum=$sum+1;
                    }
                }
            }
        }
    }
    else
    {
        #throughput=$sum/$granularity;
        print STDOUT "$x[1] $sum\n";
        $clock=$clock+$granularity;
        $sum=0;
    }
}

#throughput=$sum/$granularity;
print STDOUT "$x[1] $sum\n";
$clock=$clock+$granularity;
$sum=0;
close DATA;
exit(0);
```

To calculate the good put in NS-2 simulator :

perl goodput.pl <trace file> <required node> <granlarity> > output file

Packet Delivery Ratio Perl Script: The following is the Perl script which we have used to calculate packet delivery ratio of network in NS-2.

```
# type: perl throughput.pl <trace file> <required node> <granularity> >      output file
$infile=$ARGV[0];
$snode1=$ARGV[1];
$snode2=$ARGV[2];
$snode3=$ARGV[3];
$snode4=$ARGV[4];
$dnode=$ARGV[5];

#we compute how many bytes were transmitted during time interval specified
#by granularity parameter in seconds
$count=0;
$rcount=0;
$pdr=0;
$clock=0;

open (DATA,"<$infile")
|| die "Can't open $infile $!";
while (<DATA>) {
    @x = split(' ');

    #column 1 is time

    #checking if the event corresponds to a reception
    if ($x[0] eq 's')
    {
        if ($x[3] eq 'MAC')
        {
            #checking if the destination corresponds to arg 1
            if (($x[2] eq $snode1) || ($x[2] eq $snode2) || ($x[2] eq $snode3) || ($x[2] eq $snode4)) #change according
            to your destination node
            {
                #checking if the packet type is TCP
                if ($x[6] eq 'cbr')
                {
                    $count=$count+1;
                }
            }
        }

        if ($x[0] eq 'r')
        {
            if ($x[3] eq 'MAC')
            {
                #checking if the destination corresponds to arg 1
                if ($x[2] eq $dnode) #change according to your destination node
                {
                    #checking if the packet type is TCP
                    if ($x[6] eq 'cbr')
                    {
                        $rcount=$rcount+1;
                    }
                }
            }
        }

        $pdr=$rcount/$count;
        print STDOUT "PDR $pdr \n scount $count rcount $rcount\n";
        $rcount=0;

        close DATA;
    }
    exit(0);
}

command to calculate PDR: $ perl pdr.pl trailable_name sour-node dest_node
>fname.
```