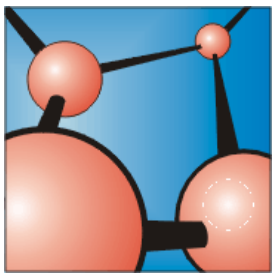
Crossroads
The ACM Student Magazine



A Distributed Security Scheme for Ad Hoc Networks

by ***Dhaval Gada***, ***Rajat Gogri***, ***Punit Rathod***, ***Zalak Dedhia***, ***Nirali Mody***, ***Sugata Sanyal***, and ***Ajith Abraham***

Introduction

In an ad hoc wireless network where wired infrastructures are not feasible, energy and bandwidth conservation are the two key elements presenting challenges to researchers. Limited bandwidth makes a network easily congested by the control signals of the routing protocol. Routing schemes developed for wired networks seldom consider restrictions of this type. Instead, they assume that the network is mostly stable and that the overhead for routing messages is negligible. Considering these differences between wired and wireless network, it is necessary to develop a wireless routing protocol that limits congestion in the network [**1**, **5**, **8**, **9**, **10**, **11**].

This paper proposes minor modifications to the existing Ad hoc On Demand Vector (AODV) routing protocol (RFC 3561) in order to restrict congestion in networks during a particular type of Denial of Service (DoS) attack. In addition to this, it incurs absolutely no additional overhead [**4**]. We describe the DoS attack caused due to Route Request (RREQ) flooding and its implications on existing AODV-driven Mobile Ad hoc Networks (MANET) [**2**, **14**]. To combat this DoS attack, a proactive scheme [**12**] is proposed. We present an illustration to describe the implications of RREQ flooding on pure AODV and the modified AODV protocols. To quantify the effectiveness of the

proposed scheme, we simulated a DoS [6] attack in a mobile environment and study the performance results.

DoS Attack Due to RREQ Flooding

In AODV, a malicious node can override the restriction put by *RREQ_RATELIMIT* [7] (the limit of initiating and forwarding RREQs) by increasing it or disabling it. A node can do so due to its self-control of parameters. The default value for the *RREQ_RATELIMIT* is 10, as proposed by RFC 3561. A compromised node may choose to set the value of *RREQ_RATELIMIT* to a very high number, allowing it to flood the network with fake RREQs [7] and cause a kind of DoS attack. In this type of DoS attack, a non-malicious node cannot fairly serve other nodes due to the network load imposed by the fake RREQs. This leads to the following problems:

- Waste of bandwidth
- Waste of nodes' processing time (more overhead)
- Exhaustion of network resources such as memory (routing table entries)
- Exhaustion of the node's battery power

This further results in degraded throughput. Most of the network resources are wasted in trying to generate routes to destinations that do not exist or that are not going to be used for any communication. This implies that the existing version of AODV is vulnerable to such types of malicious behavior from an internal node (termed a **compromised node**).

Proposed Scheme

Overview

As mentioned earlier, the default value for *RREQ_RATELIMIT* is 10 RREQs per second. This means that each node is expected to observe some self-control on the number of RREQs it sends each second. A compromised node may choose to set the value of *RREQ_RATELIMIT* to a very high number or even disable this limiting feature, allowing it to send a large number of RREQ packets per second. The proposed scheme shifts the responsibility of monitoring this parameter to the node's neighbor, ensuring compliance of this restriction. This technique solves all of the problems caused due to flooding of RREQs from a compromised node. Instead of self-control, the control exercised by a node's neighbor results in preventing an RREQ flood.

RREQ_ACCEPT_LIMIT and *RREQ_BLACKLIST_LIMIT*

The proposal is based on the application of two parameters: *RREQ_ACCEPT_LIMIT* and *RREQ_BLACKLIST_LIMIT*. *RREQ_ACCEPT_LIMIT* denotes the number of RREQs that can be accepted and processed per unit of time by a node. The purpose of this parameter is to specify a value that ensures uniform usage of a node's resources by its neighbors. RREQs exceeding this limit are dropped, but their time stamps are recorded. This information aids in monitoring the neighbor's activities. In the simulations carried out, the value of this parameter was kept as three (i.e., three RREQs can be accepted per time unit). This value can be made to be adaptive, depending upon node metrics such as it memory, processing power, and battery.

