Detection of Flooding Attacks in MANET

1 Chirag R. Patel & 2 Prof. Tushar S. Patel

1ME (CSE-Student), S.P.B.Patel Engineering Collage, Mehsana, Gujarat, India. Chiragp2571989@gmail.com
2IT Department, S.P.B.Patel Engineering Collage, Mehsana, Gujarat, India. jayambe_it@yahoo.co.in

Abstract

Mobile ad hoc network (MANET) has emerged as a new frontier of technology to provide anywhere, anytime communication. Due to its deployment nature, MANETs are more vulnerable to malicious attack. The absolute security in the mobile ad hoc network is very hard to achieve because of its fundamental characteristics, such as dynamic topology, open medium, limited power and limited bandwidth.

Major attacks on Mobile Ad hoc networks are flooding, selective forwarding, sinkhole, wormhole etc. Malicious node broadcasts too many packets in the networks and creates traffic in the network. So the network jams occur. This attack is also known denial of service attack. We have studied various methods to detect flooding attack. Our proposed mechanism is detecting of flooding attacks in MANET.

II Introduction

Ad hoc networks are temporary networks because they are formed to fulfill a special purpose and cease to exist after fulfilling this purpose. Mobile devices might arbitrarily join or leave the network at any time, thus ad hoc networks have a dynamic infrastructure. Most mobile devices use radio or infrared frequencies for their communications which leads to a very limited transmission range. Usually the transmission range is increased by using multi-hop routing paths. In that case a device sends its packets to its neighbor devices, i.e. devices that are in transmission range. Those neighbor nodes then forward the packets to their neighbors until the packets reach their destination.

The most distinguishing property of ad hoc networks is that the networks are self-organized. All network interactions have to be executable in absence of a trusted third party (TTP), such as
the establishment of a secure channel between nodes and the initialization of newly joining nodes. Hence, in contrast to wireless networks, ad hoc networks do not rely on a fixed infrastructure and the accessibility of a TTP. The self-organizing property is unique to ad hoc networks and makes implementing security protocols a very challenging task. Another characteristics of ad hoc networks are the constrained network devices. The constraints of ad hoc network devices are a small CPU, small memory, small bandwidth, weak physical protection and limited battery power. In most ad hoc networks all devices have similar constraints. This property distinguishes the architecture of an ad hoc network from a client-server structure.

In recent years, the security issues on MANET have become one of the primary concerns. The MANET is more vulnerable to be attacked than wired network. These vulnerabilities are nature of the MANET structure that cannot be removed. Attack prevention measures, such as authentication and encryption, can be used as the first line of defense for reducing the possibilities of attacks. However, these techniques have a limitation on the effects of prevention techniques in general and they are designed for a set of known attacks. They are unlikely to prevent newer attacks that are designed for circumventing the existing security measures. For this reason, there is a need of intrusion detection.

III. Related Study

AODV routing protocol is a reactive [1] routing algorithm. It maintains the established routes as long as they are needed by the sources. AODV uses sequence numbers to ensure the freshness of routes. Route discovery and route maintenance for AODV are described below:

(1) Route Discovery

The route discovery process is initiated whenever a traffic source needs a route to a destination. Route discovery typically involves a network-wide flood of route request (RREQ) packets targeting the destination and waiting for a route reply (RREP). An intermediate node receiving a RREQ packet first sets up a reverse path to the source using the previous hop of the RREQ as the next hop on the reverse path. If a valid route to the destination is available, then the intermediate node generates a RREP, else the RREQ is re-broadcast. Duplicate copies of the RREQ packet received at any node are discarded. When the destination receives a RREQ, it also generates a RREP. The RREP is routed back to the source via the reverse path.
As the RREP proceeds towards the source, a forward path to the destination is established.

(2) Route Maintenance

Route maintenance is done using route error (RERR) packets. When a link failure is detected, a RERR is sent back via separately maintained predecessor links to all sources using that failed link. Routes are erased by the RERR along its way. When a traffic source receives a RERR, it initiates a new route discovery if the route is still needed. Unused routes in the routing table are expired using a timer-based technique.

