International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

Security from flooding Fake Route Request in MANET (Mobile Adhoc Network)

Neha Jain , Piyush Singh
RKDF IST, Bhopal
neha.22j@gmail.com

Abstract: The use of Mobile Ad-hoc Networks (MANETs) has increased in recent times. Reactive routing protocols like AODV [1] used in MANETs, flood the network with route requests whenever a new route is to be discovered. This technique of flooding can be easily altered by malicious nodes to interrupt the network. Generally all nodes have a limit ahead of which requests cannot be sent. Malicious nodes can easily detour this limit and send out large numbers of fictitious route requests in the network, flooding other nodes which ultimately waste all of their processing and battery power in forwarding them. As a result, authentic route requests get ignored and many routes do not get a chance to form and network got congested. In this paper, we propose a method by which we can prevent the network congestion and reduce the bandwidth consumption.

We show by means of reasoning and simulation that our scheme increases the efficiency and throughput of the network.

Keywords: **Ad-hoc network, wireless, Routing Protocol, Flood Control, AODV**

1. Introduction:

In an ad hoc wireless network where wired infrastructures are not feasible, energy and bandwidth conservation are the two key elements presenting

Research challenges. Limited bandwidth makes a network easily overcrowded by control signals of the routing protocol. Routing schemes developed for wired networks hardly ever consider limitations of this type. Instead, they assume that the network is mostly steady and the overhead for routing messages is minor. Considering these differences between wired and wireless network, it is necessary to develop a wireless routing protocol that restricts congestion in the network [2][3][4][5][6][7].

AODV is a reactive routing protocol and its work in two phases

a. Route Discovery
b. Route maintenance

In Route Discovery phase, it discovers the route if the desired route doesn't exist from source to destination. Source initiates the RREQ packet and broadcast it to their neighbors. ach node that receives the RREQ looks in its routing table to see if it is the destination or if it has a new enough route to the destination. If it does, it unicast a route reply (RREP) message back to the source, otherwise re-broadcast RREQ. RREP is sent back along a quash route that was created by RREQ.

In this paper we deal with the problem of flooding fake RREQ in MANET and proposed a method to enhance the efficiency and throughput of network and reduce the bandwidth consumption. We show how flooding fake RREQ make the network congested and degrade its performance. In section II we explain the problem that our solutions target and describe the previous solutions related to this problem. In section III we describe proposed approach. In section IV describe the network simulation and experimental results. In section V conclusion and section VI references are there.

2. Related Work

2.1 Problem Due to Fake RREQ Flooding

AODV protocol work on two phases: Route Discovery and Route Maintenance

In Route Discovery phase whenever a source wants to communicate with the destination it revealed the route if it doesn't exist. A source node broadcast the RREQ packet to their neighbors containing information about the recipient. These packets are forwarded by other nodes until a legitimate route is found or timeout occurs. To control the number of RREQs generated by a node, AODV specifies the RREQ_RATELIMIT, a parameter that defines the maximum number of RREQs a node can generate in one second. However, a malicious node ignores this limit and floods the network with large no. of RREQs which then forwarded to their neighbors. Due to this authentic RREQs will drop.

The above explain phenomenon has various effects, from ineffective routing to complete blocking of route information. The situation worsens if many malicious nodes exist in the network. The route forming process is disrupted severely in the locality of the malicious node(s) and improves slowly as we move further away from the epicenter. This is because the neighboring nodes keep forwarding the fake RREQs till their RREQ_RATELIMIT gets exhausted and drop the rest.

We can summarize the effects of flooding fake RREQs as follows:

• Consumption of memory resources while maintaining routing table entries for routes that will never be used.

• Consumption of battery and processing power while forwarding the fake RREQs.

• Denial of service to authentic nodes when routes are not formed.

• Creation of longer routes where shorter ones could have been possible. Hence reduced throughput due to increased hop count.

The solution we propose provides a simple and efficient way to restrain this flooding Fake RREQ activity and its hazardous effects

2.2 Previous Solutions

In [8], the author proposed the distributive approach to resist the flooding attack. In this method they have used the two threshold value; RATE_LIMIT and BLACKLIST_LIMIT. If RREQ count of any node is less then RATE_LIMIT then the request is processed otherwise check whether it is less then BLACKLIST_LIMIT, if yes then black list the node but if the count is greater than RREQ_LIMIT and less than BLACKLIST_LIMIT then put the RREQ in the delay queue and process after queue time out occurs. This method cans Handel the network with high mobility.