The *RREQ_BLACKLIST_LIMIT* parameter is used to specify a value that aids in determining whether a node is acting malicious or not. To do so, the number of RREQs originated or forwarded by a neighboring node per time unit is tracked. If this count exceeds the value of *RREQ_BLACKLIST_LIMIT*, one can safely assume that the corresponding neighboring node is trying to flood the network with fake RREQs. A neighboring node identified as malicious can be blacklisted, preventing further flooding of fake RREQs into the network. The blacklisted node is ignored for a period of time given by *BLACKLIST_TIMEOUT,* after which it is unblocked. The proposed scheme has the ability to block a node for the *BLACKLIST_TIMEOUT* period on an incremental basis. The *BLACKLIST_TIMEOUT* period is doubled each time the node repeats its malicious behavior.

In our simulations, the value of *RREQ_BLACKLIST_LIMIT* is kept as 10 (i.e., more than 10 RREQs per time unit results in flooding activity). By blacklisting a malicious node, all neighbors of the malicious node restrict the flood of RREQs. In addition, the malicious node is isolated by this distributed defense and cannot hog its neighbors' resources. The neighboring nodes are therefore free to entertain the RREQs from genuine nodes. Nodes that are confident about the malicious nature of a particular node can avoid using it for subsequent network functions. In this way, genuine nodes are saved from experiencing the DoS attack.

Advantages of the Proposed Scheme

- The proposed scheme 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 (RFC 3561).

- The proposed scheme is more efficient in terms of the resultant routes established, resource reservations, and computational complexity.
- If multiple malicious nodes collaborate, they in turn will be restricted and isolated by their neighbors, because they monitor and exercise control over forwarding RREQs by nodes. Hence, the scheme successfully prevents Distributed DoS (DDoS) attacks.

Algorithm Illustration

Figure 1 depicts the pure AODV routing protocol when an internal malicious node (black) launches a DoS attack by flooding the network with RREQs. The blue nodes depict two genuine nodes that want to communicate with each other. The optimal route consists of four intermediate nodes, including the malicious node and three of its neighbors. The malicious node floods the network by generating more than 10 RREQs per second, as shown. Its immediate (non-malicious) neighbors observe the *RREQ_RATELIMIT* and hence each forward only 10 RREQs. Since at most three RREQs will be accepted from these nodes per second, the neighbors of these nodes need to forward more than seven RREQs and their neighbors in turn need to forward more than four RREQs, as shown. Because the resources of the malicious node's neighbors are completely occupied in processing and forwarding the RREQs originating from it, the route between the blue nodes, if it is established, will consist of greater number of intermediate nodes. Thus, in effect a DoS attack is launched as the genuine nodes are deprived of the services of nodes whose resources are wasted due to flooding.

