Flood Tolerant AODV Protocol (FT-AODV)

Pratik Singh
Computer Science, Institute of Engineering and Management, Kolkata, West Bengal, India

Aman Raj
Information Technology, Institute of Engineering and Management, Kolkata, West Bengal, India

Debdutta Chatterjee
Information Technology, Institute of Engineering and Management, Kolkata, West Bengal, India

ABSTRACT

An ad hoc network is a collection of wireless computers with dynamically changing topology, communicating among themselves over possibly multi-hop paths, without the help of any infrastructure. Although many ad hoc network routing protocols have been proposed (AODV, SAODV, DSR etc), none of them considers or solves the security problems efficiently within the restrictions of ad hoc networks. Ad hoc networks are vulnerable to many types of attack like Denial Of Service (DOS), Byzantine Attack, Black-hole Attack, Flooding Attack, etc. In this paper we put forward an efficient and reliable security mechanism based on the AODV routing protocol which protects the ad hoc networks from different types of flooding attacks.

General Terms

Mobile Ad Hoc Network (MANET) protocol, Wireless security protocol.

Keywords

Wireless networks, protocols, ad hoc network, security, flooding attacks, aodv, protocol.

1. INTRODUCTION

A Mobile Ad Hoc Network (MANET) [1] represents a infrastructure less distributed system that comprises wireless mobile nodes that can freely and dynamically self organize into arbitrary and temporary “ad-hoc” network topologies, allowing devices to seamlessly inter-network with no pre-existing communication infrastructure. Comparably, infrastructure wired or wireless networks refer to networks that possess communication infrastructures (e.g. Routers, gateways, base-stations, etc).All communication and control functionalities in such networks are through these infrastructures. The internet and traditional cellular wireless networks are typical examples of infrastructure networks. Though early applications of MANETs were military based, new applications appear in emergency services such as disaster management, environmental monitoring, search and rescue operations and sensor networks.

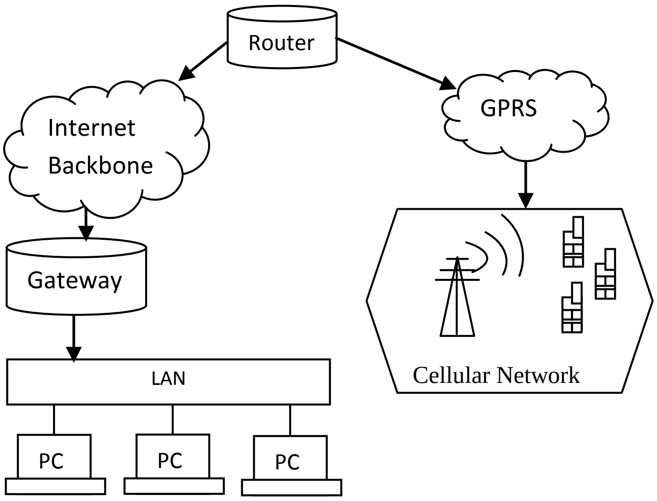


Fig 1: Infrastructure Networks

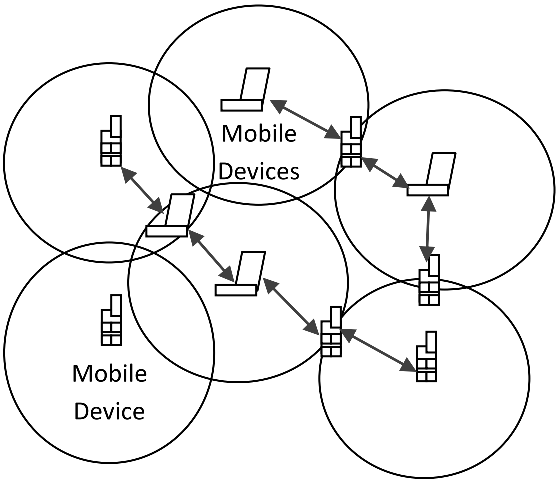


Fig 2: Infrastructure-less Networks (MANETS)

Mobile ad hoc network has been a challenging research area for the last few years because of its dynamic topology, power constraints, limited range of each mobile host's wireless transmissions and security issues etc [2]. If we consider only a stand-alone MANET then it has limited applications, because the connectivity is limited to itself. MANET user can have better utilization of network resources only when it is connected to the Internet. But, global connectivity adds new security threats to the existing active and passive attacks on MANET. Because we have to consider the attacks on access point also through which MANET is connected to Internet [1].

1.1 Challenges in MANETs

MANETS is particularly challenging due to its unique features such as [2- 11]:

1.1.1 *Network Topology*

The network topology is constantly changing as a result of nodes joining in and moving out.

1.1.2 *Decentralization*

Network functions (e.g. routing, authentication) are carried out by individual nodes in a decentralised manner.

1.1.3 *Infrastructure*

MANETS have no underlying infrastructure (e.g. base stations, access points) traditionally seen as infrastructure networks.

1.1.4 *Limited Resources*

Network nodes have limited resources (e.g. battery power, CPU capacity, memory and bandwidth).

1.1.5 *Unreliable*

Wireless links between nodes are unreliable.

1.1.6 *Certification Authority*

Absence of a certification authority and centralized monitoring or management point.

All the features mentioned above together make securing MANETs a challenging issue. Traditionally security mechanisms used in infrastructure networks may be inapplicable to MANETs because of the dynamic and transient nature of MANETs. MANETs suffer from not only the same kinds of vulnerabilities as their infrastructure counterparts, but also peculiar threats and attacks (e.g. sleep deprivation, black hole attack, selfish misbehaving and Denial Of Service (DOS) attacks) caused by unique characteristics of MANETs.