To understand the working of AODV, we take an example of five mobile nodes as shown in Figure 3.1. The circles indicate the range of communication for the nodes. As each node has a limited communication range, it can communicate with its neighbor nodes only. At an instant, Node 4 wants to communicate with Node 3, but it is uncertain of the route. Node 4 broadcasts RREQ that is received by its neighbors Node 1 and Node 5. Node 5 doesn’t have any route to Node 3 and therefore it rebroadcasts RREQ that is received back by Node 4. Node 4 drops it. On the other side, if Node 1 has a greater sequence number than RREQ, it discards RREQ and replies with RREP. If not, it updates the sequence number in its routing table and forwards RREQ to Node 2. As Node 2 has a route to Node 3, it replies to Node 1 by sending an RREP. Node 1 sends RREP to Node 4 and route Node 4-Node 1-Node 2-Node 3 is confirmed to send data packets. Node 4 can now send data packets to Node 3 through the specified route.

![Fig.1 Communication between nodes in Mobile Ad-hoc Network](image-url)
the destination or the node has a valid route to the destination, it unicasts a route reply (RREP) back to the source node. The source node or the intermediate nodes that receives RREP will update its forward route to destination in the routing tables. Otherwise, it continues broadcasting the RREQ. If a node receives a RREQ message that has already processed, it discards the RREQ and does not forward it. When a link is broken, route error packets (RERR) are propagated to the source node along the reverse route and all intermediate nodes will erase the entry in their routing tables.

The source node or the intermediate nodes that receives RREP will update its forward route to destination in the routing tables. Otherwise, it continues broadcasting the RREQ. If a node receives a RREQ message that has already processed, it discards the RREQ and does not forward it. In AODV, sequence number (SN) plays a role to indicate the freshness of the routing information and guarantee loop-free routes. Sequence number is increased under only two conditions: when the source node initiates RREQ and when the destination node replies with RREP. Sequence number can be updated only by the source or destination. Hop count (HC) is used to determine the shortest path and it is increased by 1 if RREQ or RREP is forwarded each hop. When a link is broken, route error packets (RERR) are propagated to the source node along the reverse route and all intermediate nodes will erase the entry in their routing tables. AODV maintains the connectivity of neighbor nodes by sending hello message periodically.

(3) Flooding Attack Description

Flooding RREQ packets in the whole network will consume a lot of resource of network. To reduce congestion in a network, the AODV protocol adopts some methods. A node cannot originate more than RREQ_RATELIMIT RREQ messages per second. After broadcasting a RREQ, a node waits for a RREP. If a route is not received within round-trip milliseconds, the node may try again to discover a route by broadcasting another RREQ, up to a maximum of retry times at the maximum TTL value. Repeated attempts by a source node at route discovery for a single destination must utilize a binary exponential back off. The first time a source node broadcasts a RREQ, it waits round-trip time for the reception of a RREP. If a RREP is not received within that time, the source node sends a new RREQ. When calculating the time to wait for the RREP after sending the second RREQ, the source node MUST use a binary exponential
back off. Hence, the waiting time for the RREP corresponding to the second RREQ is $2 \times$ round-trip time. The RREQ packets are broadcast in an incrementing ring to reduce the overhead caused by flooding the whole network. The packets are flooded in a small area (a ring) first defined by a starting TTL (time-to-live) in the IP headers. After RING TRAVERSAL TIME, if no RREP has been received, the flooded area is enlarged by increasing the TTL by a fixed value. The procedure is repeated until an RREP is received by the originator of the RREQ, i.e., the route has been found.

![Diagram](image)

**Fig 2.** Demonstration of flooding attack [4]

In the Ad Hoc Flooding Attack, the attack node violates the above rules to exhaust the network resource. Firstly, the attacker selects many IP addresses which are not in the networks if he knows the scope of IP address in the networks. Because no node can answer RREP packets for these RREQ, the reverse route in the route table of node will be conserved for longer. The attacker can select random IP addresses if he can not know scope of IP address. Secondly, the attacker successively originates mass RREQ messages for these void IP addresses. The attacker tries to send excessive RREQ without considering RREQ_RATELIMIT in per second. The attacker will resend the RREQ packets without waiting for the RREP or round-trip time, if he uses out these IP addresses. The TTL of RREQ is set up to a maximum without using expanding ring search method. In the Flooding Attacks, the whole network will be full of RREQ packets which the attacker sends. The communication bandwidth is exhausted by the flooded RREQ packets and the resource of nodes is exhausted at the same time. For example, the storage of route table is limited. If mass RREQ packets are coming to the node in a little time, the storage of route table in the node will exhaust so that the node cannot receive new RREQ packet. As a result, the legitimate nodes cannot set up paths to send data. Figure 3.2 shows that an example of RREQ Flooding Attack. Node H is attacker and it floods mass RREQ packets all over the networks so that the other nodes cannot build paths with each other.
(4) Countermeasures for Flooding Attack