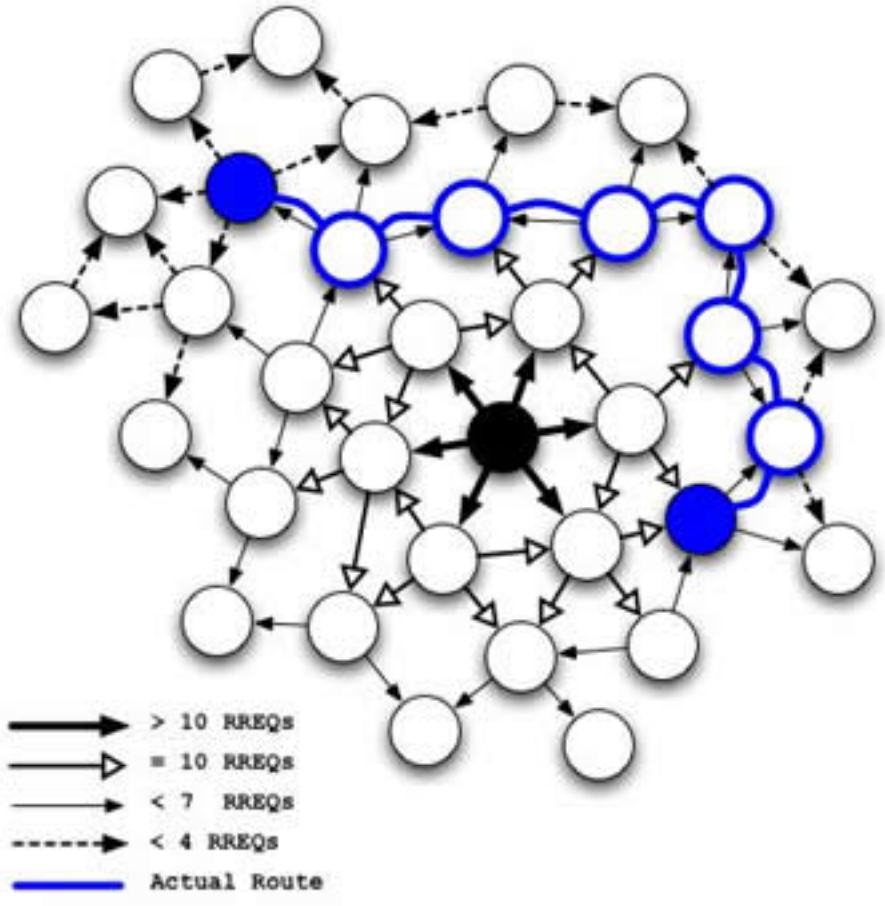


Figure 1: Simulation Network for the DAPRA 1998 data set.

Figure 2 illustrates the proposed AODV scheme. As shown in the figure, the malicious node (black) floods RREQs in the network and two genuine nodes (blue) want to communicate with each other. In this scheme, the number of RREQs that can be accepted from a neighbor is limited. Due to this, the neighbors of the malicious node will only accept and forward three received RREQ packets within a time interval of one second. This rate limit exists to ensure fair distribution of a node's resources to all of its neighbors. Moreover, whenever the malicious node crosses the *RREQ_BLACKLIST_LIMIT* of 10 RREQ packets within a time interval of one second, its neighbors will blacklist it. Because of this, in addition to limiting the clogging of network resources, the proposed scheme also isolates the malicious node. The route established in this scheme is expected to be the optimum route, which consists of the minimum number of intermediate nodes. Thus, no DoS attack is experienced in the developed scheme.