1.2 Overview of different types of attack in MANETs

Based on the actions performed, attacks can be classified as passive or active attacks [12] [13].

Active attacks: This type of attack involves actions that are actively performed to disturb network normal services. For example, adversaries can impede MANET routing functionality by improperly modifying, relaying, injecting and discarding control packets.

Passive attacks: This type of attacks attempts to discover valuable information, which can be exploited later to launch active attacks. Typical methods by passive attackers are eavesdropping, traffic monitoring and analysis.

Attacks can also be categorised into internal or external attacks according to the domain of attacks.

External attack: External attack is caused by nodes that do not belong to the network. They typically aim to cause to congestion, propagate incorrect routing information, prevent services from working properly, or shut them down.

Internal Attacks: Internal Attacks are launched from inside by compromised or hijacked nodes that belong to the network. Internal Attacks are more severe attacks because such compromised nodes belong to the network as authorized parties and are thus protected by the security mechanisms and underlying services.

The classification of the network protocol stack is shown in the following table.

Table 1. Layers And Associated Attacks

Protocol layer	**Types of attacks observed**
Physical Layer	Jamming, Eavesdropping
Data Link Layer	Traffic Analysis, Monitoring
Network Layer	Byzantine, Flooding, Wormhole, Black hole
Transport Layer	SYN Flooding, Hijacking
Application Layer	Data Corruption

Each layer and their vulnerabilities to specific attacks are discussed below:

1.2.1 *Physical Layer Attacks:*

Eavesdropping is the intercepting and reading of messages and conversations by unintended receivers. The mobile hosts in mobile ad hoc networks share a wireless medium. The majorities of wireless communications use the RF spectrum and broadcast by nature. Signals broadcast over airwaves can be easily intercepted with receivers tuned to the proper frequency. Thus messages transmitted can be eavesdropped, and fake messages can be injected into network. Moreover a radio signal can be jammed or interfered, which causes the messages to be corrupted or lost. If the attacker has a powerful transmitter, a signal can be generated that will be strong enough to overwhelm the target signals and disrupt communications. The most common types of this form of signal jamming are random noise and pulse. Jamming equipment is readily available. In addition, jamming attacks can be mounted from a location remote to the target networks.

1.2.2 *Link Layer Attacks:*

MANET is an open peer-to-peer network architecture. Specifically, one-hop connectivity among neighbors is maintained by the link layer protocols, and the network layer protocols extend the connectivity to the other nodes in the network. Attacks may target the link layer by disrupting the cooperation of the layer's protocols.

1.2.3 *Network Layer Attacks:*

A variety of attacks targeting the network layer have been identified and heavily studied in research papers. By attacking the routing protocols attackers can absorb network traffic, inject themselves into the path between the source and destination, and thus control the network traffic flow. A malicious node can inject itself into the path of communication between any two nodes in the network. The packets can be forwarded to non optimal paths that can induce significant delay or it can be discarded. Attackers can create routing loops, introduce severe network congestion and channel contention in certain areas. Multiple colluding attackers can render the source node unable to find the path to a destination node and hence performance degradation. There are some other advanced attacks in the network layer like:

Byzantine attack: In this type of attack, a compromised intermediate node or a set of compromised intermediate nodes works in collusion and carries out attacks such as creating routing loops, forwarding packets on non-optimal paths and selectively dropping packets which results in disruption or degradation of the routing services. It is hard to detect byzantine failures. The network would seem to be operating normally in the viewpoint of the nodes, though it may actually be showing Byzantine behavior [14].

Black hole attack: In this attack, an attacker uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. An attacker listen the requests for routes in a flooding based protocol. When the attacker receives a request for a route to the destination node, it creates a reply consisting of an extremely short route. If the malicious reply reaches the initiating node before the reply from the actual node, a fake route gets created. Once the malicious device has been able to insert itself between the communicating nodes, it is able to do anything with the packets passing between them. It can drop the packets between them to perform a denial-of- service attack, or alternatively use its place on the route as the first step in a man-in-the-middle attack.

Wormhole attack: In wormhole attack, a malicious node receives packets at one location in the network and tunnels them to another location in the network, where these packets are resent into the network. This tunnel between two colluding attackers is referred to as a wormhole. It could be established through wired link between two colluding attackers or through a single long-range wireless link. In this form of attack the attacker may create a wormhole even for packets not addressed to itself because of broadcast nature of the radio channel. The wormhole attack is particularly dangerous for many ad hoc network routing protocols in which the nodes that hear a packet transmission directly from some node consider themselves to be in range of (and thus a neighbor of) that node.

For example, when used against an on-demand routing protocols such as DSR [15], a powerful application of the wormhole attack can be mounted by tunneling each route request packet directly to the destination target node of the request. When the destination node’s neighbors hear this request packet, they will follow normal routing protocol processing to rebroadcast that copy of the request and then discard without processing all other received route request packets originating from this same route discovery. This attack thus prevents any routes other than through the wormhole from being discovered, and if the attacker is near the initiator of the route discovery. This attack can even prevent routes more than two hops long from being discovered. Possible ways for the attacker to then exploit the wormhole include discarding rather than forwarding all data packets, thereby creating a permanent Denial-of-Service attack or selectively discarding or modifying certain data packets. So, if proper mechanisms are not employed to protect the network from wormhole attacks, most of the existing routing protocols for ad hoc wireless networks may fail to find valid routes.

