Resisting Flooding Attacks on AODV

Mohamed A. Abdelshafy and Peter J. B. King
School of Mathematical & Computer Sciences
Heriot-Watt University, Edinburgh, UK
{ma814, P.J.B.King}@hw.ac.uk

***Abstract*—AODV is a reactive MANET routing protocol that is vulnerable to a dramatic collapse of throughput when malicious intruders flood the network with bogus route requests. We introduce a simple mechanism to resist such attacks that can be incorporated into any reactive routing protocol. It does not require expensive cryptography or authentication mechanisms, but relies on locally applied timers and thresholds to classify nodes as malicious. No modifications to the packet formats are needed, so the overhead is a small amount of calculation at nodes, and no extra communication. Using NS2 simulation, we compare the performance of networks using AODV under flooding attacks with and without our mechanism, showing that it significantly reduces the effect of a flooding attack.**

Keywords–MANET, Routing, AODV, Security, Attack, Flooding

I. Introduction

A Mobile Ad Hoc Network (MANET) is a decentralized infrastructureless network in which nodes cooperate to forward data from a source to a destination. Each node in a MANET acts both as a router and as a host. Several routing protocols have been designed for MANETs [1] to optimize network routing performance. The major issues involved in designing a routing protocol for MANET are nodes mobility, bandwidth constrained and error prone wireless channel, resource constrained nodes, and dynamic changing of the network topology [2].

MANET routing protocols can be classified as proactive or reactive routing protocols. In proactive (table-driven) routing protocols, each node maintains one or more tables containing routing information to every other node in the network. While in reactive (on-demand) routing protocols, routes are created whenever a source requires to send data to a destination node which means that these protocols are initiated by a source on-demand. In this paper, we focus on the AODV protocol [3] which is one of the extensively studied reactive protocols, considered by the IETF for standardization.

AODV [3] is a reactive routing protocol. It uses destination sequence numbers to ensure the freshness of routes and guarantee loop freedom. To find a path to a destination, a node broadcasts a route request (RREQ) packet to its neighbors using a new sequence number. Each node that receives the broadcast sets up a reverse route towards the originator of the RREQ unless it has a fresher one. When the intended destination or an intermediate node that has a fresh route to the destination receives the RREQ, it unicasts a reply by sending a route reply (RREP) packet along the reverse path established at intermediate nodes during the route discovery process. Then the source node starts sending data packets to the destination node through the neighboring node that first responded with an RREP. When an intermediate node along the route moves, its upstream neighbor will notice route breakage due to the movement and propagate a route error (RERR) packet to each of its active upstream neighbors. Routing information is stored only in the source node, the destination node, and the intermediate nodes along the active route which deal with data transmission. This scenario decreases the memory overhead, minimizes the use of network resources, and runs well in high mobility situation.

MANET inherits security threats that are faced in wired as well as wireless networks and also introduces security attacks unique to itself [2] due its characteristics. The limitations associated with battery powered MANET nodes mean that computationally expensive cryptographic techniques such as public key algorithms are undesirable.

MANET routing protocols are designed based on the assumption that all nodes cooperate without maliciously disrupting the operation of the protocol. However, the existence of malicious nodes cannot be disregarded in any system, especially in MANETs because of the wireless nature of the network. A malicious node can attack the network layer in MANET either by not forwarding packets or by changing some parameters of routing messages such as sequence number and IP addresses, sending fake messages several times and sending fake routing information to cause congestion and so disrupt routing operations. Node mobility introduces also the difficulty of distinguishing between stale routes and fake routes. Attacks on MANETs come in a number of classes [4] and a number of defences to these attacks have been proposed and evaluated by simulation [4]–[6]. Attacks against MANET are classified based on modification, impersonation or fabrication of the routing messages. While there are large number of existing attacks, our paper is focused on flooding attack which has a dramatic impact on AODV [2] [4].