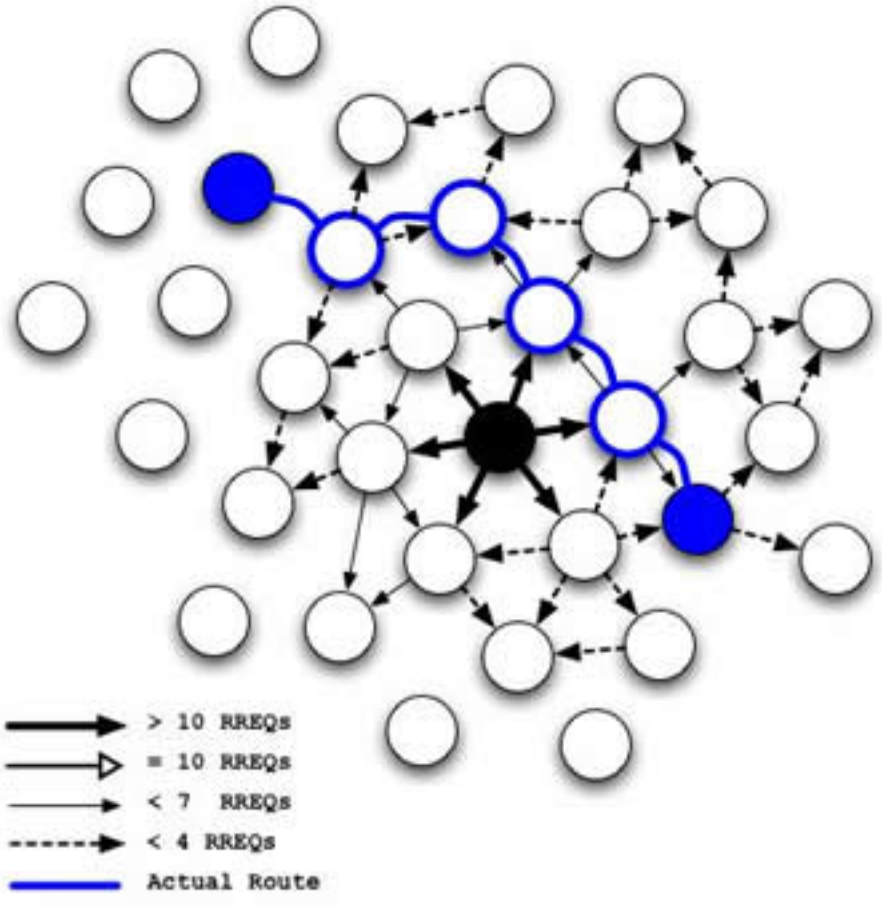


Figure 2: Illustration of the proposed AODV.

Experiments and Analysis

The NS-2 simulator [**3**, **13**] was used for the implementation of the proposed scheme. The IEEE 802.11 [**15**] protocol was used for the MAC layer. The AODV protocol incorporated in NS-2 by Uppsala University was used as the base protocol. Modifications were made to this version of AODV protocol that confirm to RFC 3561. TCP was used as the transport protocol. Radio transmission range was set as 250 meters. Traffic sources used Constant-Bit-Rate (CBR) and the field configuration was 2000 m. x 2000 m. with 69 nodes.

Traffic Scenario

Node 0 was configured as the malicious node. It started flooding the network with fake RREQs at a simulated time of one second for 17 seconds. The traffic was generated such that the source and destination pairs were randomly spread over the entire network. The other source-destination pairs are shown in **Table 1**.

Source	Destination	Simulation Time
Node 48	Node 20	11-16 sec
Node 18	Node 27	5-12 sec
Node 31	Node 66	6-11 sec
Node 45	Node 16	9-12 sec

Table 1: Traffic generation summary.

The performance evaluation of the proposed detection scheme involves study of two different aspects:

- Performance of the original AODV protocol in the presence of compromised nodes.
- Performance of the proposed AODV protocol in the presence of compromised nodes.

Each simulation was carried out for 17.2 seconds. The results for both cases were observed, and the following section gives the parameters that were measured for both the original and the modified protocols.

Network Simulation Metrics

The metrics below are the important determinants of network performance, and have been used to compare the performance of the proposed scheme with the performance of the original protocol. This study has been done to show that the proposed scheme enhances the security of the routing protocol without causing substantial degradation in network performance.

1. **End-to-End Delay:** Average time difference (in seconds) between the time of the packet receipt at the destination node and the packet transmission time at the source node.
2. **Round Trip Time (RTT):** Time difference between the receipt of the acknowledgement from the destination node to the source node and the time of transmission of the original packet at the source node.
3. **Average simulation processing time at nodes for a packet:** Time difference between the packet forwarding time and the packet receipt time at a given node.

4. **Average number of nodes receiving packets:** Sum of numbers of all the intermediate nodes receiving packets sent by all the source nodes divided by the number of received packets at all the destination nodes.
5. **Average number of nodes forwarding packets:** Sum of numbers of all the intermediate nodes forwarding packets sent by all the source nodes divided by the number of received packets at all the destination nodes.
6. **Delays between current and other node:** Shows the end-to-end delays (in seconds) between the current node (sender) and another node (receiver).
7. **Number of data packets dropped:** The number of data packets dropped at any given node. This is an important parameter because if the number of dropped packets increases, the throughput decreases.
8. **Throughput:** Sum of sizes (bits) or number of generated/sent/forwarded/ received packets, calculated at every time interval and divided by its length. Time interval length is equal to one second by default.

