Mitigating Malicious Control Packet Floods in Ad Hoc Networks

Saman Desilva Rajendra V. Boppana
Computer Science Department
The Univ. of Texas at San Antonio, San Antonio, TX 78249
sdesilva@cs.utsa.edu boppana@cs.utsa.edu

Abstract—**In this paper, we investigate the impact of hacker attacks by malicious nodes on the overall network performance. These malicious nodes mimic normal nodes in all aspects except that they do route discoveries much more frequently than the other nodes. We show, using simulations, that the basic route discovery mechanism used in many ad hoc network protocols can be exploited by as few as one malicious or compromised node to bring down the throughput dramatically. We propose an adaptive statistical packet dropping mechanism to mitigate such situations and reduce the loss of throughput. The proposed mechanism works even when the identity of the malicious nodes is unknown and does not use any additional network bandwidth. It is simple to implement and maintains or improves network throughput when there are no malicious nodes but the network is congested with excess traffic.**

I. Introduction

Recent advances in wireless communication technology and portable devices have generated a lot of interest in mobile ad hoc networks (MANETs). A MANET is a collection of wireless devices moving in seemingly random directions and communicating with one another without the aid of an established infrastructure. So the communication protocols for MANETs are designed to work in peer-to-peer networking mode. To extend the reachability of a node, the other nodes in the network act as routers. Thus, the communication may be via multiple intermediate nodes from source to destination. Because of node mobility, network topology and hence the routes change frequently. So designing routing protocols for ad hoc networks is a challenging problem.

The current design and intended use of MANETs are such that nodes are susceptible to various types of hacker attacks. Malicious nodes may become part of actively used routes and disrupt network operation. Extensive research was done to handle situations where malicious nodes provide incorrect information to other nodes or drop or alter data and control packets to disrupt a MANET. Ning and Sun [12] presented several insider attacks on MANET, using AODV protocol as example. Wang et al. [19] analyzed and demonstrated that false distance vector and false destination sequence attacks can decrease delivery ratio by 75% when the AODV routing protocol is used. Hu et al. [8] introduced a $rushing$ attack that result in denial-of-service attack on ad hoc networks, if an on-demand routing protocol is used. Similarly, Marit et al. [10] introduced $watchdog$ and $pathrater$ to identify misbehaving nodes (dropping control packets) in ad hoc network.

This research has been partially supported by NSF grant EIA-0117255 and AIA grant F30602-02-1-0001.

In this paper, we evaluate the affect of control packet flooding by one or more malicious nodes on a MANET's performance. We show that one or more malicious nodes flooding the MANET with control packets related to bogus route discoveries can cause a sharp drop in network throughout. These malicious nodes behave like the normal nodes in all aspects except that they initiate frequent control packet floods. This type of attack is hard to detect since any normal node with frequently broken routes could legitimately initiate frequent route discoveries. To find a solution, we have looked at the large amount of work done in the literature to reduce number of transmissions for network-wide floods, generally called broadcast management techniques [11], [14], [20]. These techniques attempt to reduce redundant broadcasts using passive hearing and keeping track of neighbors. We show that these broadcast management techniques do not offer any significant relief from the malicious control packet floods. We have not seen any prior study evaluating the impact of route flood attacks.

In this paper, we introduce a simple rate based control packet forwarding mechanism to mitigate malicious control packet floods. This technique has no adverse impact in the absence of malicious control packet floods, but stops any harmful effects of frequent control packet floods without the need to identify the malicious nodes. Using simulations, we show that the proposed technique is very effective even when there is only one malicious node generating as little as 1 RREQ packet/s.

The rest of the paper is organized as follows. Section 2 provides the background on the routing protocols for mobile ad hoc networks. Section 3 evaluates the effects of route flooding on MANET. The effectiveness of broadcast management techniques and an adaptive rate based mechanism these technique are evaluated in Section 4. Section 5 concludes the paper.