1.2.4 *Transport Layer Attacks:*

The objectives of the TCP-like Transport layer protocols in MANET include setting up of end-to-end reliable connection. Similar to TCP protocols in internet, the mobile node is vulnerable to the classic SYN flooding attack or session hijacking attacks. The SYN attack is a typical DOS attack. In this form of attack, a malicious node sends a large amount of SYN packets to a victim node, spoofing the return addresses of the SYN packets. The SYN-ACK packets are sent out from the victim right after it receives the SYN packets from the attacker and then the victim waits for the response of ACK packet. Without any response of ACK packets, the half-open data structure remains in the victim node. If the victim node stores these half-opened connections in a fixed-size table while it awaits the acknowledgement of the three-way handshake, all of these pending connections could overflow the buffer, and the victim node would not be able to accept any other legitimate attempts to open a connection. Normally there is a time-out associated with a pending connection, so the half-open connections will eventually expire and the victim node will recover. However, malicious nodes can simply continue sending packets that request new connections faster than the expiration of pending connections.

1.2.5 *Application Layer Attacks:*

There is an attack that is specific to application layer and a brief description about it is given below:

Multi-layer attacks: Multi-layer attacks are those that could occur in any layer of the network protocol stack. Denial of service and impersonation are some of the common multi-layer attacks. Here we will discuss some of the multi-layer attacks in ad hoc wireless networks.

Denial of service: A denial of service (DOS) attack can be carried out in many ways. The classic way is to flood packets to any centralized resource present in the network so that the resource is no longer available to nodes in the network, as a result of which the network no longer operating in the manner it was designed to operate. This may lead to a failure in the delivery of guaranteed services to the end users. Due to the unique characteristics of ad hoc wireless networks, there exist many more ways to launch a DOS attack in such a network, which would not be possible in wired networks. DOS attacks can be launched against any layer in the network protocol stack. On the physical and MAC layers, an adversary could employ jamming signals which disrupt the on-going transmissions on the wireless channel. On the network layer, an adversary could take part in the routing process and exploit the routing protocol to disrupt the normal functioning of the network. For example, an adversary node could participate in a session but simply drop a certain number of packets, which may lead to degradation in the quality of service (QOS) being offered by the network. On the higher layers, an adversary could bring down critical services such as the key management service.

Impersonation attack: Impersonation attack are just the first step for most attacks, and used to launch further sophisticated attacks. For example, a malicious node can precede an attack by altering its MAC or IP address.

Man-in-the-middle attack: A man-in-the-middle attack is an example of impersonation attack. Here, the attacker reads and possibly modifies messages between two end nodes without letting either of them know that they have been attacked. Suppose two nodes A and B are communicating with each other; the attacker impersonates node B with respect to node A and impersonates node A with respect to node B, exploiting the lack of third-party authentication of the communication between nodes A and B.

2. AODV ROUTING PROTOCOL

AODV is a method of routing messages between mobile computers. It allows these mobile computers, or nodes, to pass messages through their neighbors to nodes with which they cannot directly communicate. AODV does this by discovering the routes along which messages can be passed. AODV makes sure these routes do not contain loops and tries to find the shortest route possible. AODV is also able to handle changes in routes and can create new routes if there is an error. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes. AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it uncast a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.[23]

As the RREP packet propagates back to the source, nodes set up forward pointers to the destination node. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery. We now give a pseudo code representation of AODV, which is very simple and straightforward to understand. We have made some assumptions about the nomenclature we are going to use in the pseudo code which are as follows:

2.1 Terminologies Used:

RREQ: Route Request Packet
RREP: Route Reply Packet
RERR: Route Error Packet

//S is source node,

//D is the destination,

//RT(X) stands for routing table of node X,

//N denotes current node,

//RREQ contains the destination address (DestAddr), sequence number (SeqNo), broadcast id (Bid).

/*S wants to communicate with D, then sequence of action will be as follows:*/

2.2 AODV Algorithm:

```
//Start

If  RT (S) contains a route to D
        {
         S establishes communication with D.
        }

Else
        S creates a RREQ packet and broadcasts it to all its
        neighbors.

End If

For all nodes receiving RREQ

        If  RREQ is invalid
                        {
                        Discard RREQ.
                        }
        If  D is N
                {
                Send a RREP back to the Sender.
                }

        Else If   N has a route to D with
                 SeqNo>=RREQ.SeqNo
                {
                Send a RREP back to the Sender.
                }

                Else
                {
                Record the node from which RREQ was
                received in RT and broadcast RREQ.
                }
                End If
        End For
    While   N receives RREP    and    N is Not S
        {
        Forward RREP on the reverse path.
        Store information about the node sending RREP in RT.
        }
    End While
    S receives the RREP and updates its RT
    S communicates with D
    //End
```

3. MOTIVATION OF OUR ALGORITHM

Many approaches have been suggested to control and prevent network jamming attacks in ad hoc networks. However, none of the approaches have been able to defend themselves against flooding attacks. Most of the algorithms give high priority to control packets over data packets and serve the data packets in first-in-first-out queue. But most attackers flood the network with tremendous RREQ packets. The ramifications of this action are that the attackers succeed to slow down the network and hence most approaches or preventive algorithms fail to defend the network. We here put forward an elegant and efficient scheme which defends itself from network congestion due to network jamming. We will specifically try to provide a resilient defense mechanism for the following cases of flooding attacks:

3.1 Case a

Attacker floods the network with excessive RREQ packets while keeping the same IP address.