Performance Evaluation

This section consists of the results for the test cases. The recorded values are obtained by averaging over three runs for each test case.

Acknowledgement Packet Receive Time Versus RTT

As simulation time increases, the network resources available to the nodes vary. The availability of network resources is one of the parameters that help in determining the RTT. **Figure 3** shows a graph of acknowledgment packet receive time versus RTT.

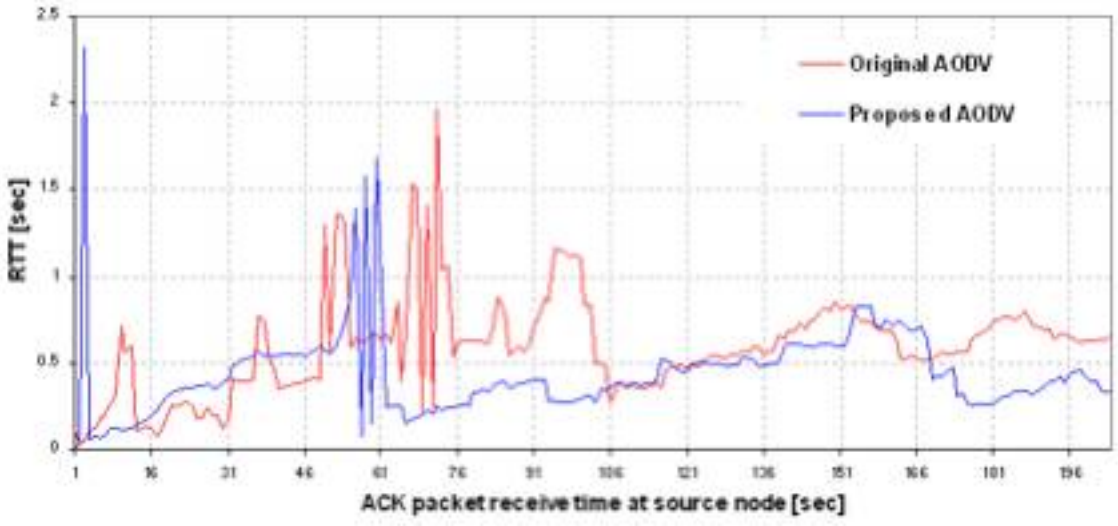


Figure 3: Acknowledgement packet receive time versus round trip time.

It is evident from **Figure 3** that as time proceeds, RTT is less in the proposed AODV scheme than in the original scheme. This is because of the limit imposed on the number of RREQ packets flooding in the network by malicious nodes and the decreased number of intermediate nodes in the routes between genuine nodes.

Dropped Packet Sum

The number of packets dropped at a given time in the simulation run determines the efficiency of the protocol. **Figure 4** shows the number of dropped packets throughout the simulation.

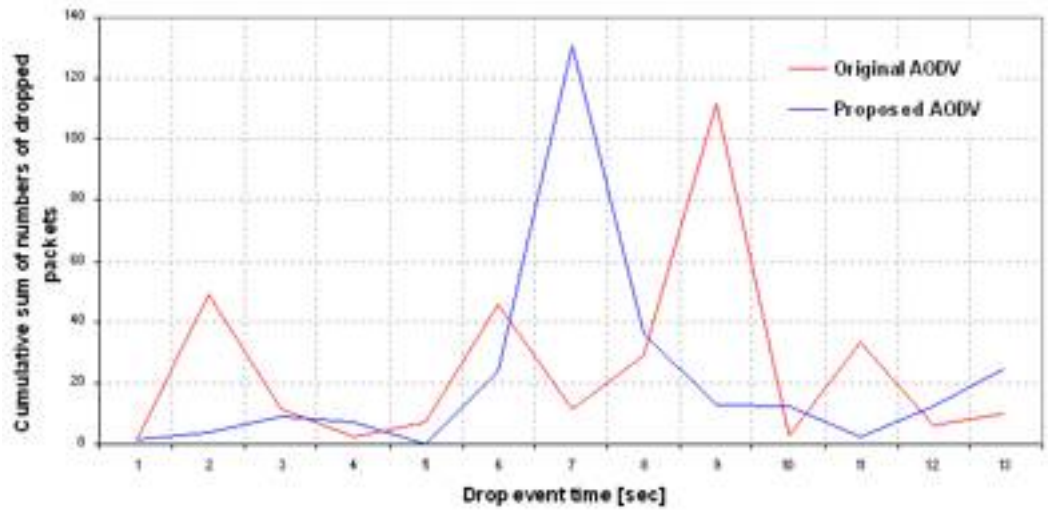


Figure 4: Throughput of dropping packets.