II. Routing Protocols for Ad Hoc Networks

Routing protocols can be divided into proactive and reactive (or on demand) categories. Several different dynamic routing protocols in both proactive and reactive protocol categories [17], [9], [3], [13] were proposed for MANETs. The advantages and disadvantages of proactive and reactive protocols are studied in detail in [4], [2], [7]. Both proactive and reactive protocols can suffer from control packet floods

caused by malicious nodes. In this paper, we use an on demand routing protocol known as AODV [16] for performance analyses.

A. Routing in on demand Protocols

A table-based dynamic routing protocol maintains a routing table (essentially, <destination node, next hop, no. of hops to destination>-tuples) in each node. When a node attempts to send a data packet to a destination for which it does not already know the route, it uses a "route discovery" process to dynamically obtain a route. The route discovery works by flooding the network with route request (RREQ) control packets. A node, say, x, receiving a RREQ, rebroadcasts it, unless it has already seen it from another neighbor or it has a route to the destination indicated in the RREQ. If the received RREQ is a duplicate, node x drops it. If node x has the route because it is the destination or it has learned it in another route discovery, then it replies to the RREQ with a route reply (RREP) packet that is routed back to the original sender of the RREQ.

A drawback of flooding based route discovery process is the high control overhead. Each RREQ initiated by a node results in n broadcasts in the MANET, where n is the number of nodes in the MANET. As the mobility and load of the network increases, the control packets used for route discoveries may consume more bandwidth than the data packets. So several heuristics are used to reduce the number of control packet broadcasts needed to be done to discover a route. In particular, Williams et al. [20] discuss several techniques to broadcast packets efficiently, but none of these techniques are adopted by any of the present on demand routing protocols.

When the network is saturated, increasing the offered load causes rapid decrease in achieved throughput due to $false$ route breaks. In a false route break, the sender assumes the next hop does not exist, though it is still within the radio range of the sender but did not respond owing to busy wireless channel [5]. This creates a vicious circle. False route breaks cause sending nodes to initiate frequent route discoveries, which further increases wireless channel usage. Since a broken route disrupts data flow, control packets are given higher priority over data packets in transmitting in order to repair broken routes as quickly as possible. So at high loads, the wireless channel usage can be completely dominated by the control packets used for route discoveries [6]. This potential weakness of on-demand routing protocols could be exploited by malicious nodes.

III. Impact of Malicious Broadcasts

To evaluate the impact of excess control packet floods by malicious nodes, we simulated MANETs using the Glomosim simulator, version 2.03 [1]. 100 randomly placed nodes moving at a randomly chosen speed in [1,19] meters/s in a 1200 $\times$ 1200 terrain are simulated. The random waypoint mobility pattern was used to model node movements [18]. 50 CBR connections sending 512-byte data packets

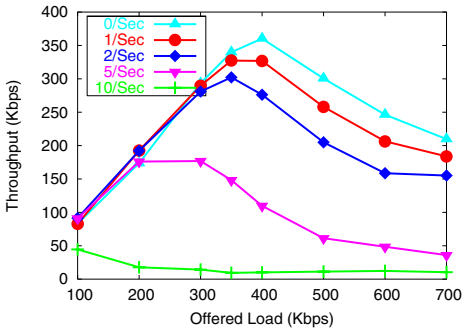

Fig. 1. Loss of throughput with bogus route discoveries by a malicious node. The route discoveries are initiated at the rate of 1,2,5 or 10/Second.

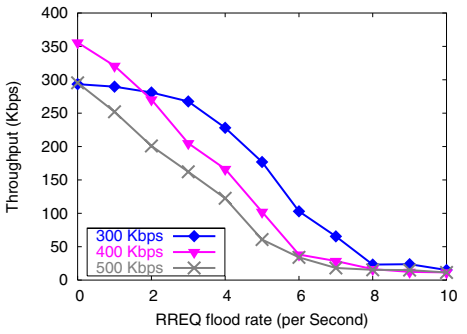

Fig. 2. Alternate view of throughput loss with bogus route discoveries.

were used to simulate traffic on the network. The load offered to the network was varied by changing packet interarrival times. Each simulation was run for 600 seconds (100 seconds of warm-up during which no statistics are collected and 500 seconds of post-warm-up simulation). Each simulation was repeated 9 times with different initial placement of nodes. Averages of these ten runs are reported in the following results. The AODV routing protocol is used for all simulations.