3.2 Case b

Attacker floods the network with excessive RREQ packets while changing the IP address using IP spoofing.

3.3 Case c

Attacker floods the network with excessive data packets while keeping the same IP address.

3.4 Case d

Attacker floods the network with excessive data packets while changing the IP address using IP spoofing.

4. PROPOSED SCHEME: FLOOD TOLERANT AODV (FT- AODV)

The basic idea of our scheme is as follows:

a. When the system is under attack, we give packet processing priority to those nodes which have a longer history of stable behavior.

b. We drop the packets of misbehaving nodes.

c. The new comer initially gets a new priority but overtime after showing stable behavior, their priority increases.

d. Using this approach we spend more time and resources in processing only the packets and minimizing the processing attacks-packets or useless packets improving the QOS.

Our scheme assumes AODV for routing. The goal of our scheme is to provide a resilient defense mechanism against network jamming due to flooding attacks. We divide the single packet buffer into three different queues:

Immediate Processing Queue [IPQ], RREQ Request Processing Queue [RREQQ], and Data Processing Queue [DPQ].

4.1 Immediate Processing Queue [IPQ]:

Any RRER or RREP packet arriving at the node enters this node.

4.2 RREQ Request Processing Queue [RREQQ]:

Any RREQ packet arriving at the node enters this queue.

4.3 Data Processing Queue [DPQ]:

Any Data packet arriving at the node enters this queue.

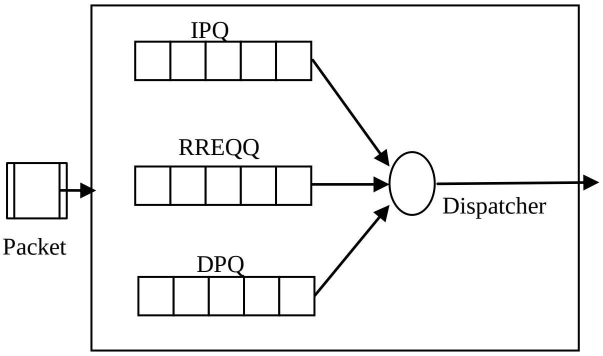


Fig 3: Buffer System of Our Scheme

Our scheme also suggests every node to maintain a data structure to monitor the traffic history of other nodes in the network. Let’s call it Traffic History List [THL] .Every node in THL has the following fields:

Source Address [Addr], No. Of Useful Data Packet Received [D_{packet}], and Time Of Receipt Of Last RREQ Packet [T_{RREQ}]

Whenever a node receives a RREQ or DATA packet, the source address of that packet is evaluated, and the THL list entry of the corresponding source address is updated. If no previous entry of that source address exists, a new entry is added in THL. Therefore, whenever a new node joins the network, a new entry is made in THL. Some terminologies used in our algorithm have been explained below:

RREQ_RATE_LIMIT: In order to reduce the congestion of a network, the AODV protocol adopts some methods. A node cannot originate RREQ messages per second more than the RREQ_RATE_LIMIT. After broadcasting, a RREQ, the node waits for RREP. If a route is not received within round trip milliseconds, the node may try to discover a route by broadcasting another RREQ, up to a maximum of retry times at maximum TTL (time-to-live) value.

$Threshold_{Reliable}$: We introduce a quantity called $Threshold_{Reliable}$ which denotes the minimum transfer of useful data packets from any node in the network for that node to be considered reliable. We assume that a node which has stayed in the network for a long time is more reliable than a new comer or a spoofer by virtue of its constant IP address and good history of many data packets transferred. The $Threshold_{Reliable}$ of the network at any instant is not constant but will depend upon the history of the data packet transferred from the majority of the nodes of the network. As an example of how this concept works, let us take an example of an attacking node, which floods target nodes with data packets in an attempt to increase its priority and gain attack benefits. However the destination node will reject those packets and drops the connection immediately and also updating its THL by making the reliability of that node as negative.

4.4 Queue Processing Priorities:

Both RRER and RREP should be given high priority as they carry crucial routing information. Fast Transmission of RREP helps in completing the route discovery process fast. Similarly, a fast transmission of RRER helps in erasing stale routes starting routes discovery faster. So whenever there is a packet in IPQ, it should be immediately processed. During normal operation, the priority of RREQQ should be higher than DPQ since as we are dealing with mobile nodes, routes are expected to break frequently and hence more route discovery operation are needed. However when RREQ flooding attack is detected, the priority of DPQ may be moderately increased. The reason for this is that we want to increase the data transmission speed by depriving the attacker of time slots by itself taking up more bandwidth.

5. FLOOD TOLERANT AODV (FT-AODV) ALGORITHM

The algorithm involves prioritised handling of different packets coming from different sources or nodes. Depending on priorities these packets get preference of memory in the queue and dispatcher schedules. The more the priority of the packet the faster it will be processed. The pseudo code of our algorithm is as follows:

```
A New Packet arrives at a node.

If the Packet is a RRER or RERR Packet
        {
        Send it to Immediate Processing Queue [IPQ] to be
        processed immediately.
        }

Else

If the Packet is RREQ Packet

        {

Evaluate source address of the packet.

Update the Traffic History List [THL].

Calculate the Rate or RREQ received from that node
($Rate_{RREQ}$).

If $RATE_{RREQ}$>= RREQ_RATE_LIMIT

        {

        The system is under flooding attack from that node.

        Update the Traffic History List [THL].
        THL->$D_{Packet}$ as -1.

        Drop that packet.

        }

Else If THL->$D_{Packet}$>= $Threshold_{Reliable}$

        {

        Add the packet to the RREQQ even if RREQQ is
        full by dropping the last packet in RREQQ.

        }

        Else If THL->$D_{Packet}$< $Threshold_{Reliable}$

        {

        If RREQQ is not full

                {
                Add the Packet to RREQQ.
                }

        Else
                {
                Drop the Packet
                }
        }
  }

  If the Packet is a Data Packet

        {

        Evaluate source of the Packet.

        Update the Traffic History List [THL].

        If DPQ is not full

                {
                Add the Packet to DPQ.
                }

        Else If THL->$D_{Packet}$>= $Threshold_{Reliable}$

                {
                Add the packet to the DPQ even if
                RREQQ is full by dropping the last packet
                in DPQ.

                }
        Else

                {
                Drop the Packet
                }
    }
```