A simple mechanism proposed to prevent the flooding attack in the AODV protocol [2]. In this approach, each node monitors and calculates the rate of its neighbors’ RREQ. If the RREQ rate of any neighbor exceeds the predefined threshold, the node records the ID of this neighbor in a blacklist. Then, the node drops any future RREQs from nodes that are listed in the blacklist. The limitation of this approach is that it cannot prevent against the flooding attack in which the flooding rate is below the threshold. Another drawback of this approach is that if a malicious node impersonates the ID of a legitimate node and broadcasts a large number of RREQs, other nodes might put the ID of this legitimate node on the blacklist by mistake.

In [3], the authors proposed an adaptive technique to mitigate the effect of a flooding attack in the AODV protocol. This technique is based on statistical analysis to detect malicious RREQ floods and avoid the forwarding of such packets. Similar to [2], in this approach, each node monitors the RREQ it receives and maintains a count of RREQs received from each sender during the preset time period. The RREQs from a sender whose RREQ rate is above the threshold will be dropped without forwarding. Unlike the method proposed in [2], where the threshold is set to be fixed, this approach determines the threshold based on a statistical analysis of RREQs. The key advantage of this approach is that it can reduce the impact of the attack for varying flooding rates.

Resisting flooding attacks in ad hoc networks presented in [4] describes two flooding attacks: Route Request (RREQ) and Data flooding attack. The proposed algorithm is composed of neighbor suppression and path cutoff. When the intruder broadcasts exceeding packets of Route Request, the immediate neighbors of the intruder observe a high rate of Route Request and then they lower the corresponding priority according to the rate of incoming queries. Moreover, not serviced low priority queries are eventually discarded. When the intruder sends many attacking DATA packets to the victim node, the node may cut off the path and does not set up a path with the intruder any more. Mobile ad hoc networks can prevent the Ad Hoc Flooding Attack by proposed algorithm with little overhead.

A new trust approach based on the extent of friendship between the nodes is proposed which makes the nodes to co-operate and prevent flooding attacks in an ad hoc
environment in [5]. All the nodes in an ad hoc network are categorized as friends, acquaintances or strangers based on their relationships with their neighboring nodes. A trust estimator is used in each node to evaluate the trust level of its neighboring nodes. The trust level is a function of various parameters like length of the association, ratio of the number of packets forwarded successfully by the neighbor to the total number of packets sent to that neighbor, ratio of number of packets received intact from the neighbor to the total number of received packets from that node, average time taken to respond to a route request etc. Accordingly, the neighbors are categorized into friends (most trusted), acquaintances (trusted) and strangers (not trusted).

To prevent RREQ flooding, the threshold level is set for the maximum number of RREQ packets a node can receive from its neighbors [5]. To prevent DATA flooding, the intermediate node assigns a threshold value for the maximum number of data packets it can receive from its neighbors. If Xrs, Xra, Xrf be the RREQ flooding threshold for a stranger, acquaintance and friend node respectively, Xrf > Xra > Xrs. If Yrs, Yra, Yrf be the DATA flooding threshold for a stranger, acquaintance and friend node respectively then Yrf > Yra > Yrs. If the specified threshold level is reached, further RREQ packets from the initiating node are ignored and dropped. Thus, flooding is prevented in the routing table.