In AODV under flooding attack [7], a malicious node floods the network with a large number of RREQs to non-existent destinations in the network. Since the destination does not exist in the network, a RREP packet cannot be generated by any node in the network. When a large number of fake RREQ packets are being injected into the network by malicious nodes, significant proportions of the network capacity are consumed by the RREQ packets, depleting the bandwidth available for data. In addition, routing tables accumulate reverse routes to the source of the fake packets, often leading to table overflow and the inability to record new valid routes. This is a type of denial of service attack.

Security mechanisms are added to existing routing protocols to resist attacks. Cryptographic techniques are used to ensure the authenticity and integrity of routing messages [8]. A major concern is the trade off between security and performance, given the limited resources available at many MANET nodes. Both symmetric and asymmetric cryptography have been used as well as hash chaining. Examples of these security enhanced protocols are Authenticated Routing for Ad-hoc Networks (ARAN) [9], Secure Link State Routing Protocol (SLSP) [10], and Secure Ad-hoc On-demand Distance Vector routing (SAODV) [11]. In addition to the power and computation cost of using cryptographic techniques, the performance of secured mechanism such as SAODV is worse than AODV [4] in the presence of flooding attack because of the malicious nodes impersonating non-existent nodes which cannot be discovered by other non-malicious nodes. Thus, securing the routing messages cannot guarantee the detection of the flooding malicious nodes.

We introduce a new Anti-Flooding mechanism that can be used for all on-demand routing protocols. Each node in this mechanism is responsible for monitoring the behaviour of its neighbors to detect malicious nodes and exclude them. We integrate our proposed mechanism into AODV and SAODV as examples of on-demand routing protocols. This paper demonstrates a significant improvement in performance when using our mechanism. The results reported here related to AODV, but we have also measured SAODV with this mechanism and the improvement in performance is significantly higher than AODV.

The rest of the paper is organized as follows. Section II presents the related work. In Section III, our proposed mechanism to detect the flooding attack is introduced. In Section IV, the simulation approach and parameters is presented. In Section V, simulation results are given. In Section VI, conclusions are drawn.

II. Related Work

Although significant algorithms have been introduced to secure MANET, most of these algorithms cannot resist a flooding attack. A malicious node initiating a flooding attack generates a large number of RREQs to non-existant nodes. These RREQ flood out through the MANET and because the destination does not exist, are propagated by all nodes. A node has no way of detecting whether the neighbor that sent the RREQ is malicious or not. All suggested solutions to the flooding attack attempt to classify neighbors as normal or malicious nodes and then suppress malicious ones.

Flooding Attack Prevention (FAP) [12] is the first solution to resist against flooding attack. The algorithm defined a neighbor suppression method which prioritizes the node based on the number of RREQ received. A node gets higher priority if it sends fewer RREQ packets. When a malicious node broadcasts large number of RREQ packets, the immediate neighbors of the malicious node observe a high rate of RREQ and then they lower the corresponding priority according to the rate of incoming queries. Forwarding received RREQ depends on the priority value of the sending neighbor. The disadvantage of this algorithm is that it still disseminates flooding packets albeit at a reduced rate.

Threshold prevention [13] is introduced to modify FAP by defining a fixed RREQ threshold. The algorithm assumes that if the number of RREQ packets received from a neighbor exceeds the threshold value, this neighbor is a malicious node and discards all future packets from this malicious node. The algorithm becomes useless if a malicious node knows the threshold value then it can bypass the mechanism. Another disadvantage of this algorithm is that it treats a high mobility normal node as if it is a malicious node.

A distributed approach to resist the flooding attack is introduced in [14]. The algorithm defines two threshold values; RATE_LIMIT and BLACKLIST_LIMIT. A RREQ from a neighbor is processed only if the number of previously received RREQ from this neighbor is less than RATE_LIMIT. On the other hand, if the number of previously received RREQ from this neighbor is greater than BLACKLIST_LIMIT, the RREQ is discarded and this neighbor is blacklisted. If the number of previously received RREQ from this neighbor is greater than RREQ_LIMIT and less than BLACKLIST_LIMIT, the RREQ is queued for processing after a delay expires. A disadvantage of this approach is the ability of the attacker to subvert the algorithm by disseminating thresholds levels and the possibility of permanently suspending a blacklisted neighbor that is not malicious.