In [9], the author analyzed the flooding attack in anonymous communication. They used the threshold topple which consist of three components: transmission threshold, blacklist threshold and white listing threshold. If any node generates RREQ packet more than transmission threshold then its neighbor throw-outs the packet if it crosses the transmission threshold more than blacklist threshold then it black list the node. But to deal with fortuitous blacklisting they defined white listing threshold. If any node performs good for number of intervals equal to white listing threshold then it again start treating as a normal node.

- In [10], the author proposed a proactive scheme to avoid a specific kind of DoS attack and recognize the unruly node. In this scheme, the number of RREQs that can be accepted from a neighbor is limited 3 RREQ_ACCEPT_LIMIT.Hence, the neighbors of the malicious node, will only accept an forward three RREQs packets received from it within a time interval of one sec. Whenever the malicious node crosses the RREQ_BLACKLIST_LIMIT of 10 RREQ packets, its neighbors will blacklist it and separate the malicious node.

3. Proposed Approach

The proposed flooding attack detection and prevention model is distributed model in which node cooperate with each other to 7detect and prevent flooding attack in the network. In our work we have used AODV routing protocol. AODV routing protocol work sin 2 phases:

a) Route Discovery

b) Route Maintenance

In Route Discovery phase whenever a source wants to communicate with the destination it discovered the route if it doesn't exist. A source node broadcast the RREQ packet to their neighbors containing information about the recipient. These packets are forwarded by other nodes until a valid route is found or timeout occurs.

In previous approach to control the Fake RREQ flooding the RREQ_RATELIMIT parameter which define the maximum number of RREQ a node can generate in one second , this limit is 10 [10]

In proposed approach we use following data structures:

RREQ_RATELIMIT: It is the number of RREQs that a node can transmit to their neighbor for route discovery.

RREQ_ACCEPT_LIMIT (RAL): It is the Number of RREQs that a node can accept and Process from each of its neighbors per unit Time. Its purpose is to try and ensure fairness By accepting some RREQs from all neighbors Rather than many from just one.

RREQ_BLACKLIST_LIMIT (RBL): It is the threshold value that determines if a particular neighbor is malicious or not. If the no. of RREQs sent by a neighbor per unit time exceeds this value, the neighbor is assumed to be acting malicious and is blocked by the node.

DROPTAIL: It contains all the fake RREQs. When the TTL of any RREQ reached 0, the packet will discard from the queue.

In proposed approach this RREQ _RATE LIMIT is set to infinite that means a node can send as many as packets but the neighbor of that node can accept only 3 packets in one second. Whenever the malicious node crosses the BLACKLIST_LIMIT of 10 RREQ packets and the node is still receiving the RREQ packet after crosses the BLACKLIST_LIMIT the packets add to the DROPTAIL_QUEUE, packets will remain in the DROPTAIL_QUEUE until the TTL of the packet will 0. In this approach the neighbor node will act as a black hole for the sending node.

As the neighbor node of the sender node is accepting all the RREQ packets and packet is not in the network so the congestion on network will reduce and bandwidth consumption will increase.

4. Simulation/Experiments and Analysis

NS-2 simulator is used [11] [12] for the implementation of the proposed scheme. The AODV protocol is used as the base protocol. Modifications were made to this version of AODV protocol that confirms to RFC 3561. TCP was used as the transport protocol Radio transmission range is set as 250 meters. Traffic sources used are Constant-Bit-Rate (CBR) and the field configuration is 2000 x 2000m with 49 nodes.