In [6] the author has proposed the defense against RREQ flooding and data flooding attack. To resist RREQ flooding packet, the RREQ rate is checked. Here, the author maintains two threshold values. The RREQ_RATELIMIT is considered as the upper threshold (UT) and RREQ_RATELIMIT /2 is taken as lower threshold (LT). If RREQ rate is less than LT, the node which forwards the RREQ is identified as the normal node. If the RREQ rate lies between LT and UT, the forwarding node is identified as the suspicious node. The RREQ is then delayed in a queue. If RREQ rate is above UT, the forwarding node is identified as the attacker and the RREQ are dropped. The attacking node ID is broadcast to all nodes in the network. Hence, the attacking node is isolated from the network.

To resist the data flooding [6], the author has proposed a new defense mechanism that maintains the flow information monitoring table (FIMT). It contains flow id, source id, packet sending rate and destination id. Sending rates are estimated for each flow in the intermediate nodes. The updated flow information is sent to
the destination along with each flow. The destination node sends the control message to notify the sender nodes about the congestion. The sender nodes, upon seeing these packets, will then reduce their sending rate. If the channel continues to be congested because some sender nodes do not reduce their sending rate, it can be found by the destination using the updated flow details. It checks the previous sending rate of a flow with its current sending rate. When both rates are same, the corresponding sender of the flow is identified as an attacker. Once the DDoS attackers are identified, all the packets from those nodes will be discarded. The attacker is blocked from the communication. Hence network resources are made available to the legitimate nodes in the network.

The algorithm proposed in [7] creates an IDS node in which we AODV is set as a routing protocol. Then after the creation, the IDS node check the network configuration and capture lode by finding that if any node is in its radio range, then capture all the information of nodes. Else nodes are out of range or destination unreachable. With the help of this information IDS node creates a normal profile which contains information like type of packet, in our case (protocol is AODV, pkt type TCP, UDP, CBR), time of packet send and receive and threshold. After creating normal profile, it compare normal profile with each new trace value i.e. check packet type, count unknown packet type, arrival time of packet, sender of packet, receiver of packet. And after detection of any anomaly in that parameters then block that packet sender node (attacker node).

The proposed technique in [8] uses a filter to detect misbehaving nodes and reduces their impact on network performance. The aim of the filter is to limit the rate of RREQ packets. Each node maintains two threshold values. The threshold values are the criterion for each node’s decision of how to react to a RREQ message. The $RATE\_LIMIT$ parameter denotes the number of RREQs that can be accepted and processed as normal per unit time by a node. Each node monitors the route requests it receives and maintains a count of RREQs received for each RREQ originator during a preset time period. Whenever a RREQ packet is received, a check is performed. If the rate of this RREQ originator is below the $RATE\_LIMIT$, the RREQ packet is processed as normal. The $BLACKLIST\_LIMIT$ parameter is used to specify a value that aids in determining whether a node is acting malicious or not. If the number of RREQs originated by a node per unit time
exceeds the value of \texttt{BLACKLIST\_LIMIT}, one can safely assume that the corresponding node is trying to flood the network with possibly fake RREQs. On identifying a sender node as malicious, it will be blacklisted. This will prevent further flooding of the fake RREQs in the network. The blacklisted node is ignored for a period of time given by \texttt{BLACKLIST\_TIMEOUT} after which it is unblocked. The proposed scheme has the ability to block a node till \texttt{BLACKLIST\_TIMEOUT} period on an incremental basis. By blacklisting a malicious node, all neighbors of the malicious node restrict the RREQ flooding. Also the malicious node is isolated due to this distributed defense and so cannot hog its neighbor’s resources. The neighboring nodes of the malicious node are therefore free to entertain the RREQs from other genuine nodes. In this way genuine nodes are saved from experiencing the DoS attack. If the rate of RREQs originated by a node is in between the \texttt{RATE\_LIMIT} and the \texttt{BLACKLIST\_LIMIT}, the RREQ packet is added to a “delay queue” waiting to be processed. Every time a \texttt{DELAY\_TIMEOUT} expires, if there is anything in the delay queue (RREQ packet waiting to be processed), then the first packet is removed to be processed. To do so, malicious node that has a high attack rate will thus be severely delayed. Meanwhile, the proposed rate control mechanism will have no impact on other nodes and also have minimal impact on the normal nodes that send abnormally high RREQs.