From **Figure 4**, it is seen that overall, the number of packets dropped using the proposed scheme is less than the number of packets dropped in the original scheme. In the initial stages, the large number of drops in the original scheme is due to the fact that the flooding of RREQ packets in the network causes congestion and the route formation for genuine requests is delayed. Thus, the buffered data packets time out and are dropped. During the later stages, the unavailability of network resources causes the data packets to be dropped. The improvement in the proposed scheme comes from the fact that there is optimum utilization of the network resources without any overload, leading to comparatively fewer packet drops.

End-to-End Delay Versus Packet Size

Figure 5 depicts how the proposed method affects the average end-to-end delay.

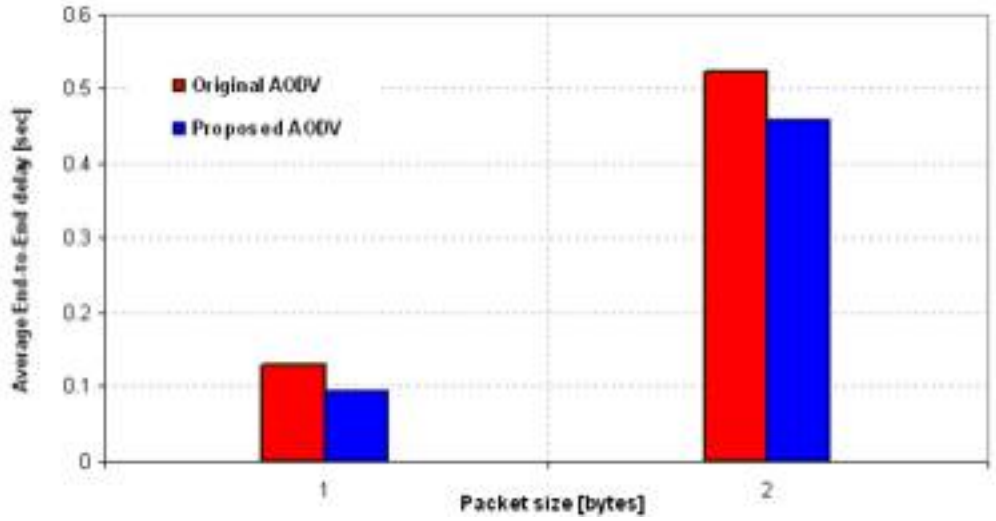


Figure 5: Packet size versus average end-to-end delay.

The delay in the cases of both data and AODV packets is less in the proposed scheme than in the original AODV.

Throughput of Generating Packets at an Intermediate Node in the Route

Figure 6 shows the throughput of generating packets at an intermediate node (numbered 48 in the sample simulation scenario) versus simulation time. The graph reflects the simulation time for which an intermediate node in the route generated packets in original AODV as compared to the proposed AODV. In other words, it depicts how long the route through the intermediate node was valid during simulation.