We consider the current node i.e. NODE 0 as a malicious node .It starts flooding the network and traffic generated from source to destination pairs are randomly spread over the entire network.

```
Simulation length in seconds: 6.998959525
Number of nodes: 49
Number of sending nodes: 49
Number of receiving nodes: 49
Number of generated packets: 8168
Number of sent packets: 7916
Number of forwarded packets: 285
Number of dropped packets: 1658
Number of lost packets: 531
Minimal packet size: 28
Maximal packet size: 1092
Average packet size: 103.5834
Number of sent bytes: 922528
Number of forwarded bytes: 113400
Number of dropped bytes: 234384
Packets dropping nodes: 0  1  2  3  4  5  6  7  8  9
10 11 12 13 14 15 16 17 18 19 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 37 38 39 40 41
43 44 45 46 47
```

Fig:4.1Simulation Information (Proposed Approach)

```
Simulation length in seconds: 6.997210577
Number of nodes: 49
Number of sending nodes: 49
Number of receiving nodes: 49
Number of generated packets: 7867
Number of sent packets: 7555
Number of forwarded packets: 211
Number of dropped packets: 1664
Number of lost packets: 508
Minimal packet size: 28
Maximal packet size: 1092
Average packet size: 103.7731
Number of sent bytes: 862480
Number of forwarded bytes: 69440
Number of dropped bytes: 236334
Packets dropping nodes: 0  1  2  3  4  5  6  7  8  9
10 11 12 13 14 15 16 17 18 19 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 37 38 39 40 41
43 44 45 46 47
```

Fig:4.2Simulation Information (Previous Approach)

```
Number of generated packets: 1909
Number of sent packets: 1909
Number of forwarded packets: 0
Number of received packets: 1480
Number of dropped packets: 198
Number of lost packets: 0
Number of sent bytes: 519100
Number of forwarded bytes: 0
Number of received bytes: 60532
Number of dropped bytes: 100646
Minimal packet size: 28
Maximal packet size: 1092
Average packet size: 171.0333
```

Fig 4.3 Current node information (Proposed Approach)

```
Number of generated packets: 1946
Number of sent packets: 1946
Number of forwarded packets: 0
Number of received packets: 1469
Number of dropped packets: 244
Number of lost packets: 0
Number of sent bytes: 549020
Number of forwarded bytes: 0
Number of received bytes: 59906
Number of dropped bytes: 121752
Minimal packet size: 28
Maximal packet size: 1092
```

Average packet size: 178.3092

Fig 4.4 Current node information (Previous Approach)

In this phenomenon we can see that proposed approach is better than the previous approach.

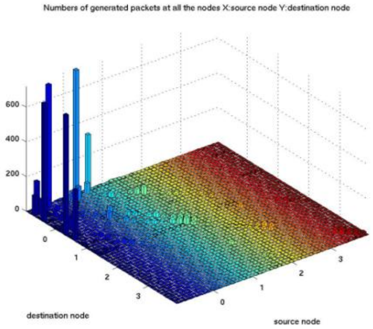

Fig 4.5 No. of generated packets at all the nodes X: Source node Y: destination node (Previous approach)

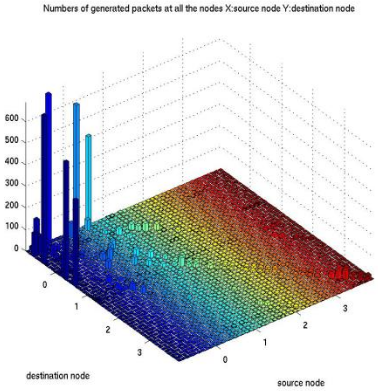

Fig 4.6 No. of generated packets at all the nodes X: Source node Y: destination node (Proposed approach)

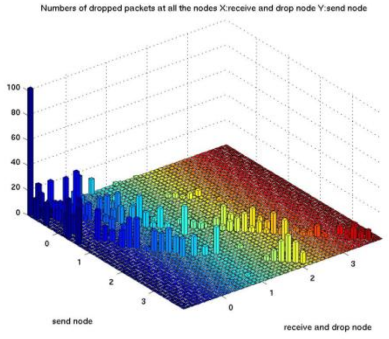

Fig 4.7 No. of dropped packets at all nodes X: source node Y: Destination node (Proposed Approach)