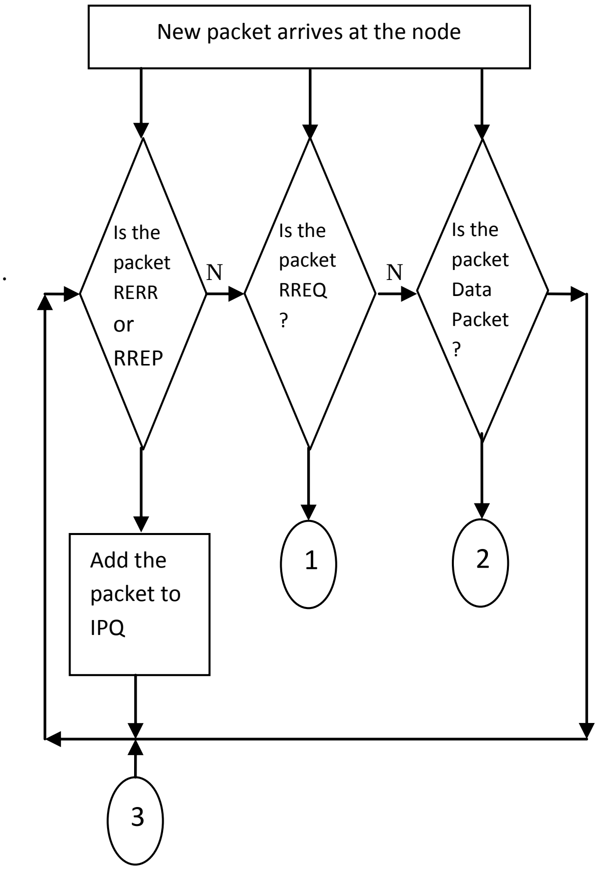
6. FLOW DIAGRAM

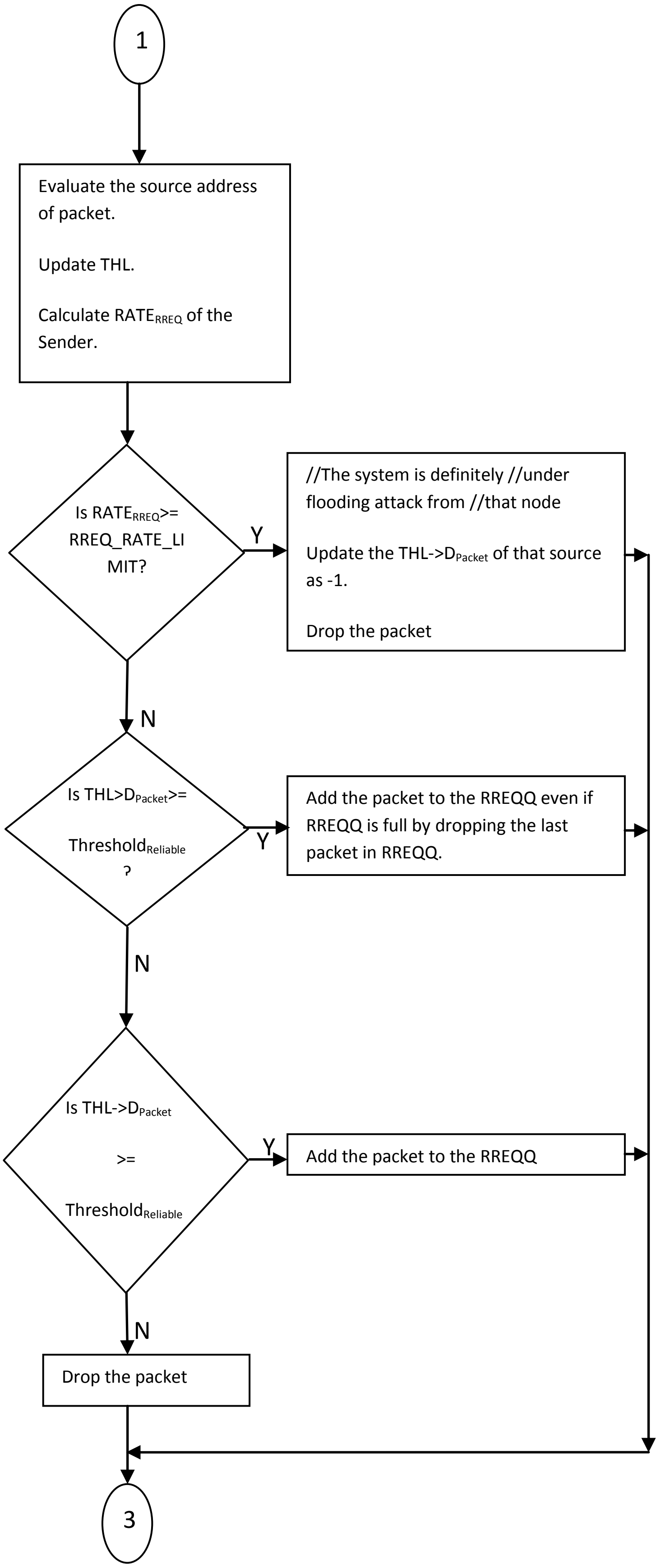
These symbols given below have been used to simplify the complexity of the flow diagram.