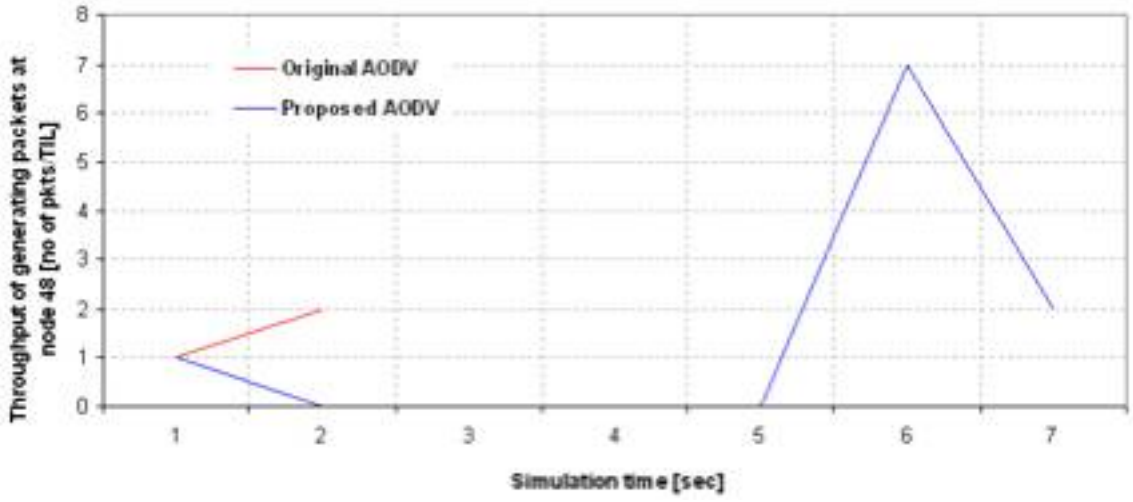


Figure 6: Throughput of generating packets at node 48.

In the scenario with the original AODV protocol, the routes become invalid quickly when no replies (ACKs) for data packets are received due to the clogging of network

resources and the DoS attack. This is reflected in the graph by having nonzero throughput of generating packets for only two seconds of simulation time, after which the route through the intermediate node 48 becomes invalid, resulting in no throughput of generating packets. However, in the scenario with the proposed AODV, the route through node 48 remains valid for a longer period of simulation time and hence it has nonzero throughput of generating packets until seven seconds into the simulation (shown by blue line in the graph). It can be inferred from the graph that routes remain valid for longer periods of time under the proposed scheme.

Packet Size Versus Simulation Time

The comparison of simulation processing times as illustrated in **Figure 7** reveals that the proposed scheme incurs no additional overhead compared with the original scheme.

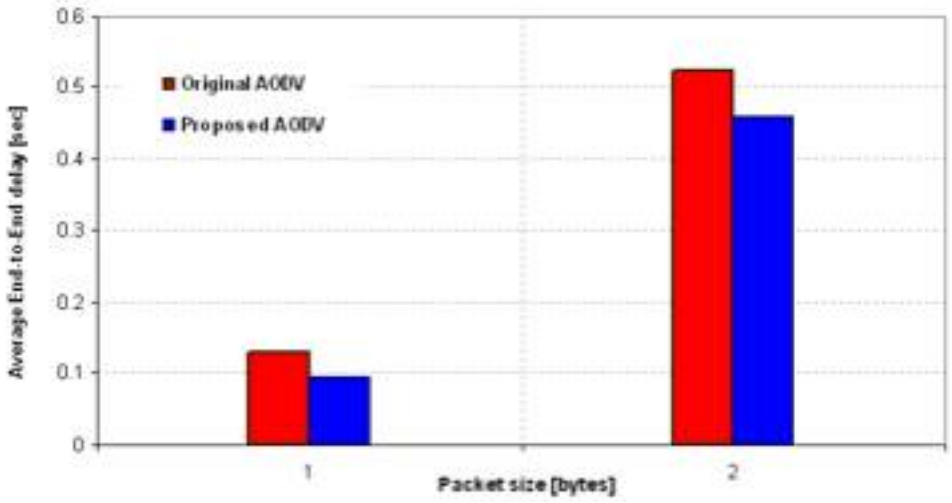


Figure 7: Average simulation processing time.

Network Information for Sample Scenario

Table 2 gives the comparative study of network information for the original AODV and the proposed AODV protocols.

	Original AODV	Proposed AODV
Average End-to-end delay (seconds)	0.32539	0.27576
Receiving packets	0.4356328083	0.3580786026
Forwarding packets	0.4285714286	0.3499688085

Average RTT	0.58819	0.45346

Table 2: Overall network simulation results.