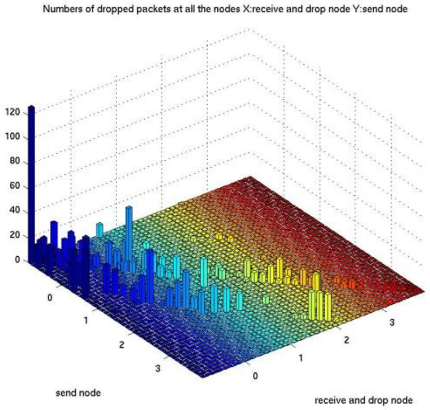

Fig 4.8 No. of dropped packets at all nodes X: source node Y: Destination node (Previous Approach)

In fig 4.5, fig 4.6, fig 4.7 and fig 4.8 shows the no. of generated packets and dropped packets at all nodes X: source node Y: Destination node of proposed and previous approach is compared. We can see that the number of routes formed in proposed approach is consistently higher than that in previous approach and thus proving the scalability of our proposed solution.

5. Conclusion

We have shown that controlling the flood of route requests in the network using a distributed approach helps in improving the overall performance of the network. The RREQ flow control achieved by proposed approach is much better and flexible than the control achieved by previous approach. Our distributed approach does not rely on malicious node information diffusion and joint decision making, which makes the scheme well suited for Ad-hoc networks.

6. References

[1] Perkins, E. Royer, "Ad-hoc On-Demand Distance Vector Routing", *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, 90 (1999).

[2] Perkins C.E., Royer E.M., Das S.R., Ad hoc On-Demand Distance Vector (AODV) Routing .draft-ietf-manet-aodv-08.txt, March 2001.

[3] Broch J., Maltz D.A., Johnson D.B., Hu Y.C., and Jetcheva J., A performance comparison of multi-hop wireless ad hoc network routing protocols. In 4th International Conference on Mobile Computing and Networking (ACM MOBICOM' 98), pages 85–97, Oct 1998.

[4] Rahman A., Security issues in mobile systems, http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs /sec-in-ob.html, 1995 (accessed on May 03, 2004).

[5] Perkins C.E. and Royer E., Ad-hoc on-demand distance vector routing. In 2nd IEEE Workshop on Mobile Computing Systems and Applications, pages 90–100, 1999.

[6] Perkins C.E., Das S.R. and Royer E.. Ad-hoc on-demand distance vector (aodv) routing. http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-05.txt, 2000 (accessed on May 03, 2004).

[7] Karpijoki V., Signaling and routing security in mobile and ad-hoc networks.http://www.hut.fi/vkarpijo/iwork00/, 2000 (accessed on May 03,2004)

[8] Jian-Hua Song1, 2, Fan Hong1, Yu Zhang1 "Effective Filtering Scheme against RREQ Flooding Attack in Mobile Ad Hoc Networks " Proceedings of the Seventh International Conference on Parallel and Distributed Computing,Applications and Technologies (PDCAT'06)0- 7695-2736-1/06 $20.00 © 2006

[9] Venkat Balakrishnan, Vijay Varadharajan,and Uday Tupakula" Mitigating Flooding attacks in Mobile Ad-hoc Networks Supporting Anonymous Communications" The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007) 0- 7695-2842-2/07 $25.00 © 2007

[10] Sugata Sanyal[1], Ajith Abraham[2], Dhaval Gada[3], Rajat Gogri[3], Punit Rathod[3], Zalak Dedhia[3] and Nirali Mody[3], Security Scheme for Distributed DoS in Mobile Ad Hoc Networks.

[11] Fall K. and Varadhan K. (Eds.), NS notes and documentation. http://www.mash.cs.berkely.edu/ns/, 1999 (accessed on May 03, 2004).
[12] UCB/LBNL/VINT, Network Simulator-NS,

[13] Candidate Neighbours to Rebroadcast the RREQ for efficient flooding in mobile ad hoc network Hamad, S.; Radhi, N.; Al-Raweshidy, H.
Wireless Advanced (WiAd), 2011 Digital Object Identifier: 10.1109/WiAd.2011.5983280 Publication Year: 2011 , Page(s): 26 – 29

[14] Malicious AODV: Implementation and Analysis of Routing Attacks in MANETs Ehsan, H.; Khan, F.A., Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on Digital Object Identifier: 10.1109/TrustCom.2012.199
Publication Year: 2012 , Page(s): 1181 - 1187