One of the nodes (that is neither a sender or receiver of CBR data) is changed into a malicious node, which floods the network with bogus route discoveries at a rate of 1 to 10 RREQs/s. A similar node is selected to be the destination for which this malicious node initiates bogus route discoveries. (Normal network performance is obtained when the specified attack rate is zero RREQs/s.) This node behaves like any other node in the network in all aspects except that it sends frequently RREQ packets, which are used for route discovery. This type of attack may be hard to detect since any normal node with a broken route could legitimately initiate multiple RREQ broadcasts in a short period of time.

The malicious node drops any route information received in response to its route discoveries and continues to initiate route discoveries at the specified rate. Figure 1 shows achieved throughput as a function of offered load and malicious node's route discovery rate. For traffic loads at or beyond saturation, any RREQ rate by the malicious node

reduces the throughput rapidly. At 10 RREQs/second, the peak throughput is reduced by 84%. For a different prospective of the network performance, Figure 2 shows achieved throughput with respect to increasing rates of route floods with offered load kept constant. We used 300, 400 and 500 Kbps to represent loads before, at and beyond network saturation. For a given offered network load, say, 300 Kbps, the network becomes nearly unusable as the rate of RREQs broadcasted by the malicious node increases. The only way to achieve any usable performance is to reduce the data rates. For network loads below saturation, there is excess bandwidth available to absorb, up to some extent, the control packet floods caused by the malicious node. For network loads at or beyond network saturation, the impact of RREQ floods by the malicious node is compounded. Even 1 RREQ/s by the malicious node causes measurable drop in throughput.

The security enhancements such as those used for secure AODV [15] do not handle this type of attack since the malicious node is not forging any information. A static limit on RREQs generated by a node can hurt the performance by restricting the route discovery capability of genuine nodes if the limit is too low. A high static limit is not effective.

IV. Mitigating Broadcast Attacks

In this section, we evaluate the effectiveness of a broadcast management technique and describe an adaptive control packet filtering technique to mitigate bogus control packet floods.

A. Broadcast management techniques

The broadcast management techniques attempt to reduce the total broadcasts in the network by ensuring that a node forwards/rebroadcasts a received RREQ only when it is determined to be non-overlapping with those of its neighbors. Such techniques may be used reduce the effects of broadcast attacks, though they can not identify or detect attackers. Several broadcast management techniques are discussed and analyzed for various network conditions in [20]. In our study, we used the random assessment delay (RAD) technique to manage route request efficiently.

RAD requires that, upon receiving a new RREQ packet, a node must keep track of redundant packets received over a short period of time, $\leq$ 200 milliseconds. If the number of redundant broadcasts exceeds a preset count (we used 5), then RREQ is not relayed. Otherwise, it will be transmitted. RAD is simple to implement and redundant broadcast count can be checked at the MAC layer level prior to transmitting a RREQ for higher performance [20].

Figures 3 and 4 show throughputs achieved with the RAD technique in the presence of various attack rates by the malicious node as in the previous simulations. From these graphs, it is clear that RAD improves the performance marginally compared to the default flooding technique. RAD achieves the purpose of broadcasting efficiently and improves throughput in a normal network without broadcast attacks. But it does not distinguish bogus broadcasts from normal ones, and the number of broadcasts reduced is not sufficient to mitigate the negative impact of bogus RREQs. So we investigate a more effective mechanism that attempts to curb the propagation of RREQs by frequent senders with little or no effect on the handling of RREQs from other nodes.

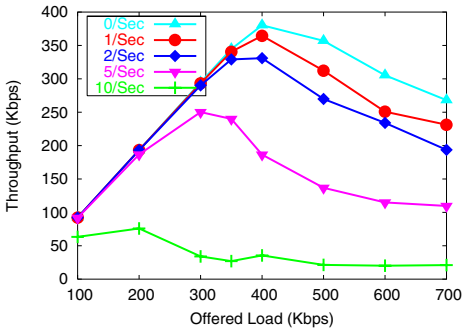

Fig. 3. Benefit of RAD broadcast technique in the presence of bogus route discoveries.