The algorithm introduced in [15] tried to find a solution to the flooding attack from the communication point of view. The algorithm defines three threshold values; transmission threshold, blacklist threshold and white listing threshold. A RREQ from a neighbor is processed only if received RREQ rate from this neighbor is less than the transmission threshold; otherwise the node will discards the RREQ. If the received RREQ rate from this neighbor is greater than the blacklist threshold, the RREQ is discarded and this neighbor is blacklisted. This algorithm avoids permanently suspending of a blacklisted neighbor by introducing a white listing threshold. A blacklisted neighbor can be returned to normal status if it behaves correctly for a whitelisting time interval.

The algorithm introduced in [16] extends DSR protocol based on the trust function to mitigate the effects of flooding attack. This algorithm classifies a node neighbors based on a trust value to three categories; friend, acquaintance and stranger. Friend is a trusted node and stranger is a non-trusted node while an acquaintance has the trust value that is greater than a stranger and less than a friend. The algorithm defines a threshold value to each neighbor type. A node decision will be taken based on the neighbor type that sends the RREQ and threshold value of this neighbor type. As a general rule, if a node receives a RREQ from a neighbor, it first checks its relationship class and based on this it checks if this neighbor runs over the relationship class threshold value or not. The node processes the RREQ if this neighbor still running under the relationship class threshold otherwise it discards the RREQ and blacklists this neighbor. The disadvantage of this algorithm is that it cannot support high node mobility. [17] introduces a modification to this algorithm to extend the algorithm for high node mobility. A significant disadvantage of this approach is that it depends on a modification of DSR and cannot be adapted to other MANET protocols.

III. AF-AODV Protocol

AF-AODV is designed to mitigate the effect of the flooding attack on the performance of AODV protocol. The mechanism does not use cryptographic techniques which conserves the power and computation resources. Each node in the network has to monitor the performance of its neighbors to detect if they are trying to flood the network or not. Malicious nodes will be detected reliably within a very few minutes. The only way for a malicious node to subvert the mechanism is to transmit fake RREQ packets at such a low rate that they do not impact the network performance significantly.

The idea is to record for each neighbor the rate at which it transmits RREQs. A node pursuing a flooding attack will be generating a high number of RREQs. If the rate exceeds a threshold, then the neighbor is added to a black list of potential malicious nodes. Once on the black list, RREQs from the black listed node are not forwarded, but they are still recorded. A node can be removed if its rate of RREQ generation reduces below the threshold. If the rate continues high, the offending node is queried - only a non-malicious node will respond. After two queries, the neighbor will be suspended for a period, and if its rate is still high after the period has elapsed it will be declared as malicious. A node implementing the Anti-Flood mechanism behaves as follows:

- Every TRAFFIC_TIME, the number of RREQs received from each neighbor since the last classification update is examined.
- If the number of RREQs received from a neighbor exceeds the threshold RREQ_THRESHOLD, that neighbour has its black_list value set to 1. If multiple neighbours exceed the threshold, the neighbor which has transmitted the largest number of RREQs has its black_list value set to 1. Other neighbors that exceeded the threshold are suspended. RREQs from suspended nodes are ignored and not forwarded. Suspension of neighbors except the one with the largest RREQ count allows the mechanism to avoid double counting of RREQs and concentrate on classification of the worst offender. Choice of the RREQ_THRESHOLD is made by running AODV on a large number of scenarios and observing the largest number of RREQs that can be received in TRAFFIC_TIME.
- RREQ packets are processed normally when received from neighbors with a black_list value of 0. If a RREQ is received from a neighbor with a black_list value of 1, then the node examines how many RREQs have been received in an interval of RREQ_TIME_1. If that is less than RREQ_COUNT_1, the black_list value for that neighbor is reset to 0. If the number exceeds RREQ_COUNT_1, the node tests the authenticity of the neighbor by constructing a fake RREP packet to the RREQ and replying with that RREP. If the neighbor is malicious, this will not result in any data flowing. If it is not malicious, data will flow to the fake RREP originator, which can respond with a RERR so that a new route can be found. If no data flows within RREP_WAIT_TIME, the neighbor's black_list value is set to 2.
- If a RREQ is received from a neighbor with a black_list value of 2, it re-examines the rate of RREQ received from that node. If the number of RREQ received from this neighbor is less than RREQ_COUNT_1 in a duration less than or equals RREQ_TIME_1, it decrements the black_list value to 1. Otherwise the node again sends a fake RREP to the RREQ sender to test its authenticity. If the RREP_WAIT_TIME expires without receiving the data, the node assigns 3 to black_list value of this neighbor and suspends this neighbor for a long period equals to the next TRAFFIC_TIME + EXCLUDE_TIME. This long suspension ensures that if the behaviour of this neighbor has been affected by a malicious node, then that malicious node will have been identified and isolated during this suspension.
- After the long-time suspension has expired, the node restarts the previous process; it counts again the number of received RREQ from this neighbor and if the number is less than the threshold RREQ_THRESHOLD, it decrements the black_list value to 2. Otherwise it will increment the black_list value to 4.
- If a RREQ is received from a neighbor with a black_list value equals 4, it monitors the rate of RREQ received from this neighbor. If the number of RREQ received from this neighbor is less than RREQ_COUNT_1 in a duration less than or equals RREQ_TIME_1, it decrements the black_list value to 3. Otherwise the node sends a fake RREP to the RREQ sender to test its authenticity for the final time. If the RREP_WAIT_TIME expires without receiving the data, the node assigns 5 to black_list value of this neighbor meaning that this neighbor is a malicious node and deletes this neighbor from neighbor list. All received RREQ from a neighbor that has black_list value equals 5 will be dropped without processing as a result of detecting a malicious node.

TABLE I. AF-AODV PARAMETERS

RREQ_THRESHOLD	10
RREQ_COUNT_1	7
RREQ_COUNT_2	3
RREQ_TIME_1	5
RREQ_TIME_2	2
RREP_WAIT_TIME	1 s
TRAFFIC_TIME	10 s
EXCLUDE_TIME	60 s

Table 1 shows the values of parameters that were used in our simulations.

IV. Simulation Approach

NS-2 simulator [18] is used to simulate flooding attack. The simulation is used to analyse the performance of AODV and our new AF-AODV routing protocols under these attacks. The parameters used are shown in Table 2. Node mobility was modelled with the random waypoint method. Our simulation results are obtained from 3 different movement scenarios, 3 different traffic scenarios and 3 different node-type (malicious or non-malicious) scenarios which means that each metric value is the mean of the 27 runs. The node-type scenario is created randomly. In all cases, the 90% confidence interval was small compared with the values being reported. In this paper, we focused on their impact of the flooding attack on the TCP traffic only. We examined our proposed mechanism for different number of nodes (25, 50, 75 and 100) and different node speeds (0, 10, 20 and 30 m/s). Node mobility had no significant effect of performance in the presence of malicious nodes, so we report here only the case of static networks. Similarly, only the case of 100 node networks is reported, corresponding to a high density of nodes. This gives malicious nodes a high number of neighbors. We choose a large simulation time to be sure that all malicious nodes have

TABLE II. SIMULATION PARAMETERS

Simulation Time	600 s
Simulation Area	500 m x 500 m
Number of Nodes	25, 50, 75, 100
Number of Malicious Nodes	0 - 10
Node Speed	0, 10, 20, 30 m/s
Pause Time	10 s
Traffic Type	TCP
Flooding Rate	2 Packets/s

been detected specially for scenarios with a large number of malicious nodes.