Conclusions and Future Work

The DoS attack caused due to RREQ flooding in ad hoc networks can be successfully detected in the proposed scheme. The scheme can accurately detect malicious nodes in a network. The malicious nodes identified are blacklisted and none of the genuine nodes in the network are wrongly accused of misbehaving. In the proposed scheme, there is an enhancement in the performance of the network in the presence of compromised nodes.

Mobile computing and communication is a new field that is capturing the imaginations of researchers worldwide. Because of this, the potential for enhancements and improvements is enormous. An immediate enhancement might be to make the limit-parameters adaptive in nature. This can be done by performing calculations based on parameters such as memory, processing capability, battery power, and the average number of requests per second in the network. Further, the protocol can be made secure against other types of possible DoS attacks that threaten it via various techniques.

Acknowledgements

The authors would like to thank Charles Perkins for answering our questions about the intricate details of the operation of the AODV protocol. Also, many thanks are due to Ryan Hogg and Srinath Perur for providing insights regarding the working of NS-2 during the implementation stage.

References

1

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), pp. 85-97, Oct .1998.

2

Corson, S. and Macker, J. Mobile ad hoc networking MANET): Routing protocol performance issues and evaluation considerations. **http://www.sunsite.dk/**

RFC/rfc/rfc2501.html, 1999.

3

Fall, K. and Varadhan, K. (Eds.), NS-2 notes and documentation. http://www-mash.cs.berkeley.edu/ns, 1999 (accessed on May 03, 2004).

4

Jacquetand, P. and Viennot, L. Overhead in Mobile Ad-hoc network Protocols. INRIA Research Project RR-3965, 2000.

5

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

6

Lundberg, J. Routing Security in Ad Hoc Networks.

7

Perkins, C. E. Terminology for Ad-Hoc Networking, Draft-IETF-MANET-terms-00. txt, November 1997.

8

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).

9

Perkins, C. E. and Royer, E. M. Ad-hoc on-demand distance vector routing. In 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90−100, 1999.

10

Perkins, C. E., Royer E. M., and Das, S. R. Ad hoc On-Demand Distance Vector (AODV) Routing. http:// www.ietf.org/internet-drafts/draft-ietf-manet-aodv-05.txt, 2001 (accessed on May 03, 2004).

11

Rahman, A. "Security Issues in Mobile Systems." http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/sec-in-ob.html. 1995. (accessed May 03, 2004).

12

Royer, E. M. and Toh, C. K., A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications, vol. 6:46−55, Apr 1999.

13

UCB/LBNL/VINT, Network Simulator - NS, http://www-mash.cs.berkeley.edu/ns, 1995 (accessed on May 03, 2004).

14

Venkatraman, L. Secured Routing Protocol for ad hoc Networks. http://www.

ohiolink.edu/etd/view.cgi?ucin982169866 (accessed on May 03, 2004)

15 Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11-1997, 1997.

Biographies

Dhaval Gada, Rajat Gogri and Zalak Dedhia are currently pursuing their Bachelor's degrees in Information Technology at the K. J. Somaiya College of Engineering at Mumbai University in India.

Punit Rathod of the Vivekanand Education Society's Institute of Technology and Nirali Mody of the Thadomal Sahani Engineering College are also currently pursuing their Bachelor's degrees in Information Technology from Mumbai University in India. Their common research interests include wireless and ad hoc networks, and routing protocols and security concerns of wireless communication networks and ad hoc networks.

Sugata Sanyal is a distinguished professor in the School of Technology and Computer Science at the Tata Institute of Fundamental Research in India since 1973. He received his Ph.D. from Mumbai University and his M.Tech from IIT, Kharagpur. His current research interests include security in wireless and mobile ad-hoc networks, distributed processing, and scheduling techniques.

Ajith Abraham is currently a faculty member of the Computer Science Department at Oklahoma State University. He received his Ph.D. from Monash University in Australia and his MS from NTU, Singapore. His research interests are in computational intelligence, information security, and web intelligence.