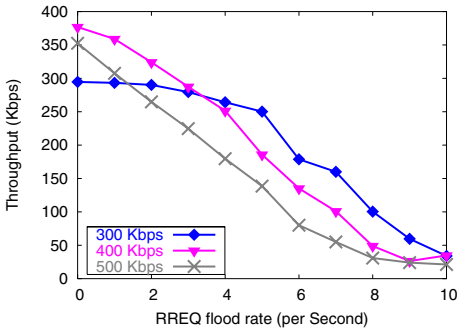

Fig. 4. Alternate view of RAD performance in the presence of bogus route discoveries and constant offered load.

B. Rate based filtering of excessive broadcasts

We propose a simple, distributed, and adaptive technique to reduce the effects of broadcast attacks using RREQ. The proposed technique uses statistical analysis to detect misbehaving nodes and reduces their impact on network performance. We assume that all RREQs are authenticated. So every node must include its ID and authentication information, which we assume cannot be forged. So malicious nodes are at one time trusted nodes that have the appropriate authentication, but attack the network when the opportunity arises.

In our design, each node monitors the route requests it receives. Each node maintains a count of RREQs received for each RREQ sender during a preset time period ($\delta\tau$). At the end of the time period, the node computes the rate at which it has been receiving route requests from each sender and smoothed average, $savg$, of the same using (1) and (2). The node also computes average rate of RREQs per sender using (3) and smoothed average, $nodeavg$, of the same using (4).

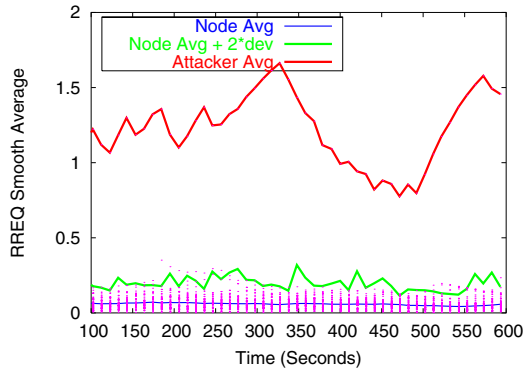

Fig. 5. Smoothed averages of RREQ attacks and overall broadcast rate at node 93, close to the malicious node. The smoothed averages for normal nodes shown as dots, which are at the bottom of the graph.

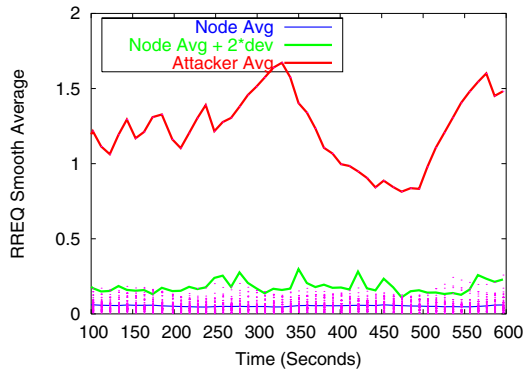

Fig. 6. Smoothed averages of RREQ attacks and overall broadcast rate at node 65, far from the malicious node.

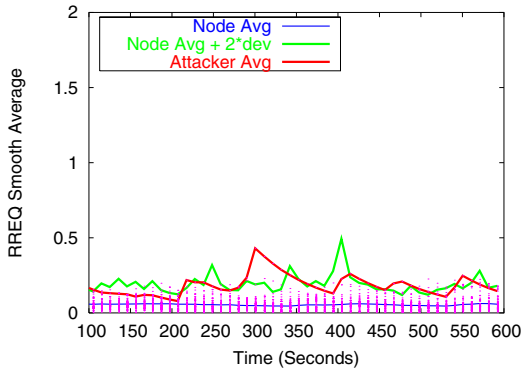

Fig. 7. Smoothed averages of RREQ attacks and overall broadcast rate at node 93 after applying the statistical rate control mechanism. The attacker is node 3.