Packet Delivery Ratio (PDR): The ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent out by the sender.

Throughput: The number of data bits delivered to the application layer of destination node in unit time measured in bps.

End-to-End Delay (EED): The average time taken for a packet to be transmitted across the network from source to destination.

Routing Overhead: The size of routing packets measured in Kbytes for route discovery and route maintenance needed to deliver the data packets from sources to destinations.

Normalized Routing Load (NRL): The total number of routing packets transmitted divided by the number of received data packets.

Route Discovery Latency (RDL): The average delay between the sending RREQ from a source and receiving the first corresponding RREP.

Sent Data Packets: The total number of packets sent by all source nodes during the simulation time.

V. SIMULATION RESULTS

The effect of flooding attack on the packet delivery ratio is shown in Figure 1. While the flooding attack has severe impact on the PDR of AODV specially for large number of malicious nodes, AF-AODV has not significantly change for low number of malicious nodes and has negligible decreasing for high number of malicious nodes. AF-AODV enhances PDR over AODV by approximately 5%.

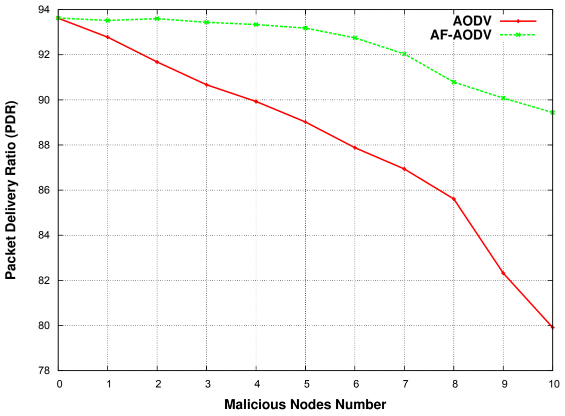

Figure 1. Packet Delivery Ratio

The enhancement of PDR becomes more remarkable if we integrate it to the number of packets that can be sent which is shown in Figure 2. The figure shows that the total number of packets that can be sent is dramatically decreasing as the number of malicious nodes increases to the extent that when the number of malicious nodes becomes 10, it can only send 15% of the packets when there is no malicious nodes. Our proposed mechanism AF-AODV introduces an enhancement of about 35% over AODV. In addition to this advantage, AF-AODV has not significantly change for low number of malicious nodes. By combining Figure 1 and Figure 2, we can notice a large enhancement in number of packets that is received by destination specially for large number of malicious nodes. As an example, if the number of malicious nodes is 10, the number of received packets by destinations in AODV is approximately 5600 packets while it is about 20250 packets in AF-AODV which means that the number of received packets is improved by approximately 360%.

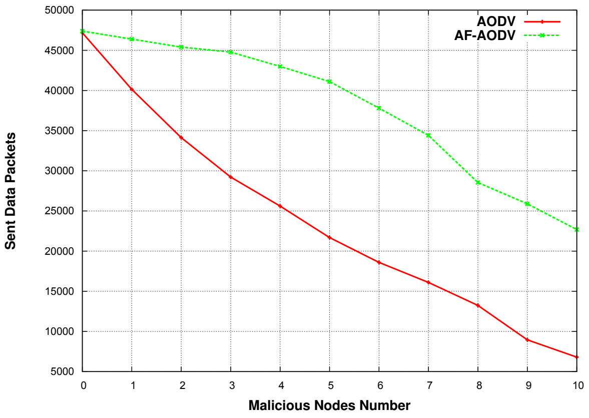

Figure 2. Send Data Packets

Figure 3 shows the effect of flooding attack on the network throughput. Throughput of AF-AODV is better than AODV by approximately 20% for each malicious node. While the throughput of AODV dramatically decreases as the number of malicious nodes increases, AF-AODV slightly decreases for the low number of malicious nodes.