The filtering forwarding scheme in [8] slows down the spread of excessive RREQs originated by a node per unit time and successfully prevents DoS attacks. It incurs no extra overhead, as it makes minimal modifications to the existing data structures and functions related to blacklisting a node in the existing version of pure AODV. Also it is more efficient in terms of resource reservations and its computational complexity. In addition to limiting the clogging up of resources in the network, it also isolates the malicious node.

In [9] the author has used the trust estimation function. Because the communication between the nodes in the MANET depends on the cooperation and the trust level on its neighbors so to calculate the trust level the author have used the trust estimation function in the Route discovery phase which will calculate the trust level of each neighboring node. Various parameters which are used for trust estimation are: Total number of RREQ packet sent by the neighbor per unit time, total number of packet successfully
transmitted by the neighbor, Ratio of number of packet received correctly from the neighbor to the total number of received packet.

During the route discovery phase of the DSR Routing protocol, the trust value is also computed for all the neighbors of any node. When any node receives the RREQ from its neighbors then it performs the following steps:

1. It increments the R[i] by one which is a counter maintained by every node for its neighbor which indicates how many RREQ packets it has received from its neighbor.

2. It checks the friendship table to check what type of relationship it is having with this neighbor. It could be friend, acquaintance or stranger.

3. Compares the R[i] with the corresponding threshold values which is a node maximum number of RREQ packets that can be allowed from its neighbor.

* If the neighbor is friend node then it compares whether the R[i] is below the threshold value Xtf then it forwards the packet to next hop otherwise discard the packet and blacklist the node.

* If the neighbor is acquaintance and the R[i] is less than Xta then it forwards the packet otherwise put the node in to the delay queue and allow the node to forward the some packets and analyze its behavior continuously, if still it is misbehaving then declare as a intruder and blacklist the node otherwise treat a normal node.

* If the neighbor is stranger and R[i] is less than Xts then forward otherwise discard the packet and blacklist the node.

### III Propose Method

Security is essential for the widespread of MANET. However, the characteristics of MANET pose both challenges and opportunities in achieving the security goals, such as confidentiality, authentication, integrity, availability, access control, and non-repudiation.

A variety of security mechanisms have been invented to counter malicious attacks. The conventional approaches such as authentication, access control, encryption, and digital signature provide a first line of defense.

As a second line of defense, intrusion detection systems and cooperation enforcement mechanisms implemented in MANET can also help to defend against attacks or enforce cooperation, reducing selfish node behavior. Most of the network resources are wasted in trying to generate routes to destinations that do not exist or routes that are not going to be used for any communication. To detect such type of attack is very crucial in MANET because it
consume lots of network resources. Our RREQ flood attack detection algorithm is as follows:

(1) Method

Step 1:
An IDS (Intrusion Detection System) enable nodes are distributed in the network to monitor the network activity.

Step 2:
The IDS nodes will identify the nodes in its communication range. The IDS node captures all the information of nodes.

Step 3:
When IDS node hears an RREQ message, it increases RREQ count by one. If the RREQ count is more than threshold, RREQ flood attack is detected. The node that sends the faked RREQ will be isolated from the network.

If there are N nodes in the network then the threshold value is N-1.

Step 4:
Periodically, the IDS nodes do the threshold checking in the network. If it found that network load is greater than the maximum limit then there is an attack in the network.

(2) Example

![Fig 3 Simple network structure without attacks](image)

Show in above figure this is a simple structure in the network without attacks. In this structure we put 7 nodes & all nodes are movable. Node 1 sends RREQ to node 2 & node 3. Same as node 2 & 3 send to RREQ to node 5 & 4 then node 5 send RREQ to node 6 then node 6 to node 8 then node 8 to node 7.

Then last make a path to node 1 - node 2 – node 5 – nodes 6 – nodes 8 – nodes 7.

And last node 1 sends data packets to this path to the node 7.

![Fig 4 Simple network structure with attacks](image)
Show in the figure node A is the feck node or attacker node. Node A is creating the attack in the network. Node A sends more RREQ packets in to the network and after some time create the traffic in the network and network become jam its call flooding attack.

Node A sends RREQ packets to node 4, node 7, node 6, node 5 and node 2.

After creating attacks I apply my proposed method and detect the flooding attacks in to the network.