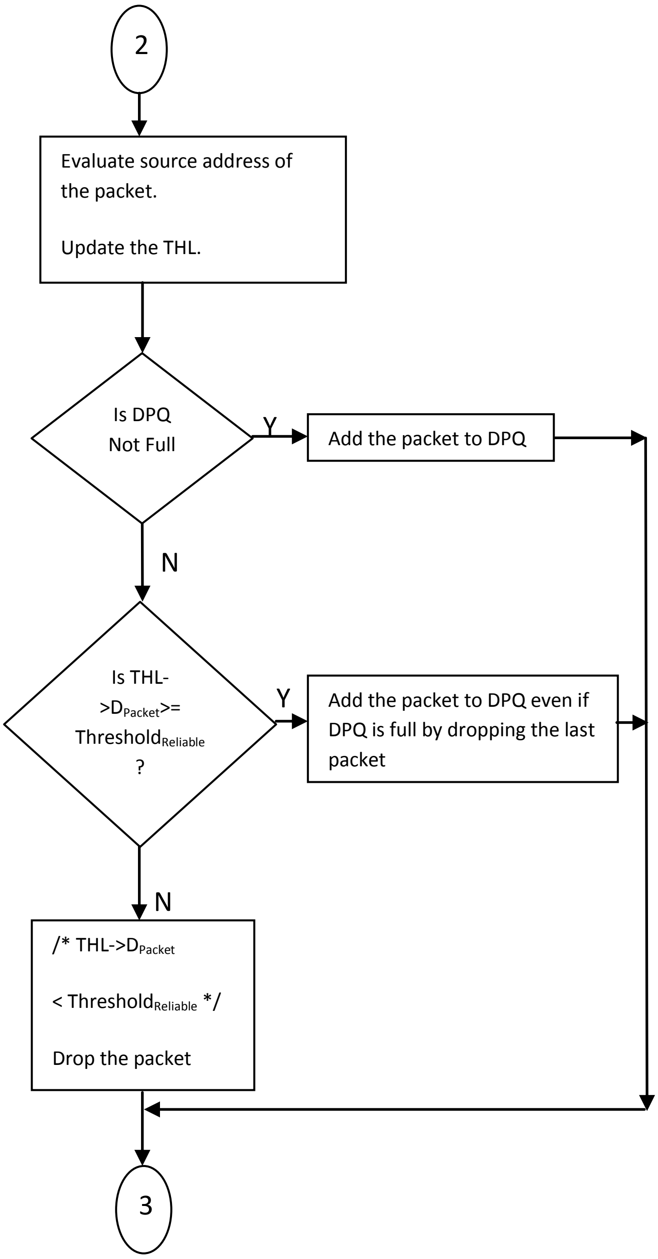
1 : Represents the set of steps involving data that will go to IPQ

2 : Represents the set of steps involving data that will to go to RREQQ

3 : Represents the set of steps involving data that will to go to DPQ


New packet arrives at the node
Is the packet RERR or RREP
N
Is the packet RREQ ?
N
Is the packet Data Packet ?
Add the packet to IPQ
1
2
3

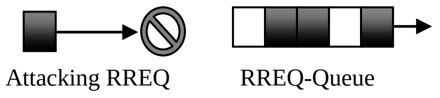

1
Evaluate the source address of packet.
Update THL.
Calculate RATE_RREQ of the Sender.
Is RATE_RREQ>= RREQ_RATE_LIMIT?
Y
//The system is definitely //under flooding attack from //that node
Update the THL->D_Packet of that source as -1.
Drop the packet
N
Is THL>D_Packet>= Threshold_Reliable ?
Y
Add the packet to the RREQQ even if RREQQ is full by dropping the last packet in RREQQ.
N
Is THL->D_Packet >= Threshold_Reliable
Y
Add the packet to the RREQQ
N
Drop the packet
3


2
Evaluate source address of the packet.
Update the THL.
Is DPQ Not Full
Y
Add the packet to DPQ
N
Is THL->D$_{Packet}$>= Threshold$_{Reliable}$?
Y
Add the packet to DPQ even if DPQ is full by dropping the last packet
N
/* THL->D$_{Packet}$ < Threshold$_{Reliable}$ */
Drop the packet
3

7. VERIFICATION OF THE PROPOSED ALGORITHM: FLOOD TOLERANT AODV (FT-AODV)

7.1 Case a

In this case the attackers wants to congest the network, but the nodes which receives the initial RREQ packets spot immediately that the rate of RREQ packets send by the attacking nodes exceeds the RREQ_RATE_LIMIT permitted by the protocol. Since the attacker address does not change, the victim node can subsequently drops all RREQ and Data packets from the attacking nodes until its behaviour rectifies. It can do so because it monitors the activities of all communicating nodes in THL.