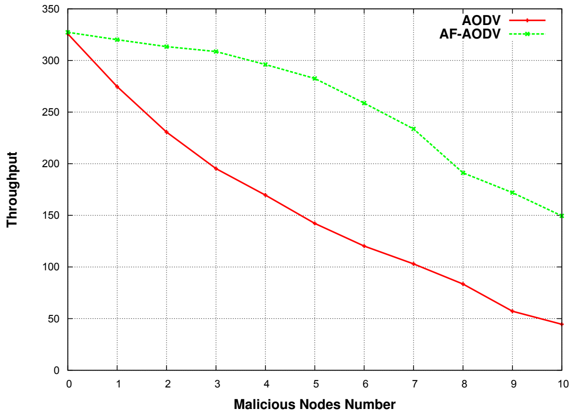

Figure 3. Network Throughput

The effect of flooding attack on the end-end-delay is shown in Figure 4. The result shows that there is no significant change of the delay of AF-AODV while the delay increases as the number of malicious nodes increases.

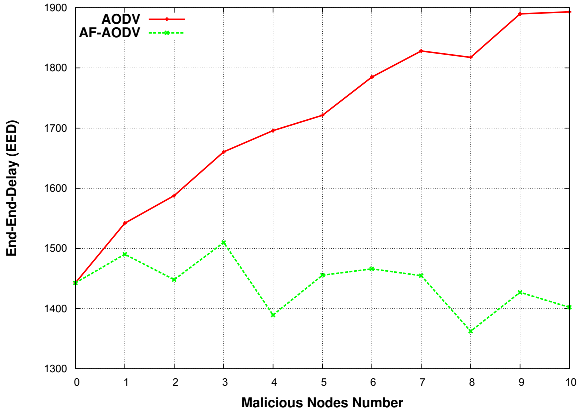

Figure 4. End-to-End Delay

Figure 5 shows the effect of flooding attack on the normalized routing load. The result shows that while the normalized routing load of AODV increases as the number of malicious nodes increases specially for large number of malicious nodes, it has not significant change for AF-AODV.

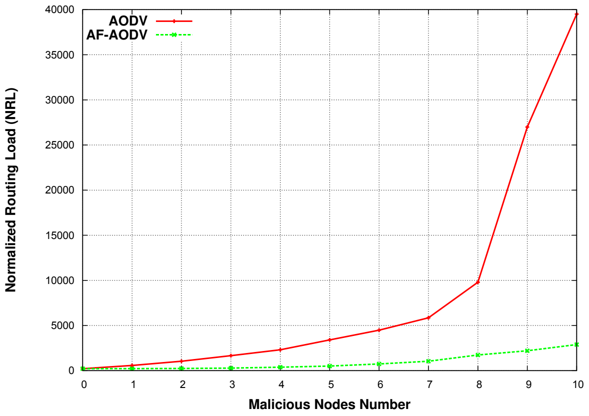

Figure 5. Normalized Routing Load

Figure 6 shows the effect of flooding attack on the routing overhead. The result shows that the routing overhead of AF-AODV has not significantly change for the low number of malicious nodes and slightly increases as the number of malicious nodes increases. On the other hand, it increases dramatically as the number of malicious nodes increases for AODV.

Figure 7 shows the effect of flooding attack on the routing discovery latency. The result shows that the routing discovery latency of AF-AODV is nearly constant regardless the number of malicious nodes. On the other hand, it increases dramatically as the number of malicious nodes increases for AODV.