Step 1

Show in figure 5 this a step 1 of my proposed skims. In this step I put two IDS node in to the network. IDS nodes menace Intrusion Detection System node. IDS node has intrusion detection software. IDS node find the attacks in to the network and monitor the network activity.

Step 2

In above figure IDS node make a profile of which nodes are in his range. Show in above figure IDS node 1 make a profile of node 1, node 2, node 3 and node 4. Same as IDS node 2 make a profile of node A, node 5, node 6, node 7, node 8 and also both IDS node communicate to each other.

In profile IDS node collect the information like node id then which transport protocol use like TCP, UDP and other then which routing protocol use like AODV or its and last most important thing threshold value.

Step 3

In above figure check the threshold value to all the node. All node have fix threshold value and it is 3.
Fig 7 Step 3 check the Th=Threshold to all the node

So node 1 to node 8 and node A have 3 threshold values. It menace that all nodes send maximum three RREQ packets to other nodes. If anyone node sends more than three RREQ packets then this node call an attacker node and detect the node.

Show in above fig node 1 send two RREQ so its threshold value is 2 same as node 2, node 3, node 4, node 5, node 6, node 7, node 8 threshold value is one because they send only one RREQ but node A send five RREQ so its threshold value is five and its more than three so node A is the attacker node and detect the node A from the network.

In above figure IDS node check all the node’s threshold value and find out that node A is the feck node or attacker node.

Step 4

Show in figure 8 IDS node also checks the threshold value to all over the network. In network if you have N node then your threshold value is N-1. So in example we use the 9 node so all over the network threshold value is 8. If threshold value is more than 8 then IDS node known that attack is create in the network so find the attacker node and detect this node.

So in figure 8 total threshold value is 2+1+1+1+1+1+1+1+5=14. So IDS node find that node A is the feck node and detect the node A.
Fig 9 detects the attacks & make without attacks network

IV Result and Analysis

For simulation we have used NS-2.34. The simulation parameters are as follow:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulator</td>
<td>NS2 (2.34)</td>
</tr>
<tr>
<td>Routing Protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>250m</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>200s</td>
</tr>
<tr>
<td>No. of Nodes</td>
<td>20</td>
</tr>
<tr>
<td>Scenario Size</td>
<td>500 *500 m²</td>
</tr>
<tr>
<td>Traffic Type</td>
<td>Constant Bit Rate</td>
</tr>
</tbody>
</table>

We have measured packet delivery count, normalized routing load and end to end delay for normal scenario, attacking scenario and after applying proposed method.

(1) Packet delivery count is used to measure the no. of packets delivered to the destination node to that of the packets delivered from the source node.

In normal case the packet delivery ratio is 99.30 percentages, in attacking case it decreases to 48 percentages, after applying proposed scheme it increases to 97.60 percentages.

(2) Routing Load is the ratio of routing packets over received data packets. Normalized routing load is denoted by N and calculated as follows:

\[ N = \frac{C}{R} \]

Where C is the number of routing (control) packets generated and R is the number of data packets received.

In normal case the routing overhead is 2.4, in attacking case it is 41, after applying proposed method it is 2.6.
(3) End-to-End Delay is average time a packet takes for delivery to its destination after it was transmitted. It tells how a protocol adapts or arranges for an immediate delivery of packets to its desired destination.

In normal case the average end to end delay is 0.015; in attacking case it is 0.081 while after applying proposed method it is 0.017. We have measured accurate detection ratio that is 95%.

![Graph showing End to End Delay]

V Conclusion

A variety of security mechanisms have been invented to counter malicious attacks. The conventional approaches such as authentication, access control, encryption, and digital signature provide a first line of defense. Flooding attack is very dangerous for mobile ad hoc network. It jams the network and stops the communication. We have presented many existing methods for detecting flooding attack in mobile ad hoc networks. The proposed mechanism eliminates the need for a centralized trusted authority which is not practical in ad hoc network due to their self-organizing nature. Our proposed method can effectively detect malicious node in mobile ad hoc networks.

An intrusion detection system aims to detect attacks on mobile nodes or intrusions into the networks. However, attackers may try to attack the security system itself. Accordingly, the study of the defense to such attacks should be explored as well.

VI REFERENCES