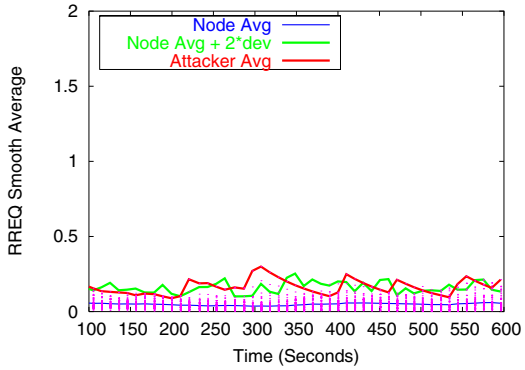

Fig. 8. Smoothed averages of RREQ attacks and overall broadcast rate at node 65 after applying the statistical rate control mechanism.

In addition, the node also computes the savg deviation of all RREQ sources at the end of the time period using Equation 5 repeatedly for each RREQ sender. This is denoted as nodedev.

$$rate_i = RREQCount_i/\delta\tau \quad (1)$$

$$delta_i = rate_i - savg_i$$

$$savg_i \leftarrow savg_i + g \times delta_i \quad (2)$$

$$noderate = \frac{TotalRREQCount/\delta\tau}{\#ofRREQsenders} \quad (3)$$

$$delta = noderate - nodeavg$$

$$nodeavg \leftarrow nodeavg + g \times delta \quad (4)$$

$$nodedev \leftarrow nodedev + h(|delta_i| - nodedev) \quad (5)$$

The nodeavg and nodedev calculations are based on the TCP retransmission timeout (RTO) calculations [21] with $g, h \leq 1$. As a starting point, the value of g is set as $\frac{1}{8}$ and h is chosen to be $\frac{1}{4}$. We have experimented g values of $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{1}{8}$ and found $\frac{1}{8}$ the best value for our network conditions.

We simulated the example 100-node MANET with one malicious node to study the rates of RREQ sources and node averages. We simulated 300 Kbps traffic load from 50 CBR connections and a 2 RRRQs/s attack rate by the malicious node. Figure 5 represents smoothed averages in a node that is mostly a neighbor of the malicious node and Figure 6 the same in a node that is mostly not a neighbor to the malicious node. As expected both nodes, have a high smoothed average count (1 to 1.5 RREQs/s) for the malicious node. The smoothed averages for all other nodes is less than 0.2 RREQs/s.

To distinguish between malicious RREQ floods and those by normal nodes, we calculate a cut-off rate (denoted, CutOffRate) as given in (6). The RREQs from a sender whose smoothed average rate is above the CutOffRate will be dropped without forwarding. Dropped RREQs are counted in computing smoothed averages, however. Examining Figures 5 and 6, we note that very few, if any, of the RREQs sent by normal nodes are dropped with this rule.

$$\text{CutOffRate} = nodeavg + 2 \times nodedev \quad (6)$$

We applied this statistical RREQ rate control technique to the example network and reran the simulations. The RREQs seen by the near node given in Figure 7 indicate that malicious node's smoothed average is substantially reduced but it is still beyond the CutOffRate. Therefore, this node will not relay malicious node's RREQs to its neighbors. Rising

TABLE I

PERCENTAGE OF MALICIOUS NODE RREQS DROPPED.

Offered Load (Kbps)	Attack Rate	
	1 RREQ/sec	10 RREQ/sec
100	93.6	99.4
200	87.4	99.1
300	83.1	98.1
400	76.3	98.0
500	77.3	99.0
600	80.4	99.9
700	85.7	99.9

slopes indicate that the node is receiving RREQs from malicious node, and falling slopes indicates no RREQs are seen from the malicious node. The far node, whose averages are given in Figure 8, receives only a few RREQs from malicious node, because most of its RREQs are dropped after 1 hop by its neighbors. Indeed, the malicious node's RREQ rate as seen by this node is only slightly above the CutOffRate for this node. So based on these graphs, the proposed rate control mechanism seems to be curbing RREQs from the malicious node but has no impact on the other nodes. Table I gives the fraction of RREQs from the malicious node dropped in proportion to received. It is clear from this table that, prior to saturation, good nodes drop over 90% of the malicious RREQs received by them. At and beyond saturation, normal nodes that need to establish long routes also send abnormally high RREQs, which tends to increase the overall node average. This in turn reduces the fraction of malicious RREQs dropped in low-rate attacks.