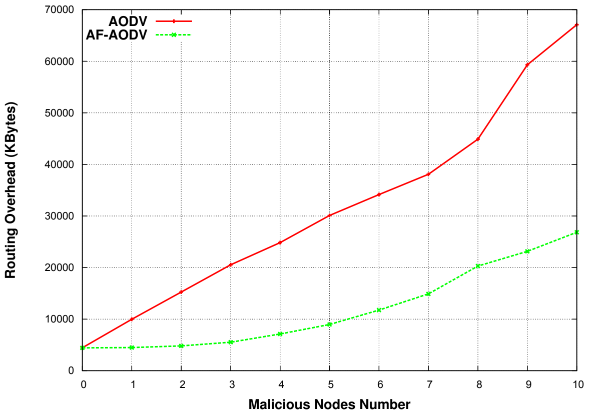

Figure 6. Routing Overhead

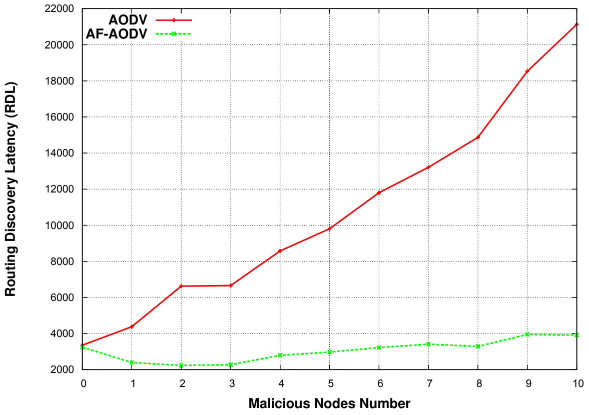

Figure 7. Route Discovery Latency

The number of packets that will be dropped as a result of detecting the presence of malicious nodes is shown in Figure 8. The result shows that while AF-AODV dropped packets increases as the number of malicious nodes increasing, AODV cannot detect the presence of malicious nodes and hence the protocol does not drop packets.

Our simulation shows that regardless the number of nodes and the number of malicious nodes in the network, the malicious node neighbor can detect its presence in a few minutes and the time to detect the last malicious node is increases for sure as the number of malicious nodes increasing. Figure 9 shows the time required by non-malicious nodes to detect the last malicious node in the network.

VI. Conclusion

In this paper, we introduced a new anti-flooding mechanism that can be integrated into any reactive routing protocol in MANET. The proposed mechanism did not use cryptographic techniques which conserves the power and computation resources. Furthermore, the mechanism did not require any additional packets and hence does not incur any additional overhead. As an example, we integrated our anti-flooding mechanism with AODV to study the performance of the

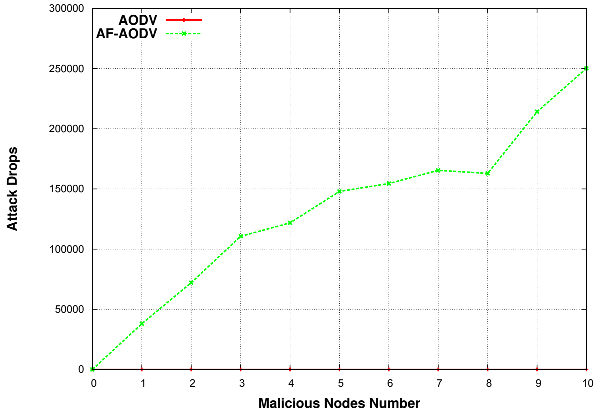

Figure 8. Attack Dropped Packets

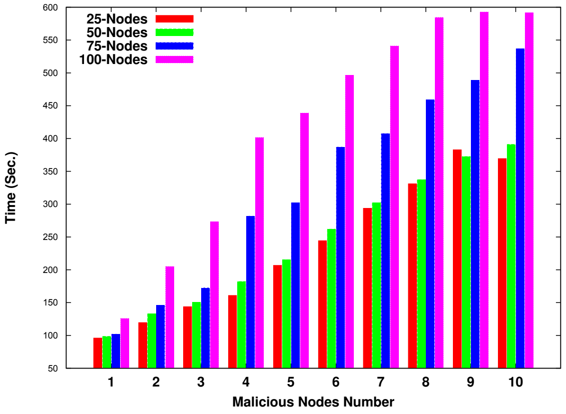

Figure 9. Time Required to Detect the Last Malicious Node by its Neighbors

network under the presence and absence of the mechanism. We validated the performance analysis of our mechanism through NS2 simulations. Simulation results showed that AF-AODV has a remarkable improvement of the network performance in all network metrics than AODV. The proposed mechanism succeeded to detect malicious nodes that try to flood the network within a few minutes regardless the number of malicious nodes and the time they are participating in the network. Future work includes extending this idea to other reactive protocols, and confirming its general applicability.