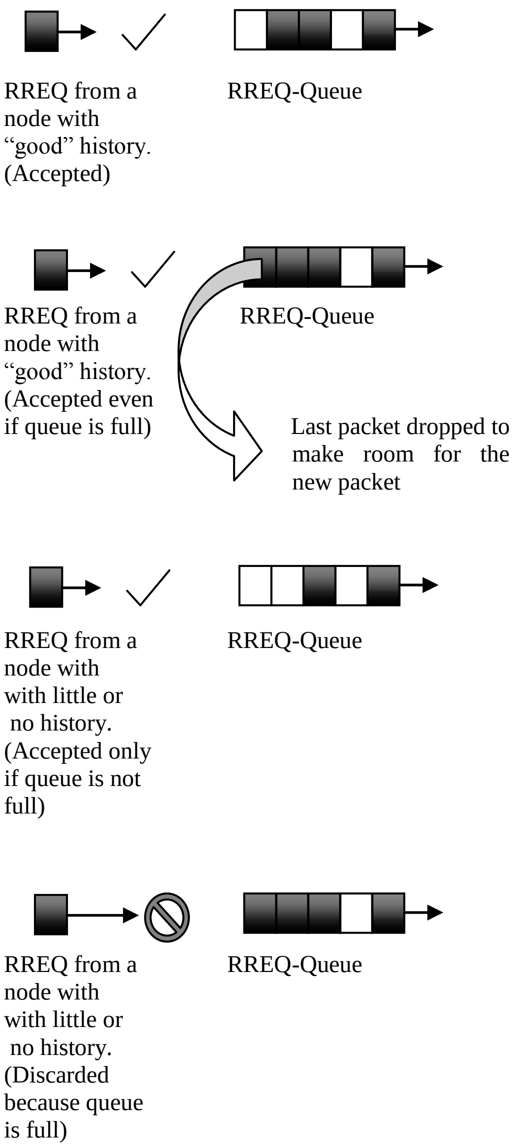

RREQ from a node with “good” history. (Accepted)
RREQ-Queue
RREQ from a node with “good” history. (Accepted even if queue is full)
RREQ-Queue
Last packet dropped to make room for the new packet
RREQ from a node with with little or no history. (Accepted only if queue is not full)
RREQ-Queue
RREQ from a node with with little or no history. (Discarded because queue is full)
RREQ-Queue

Fig. 4 Various scenarios possible in case a.

7.2 Case b

Attackers chooses to flood the network with excessive RREQ packets, while using address spoofing techniques the change the source address of the packet to prevent detection and isolation. The attackers change their IP address frequently while flooding extraneous RREQ packets. As a result , the RREQQ becomes full. After which the receiving node only processes packets with a long and good history (whose source address has not changed for a long time during which it has exhibited non-malicious behaviour).The packets of the dynamic attackers are discarded. However, during the initial period, the packet of a Newly Joined non malicious node in the network also gets discarded. But after a certain period, with consistency of behaviour, its packets also get accepted.

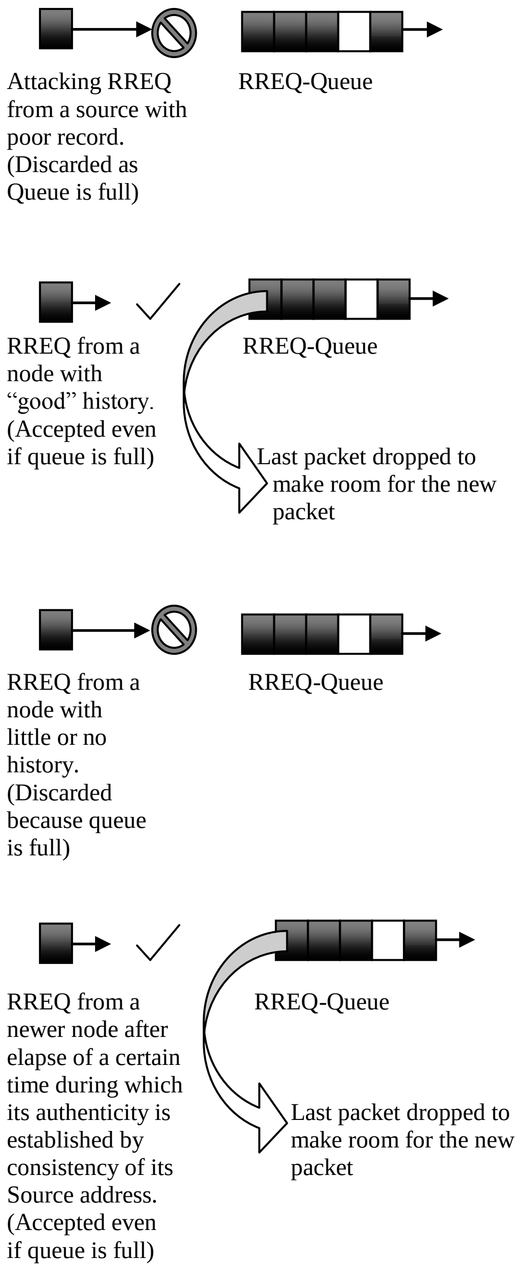

Attacking RREQ from a source with poor record. (Discarded as Queue is full)
RREQ-Queue
RREQ from a node with "good" history. (Accepted even if queue is full)
RREQ-Queue
Last packet dropped to make room for the new packet
RREQ from a node with little or no history. (Discarded because queue is full)
RREQ-Queue
RREQ from a newer node after elapse of a certain time during which its authenticity is established by consistency of its Source address. (Accepted even if queue is full)
RREQ-Queue
Last packet dropped to make room for the new packet

Fig. 5 Various scenarios possible in case b

7.3 Case c

Attacker chooses to flood the network with extraneous data packets, while keeping the same source address. The main complexity of this kind of attack is that the attackers can target the specific nodes to flood them with erroneous data packets and disrupting their service, there is no way to determine in the network layer whether these data packets are useful or not. Hence we suggest an application layer approach for this kind of situations. In the application layer, if any data packets are found to be unwanted, then corrective measures can be taken like dropping the connection and denying connection to that IP for subsequent requests.