Figures 9 and 10 present network throughputs for varying traffic loads. Figure 9 shows that the proposed filtering technique effectively eliminates any drag on performance by the bogus route discoveries (compare with Figure 1). The alternative view in Figure 10 shows that throughput loss is very minimal even for RREQ rates as high as 10 RREQs/s by the malicious node when the offered load is constant.

Implementing the rate control mechanism

The proposed technique is simple and inexpensive to implement as part of the routing algorithm. Each node is required to maintain two values for each RREQ source ($savg$ and $count$). At the end of each sampling period smoothed averages are calculated. These calculations are simple and require only a few CPU cycles. Also each node needs to maintain the overall smoothed average and deviation of averages given by (4) and (5). The deviation calculation requires iterative calculations with one iteration for each RREQ sender.

It is noteworthy that the rate control mechanism does not significantly hurt the performance of a normal network (see Figure 11). In a normal network, the rate control mechacnism drops excessive RREQs from hyperactive nodes. Our

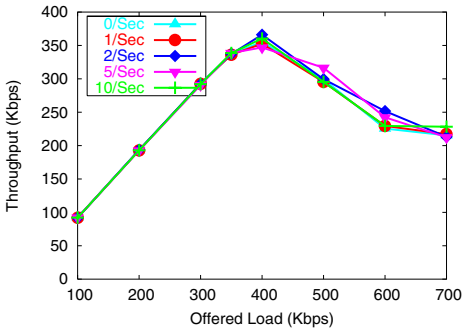

Fig. 9. Benefit of statistical rate control mechanism in the presence of various rates of bogus route discoveries.

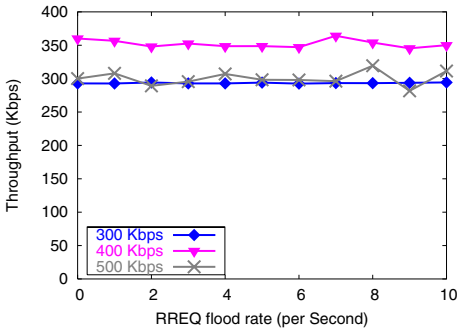

Fig. 10. Alternate view of the benefit of statistical rate control mechanism. Offered load is kept constant while the rate of bogus route discoveries is varied.

simulations indicate that, very few RREQs are dropped for loads below saturation. At and beyond saturation, however, the rate control mechanism drops more and more RREQs from highly active nodes. To verify this we examined the TTLs of RREQs that were dropped after 1 hop. This data given in Table II clearly shows that the nodes that are sending too many RREQs above the threshold are the ones that need to establish long routes. In saturation, repairing a broken route takes a lot of time. In the mean time, these nodes send more and more route requests which further increases network congestion. With the rate control mechanism, this is reduced and the saved bandwidth is used for data packet transmissions. We believe with a more adaptive mechanism to drop RREQs, the small drop in throughput can be mitigated.

V. CONCLUSIONS

The route discovery based denial of service attacks cause severe drop in network performance for MANETs. While there have been several studies of routing attacks by malicious nodes, these studies assume that the malicious nodes behaves in an obvious way by advertising misinformation or dropping packets. The type of attack we investigated makes malicious nodes appear as normal nodes with frequent route discoveries. Over a short period of time, route requests from normal and malicious nodes are not easy to distinguish. But

TABLE II

TTLS OF RREQS DROPPED BY THE RATE CONTROL MECHANISM IN A NORMAL NETWORK WITH NO MALICIOUS NODES.