References

[1] A. Boukerche and et al., "Routing protocols in ad hoc networks: a survey," Computer Networks, vol. 55, no. 13, September 2011, pp. 3032–3080.

[2] M. A. Abdelshafy and P. J. King, "Analysis of security attacks on AODV routing," in 8th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, Dec 2013, pp. 290–295.

[3] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1997, pp. 90–100.

[4] M. A. Abdelshafy and P. J. King, "AODV & SAODV under attack:performance comparison," in ADHOC-NOW 2014, LNCS 8487, Benidorm, Spain, Jun 2014, pp. 318–331.

[5] M. Patel and S. Sharma, "Detection of malicious attack in manet a behavioral approach," in IEEE 3rd International on Advance Computing Conference (IACC), 2013, pp. 388–393.

[6] G. Usha and S. Bose, "Impact of gray hole attack on adhoc networks," in International Conference on Information Communication and Embedded Systems (ICICES), 2013, pp. 404–409.

[7] Y. Guo and S. Perreau, "Detect DDoS flooding attacks in mobile ad hoc networks," International Journal of Security and Networks, vol. 5, no. 4, Dec. 2010, pp. 259–269.

[8] P. Joshi, "Security issues in routing protocols in MANETs at network layer," Procedia CS, vol. 3, 2011, pp. 954–960.

[9] K. Sanzgiri and et al., "Authenticated routing for ad hoc networks," IEEE Journal On Selected Areas In Communications, vol. 23, 2005, pp. 598–610.

[10] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in Symposium on Applications and the Internet Workshops. IEEE Computer Society, 2003, pp. 379–383.

[11] M. G. Zapata, "Secure ad hoc on-demand distance vector routing," SIGMOBILE Mob. Comput. Commun. Rev., vol. 6, no. 3, jun 2002, pp. 106–107.

[12] P. Yi, Z. Dai, Y.-P. Zhong, and S. Zhang, "Resisting flooding attacks in ad hoc networks," in International Conference on Information Technology: Coding and Computing (ITCC), vol. 2, April 2005, pp. 657–662.

[13] B.-C. Peng and C.-K. Liang, "Prevention techniques for flooding attacks in ad hoc networks," in 3rd Workshop on Grid Technologies and Applications (WoGTA 06), Hsinchu, Taiwan, December 2006, pp. 657–662 Vol. 2.

[14] J.-H. Song, F. Hong, and Y. Zhang, "Effective filtering scheme against rreq flooding attack in mobile ad hoc networks," in Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT). Washington, DC, USA: IEEE Computer Society, 2006, pp. 497–502.

[15] V. Balakrishnan, V. Varadharajan, U. Tupakula, and M. Moe, "Mitigating flooding attacks in mobile ad-hoc networks supporting anonymous communications," in 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless), Aug 2007, pp. 29–34.

[16] R. Venkataraman, M. Pushpalatha, R. Khemka, and T. R. Rao, "Prevention of flooding attacks in mobile ad hoc networks," in Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3). New York, NY, USA: ACM, 2009, pp. 525–529.

[17] U. D. Khartad and R. K. Krishna, "Route request flooding attack using trust based security scheme in MANET," International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN), vol. 1, no. 4, 2012, pp. 27–33.

[18] The Network Simulator NS-2, http://www.isi.edu/nsnam/ns/ [retrieved: September, 2014].