7.4 Case d

Attackers chooses to flood the network with excessive data packets, while using address spoofing techniques the change the source address of the packet to prevent detection and isolation.

These attackers attempt to flood the network using data packets but since they also frequently change the source address of the data packets, the packets are discarded by the nodes due to lesser priority of those packets with newer: source addresses.

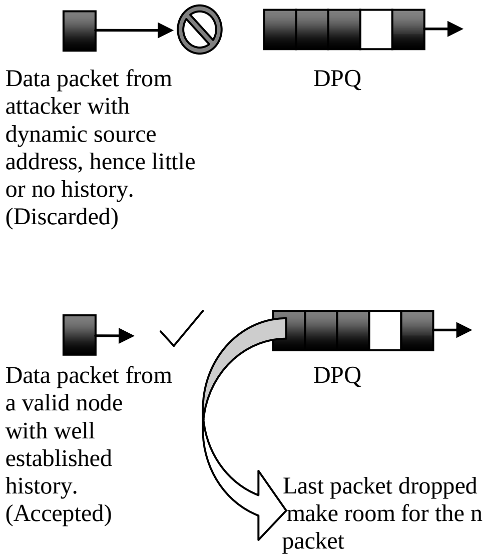

Data packet from attacker with dynamic source address, hence little or no history. (Discarded)
DPQ
Data packet from a valid node with well established history. (Accepted)
DPQ
Last packet dropped to make room for the new packet

Fig. 6 Various scenarios possible in case d.

8. CONCLUSION

We here put forward an elegant and efficient scheme which defends itself from network congestion due to network jamming which in turn are effects of various flooding attacks. In this approach we presented an unconventional and unique approach to the problem of flooding attacks in ad hoc networks. Our unique approach even addresses the difficult issue of flooding attack prevention where the malicious node uses address spoofing techniques to prevent detection and isolation. We have already shown that the network that will use our protocol i.e. FT-AODV will be less prone to congestion due to flooding attack and hence more reliable and efficient. Our approach is simple and requires no infrastructural changes. Our approach works within the realms and various restrictions of ad hoc networks and hence it can be implemented efficiently in MANETs. Thus our scheme provides an elegant solution for protection of ad hoc networks from malicious attackers. This algorithm also works well to increase the speed of the network in all circumstances independent from whether there are no intruders or many intruders in the network. There can be various future practical applications of this flood resistant algorithm in wireless ad hoc communications which secures the network and makes it more efficient and reliable than the AODV based networks.

8. REFERENCES

[1] C.E.Perkins and E.M. Royer, "Ad-hoc On Demand Distance Vector Routing ", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp-90-100, Feb, 1999.

[2] Mobile Ad hoc Networks: Challenges and Future Kavita Taneja and R. B. Patel Proceedings of National Conference on Challenges & Opportunities in Information Technology (COIT-2007) RIMT-IET, Mandi Gobindgarh. March 23, 2007.

[3] Basagni, S., Conti, M., Giordano S., andStojmenovic, I. (Eds.) Ad Hoc Networking.IEEE Press Wiley, New York, 2003.

[4] Chlamtac, I., Conti, M., and Liu, J. J.-N. Mobile ad hoc networking: imperatives and challenges. Ad Hoc Networks, 1(1), 2003, pp. 13–64.

[5] Freebersyser, J. A., and Leiner, B. A DoD perspective on mobile ad hoc networks. In: Perkins, C. (Ed.) Ad Hoc Networking, Addison Wesley, Reading, MA, 2001, pp. 29–51.

[6] IETF MANET Working Group. http://www.ietf.org/html.charters/manetcharter.html

[7] Toh, C-K. Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall, 2002.

[8] Corson, S., and Macker, J. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, IETF, Jan. 1999.

[9] Abolhasan, M., Wysocki, T., and Dutkiewicz, E. A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks, 2(1), 2004, pp. 1–22.

[10] Kozat, U. C., and Tassiulas, L. Service discovery in mobile ad hoc networks: anoverall perspective on architectural choices and network layer support issues. Ad Hoc Networks, 2(1), 2004, pp. 23–44.

[11] B. Awerbuch, D. Holmer, C. Nita Rotaru and Herbert Rubens. "An On-Demand Secure Routing

[12] A Survey of Routing Attacks and Security Measures in Mobile Ad-Hoc Networks Sudhir Agrawal, Sanjeev Jain, Sanjeev Sharma JOURNAL OF COMPUTING, VOLUME 3, ISSUE 1, JANUARY 2011, ISSN 2151-9617 page 41-48.

[13] A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks Bing Wu, Jianmin Chen, Jie Wu and Mihaela Cardei Wireless Network Security Signals and Communication Technology, 2007, Part II, 103-135, DOI: 10.1007/978-0-387-33112-6_5.

[14] Protocol Resilient to Byzantine Failures". Proceedings of the ACM Workshop on Wireless Security 2002, Pages 21-30, September 2002.

[15] Deborah Estrin, Daniel Zappala, Tony Li, Yakov Rekhter, and Kannan Varadhan. Source Demand Routing: Packet format and forwarding specification (version 1). Internet Draft, January 1995. Work in progress.