Offered Load Kbps	% RREQs Dropped	Average TTL
100	14.7	3.3
200	4.6	3.3
300	2.4	3.4
400	5.2	3.8
500	5.3	4.1
600	12.3	4.6
700	11.2	4.7

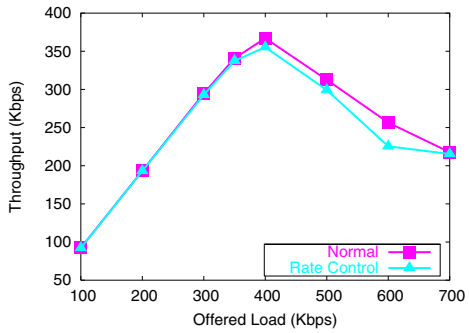

Fig. 11. Impact of statistical rate control in a normal network with no malicious nodes.

over a long period of time, malicious nodes can be easily detected since normal nodes send a high rate of RREQs for a short duration, but malicious nodes do so at all times. Based on this observation, we have proposed a simple statistical packet dropping mechanism that curbs attacks from malicious nodes effectively without hurting normal nodes.

In future, we will investigate the proposed mechanism for TCP traffic and for the case of multiple attackers.

REFERENCES

[1] R. Bagrodia. Global mobile information system simulation library.

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

[3] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). In *IETF, Internet Draft, IEEE RFC 3626, http://www.ietf.org/rfc/rfc3626.txt*, October 2003.

[4] S. R. Das, R. Castaneda, J. Yan, and R. Sengupta. Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In *Proceedings of the 7th Int. Conf. on Computer Communications and Networks (ICCCN)*, pages 153–161, Lafayette, LA, October, 1998.

[5] S. Desilva and R. Boppana. On the impact of noise sensitivity on performance in 802.11 based ad hoc networks. In *To be appeared in Proceeding of International Conference on Communications (ICC04) (http://www.cs.utsa.edu/ sdesilva/publications/icc04.pdf)*, Paris, France, June, 2004.

[6] S. Desilva and R. V. Boppana. Sustaining performance under traffic overload. In *2004 International Workshop on Mobile and Wireless Ad Hoc Networking*, Las Vegas, Nevada, June, 2004.

[7] T. Dyer and R.V. Boppana. A comparison of tcp performance over three routing protocols for mobile ad hoc networks. In *Proceedings of ACM Mobihoc*, October 2001.

[8] Yih-Chun Hu, Adrian Perrig, and David Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *ACM Workshop on Wireless Security*, 2003.

[9] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[10] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, 2000.

[11] Ze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8, March 2003.

[12] Peng Ning and Kun Sun. How to misuse aodv: A case study of insider attacks against mobile ad-hoc routing protocols. In *4th Annual IEEE Information Assurance Workshop*, pages 60–67, West Point, June 2003.

[13] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF). In *IETF, Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-tbrpf-11.txt*, April 13, 2004.

[14] Wei Peng and Xi-Cheng Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 129–130. IEEE Press, 2000.

[15] C. E. Perkins, E. M. Moyer, and S. R. Das. Secure ad hoc on-demand distance vector (SAODV) routing. In *Mobile Ad Hoc Networking Working Group Internet Draft, http://www.cs.ucsb.edu/ ebelding/txt/saodv.txt*, August 2001.

[16] C. E. Perkins, E. M. Moyer, and S. R. Das. Ad hoc on demand distance vector (AODV) routing. In *IETF, Internet Draft, IEEE RFC 3561, http://www.ietf.org/rfc/rfc3561.txt?number=3561*, July 2003.

[17] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proceeding of the IEEE workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[18] Giovanni Resta and Paolo Santi. An analysis of the node spatial distribution of the random waypoint mobility model for ad hoc networks. In *Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 44–50. ACM Press, 2002.

[19] Bharat Bhargava Weichao Wang, Yi Lu. On vulnerability and protection of ad hoc on-demand distance vector prototol. In *International Conference on Telecommunication*, France, Paris, 2003.

[20] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 194–205. ACM Press, 2002.

[21] Gary R. Wright and W. Richard Stevens. *TCP/IP Illustrated*, volume 2 of *The Implementation*, chapter 25, pages 817–849. Addison-Wesley, April A994.