

Inherent Robustness of Reactive Routing Protocols against Selfish Attacks

Asad Amir Pirzada and Chris McDonald
School of Computer Science & Software Engineering
The University of Western Australia
35 Stirling Highway, Crawley, Western Australia, 6009
Email: {pirzada, chris}@csse.uwa.edu.au

Abstract— Mobile ad-hoc wireless networks generally comprise nodes having meagre computation and communication resources. To perform multi-hop communication in a dynamic topology, these nodes execute special routing protocols. Each node performs the function of a mobile router and directs packets to other nodes in the network. For accurate functioning of the network it is imperative that all nodes execute these routing protocols in a benevolent manner. However, as ad-hoc networks are usually established in a physically insecure wireless environment, the network memberships are violated allowing malicious nodes to also participate in the network. These nodes can launch an array of attacks against different network services including the routing process. In this paper, we evaluate the performance of three well known reactive routing protocols, in a network with varying numbers of malicious nodes. With the help of exhaustive simulations, we demonstrate that the performance of the three protocols varies significantly even under similar attack, traffic and mobility conditions.

Keywords: Ad-hoc, Network, Routing Protocols, Security, Attacks.

I. INTRODUCTION

Ad-hoc networks are primarily used in improvised environments where communication can be rapidly established between the nodes without requiring any fixed infrastructure. These networks emerge in a spontaneous manner when a node is within the transmission range of one or more nodes. As the topology of the network is fundamentally fluid at all times, the routing protocols act as the binding force ensuring connectivity between the nodes. These routing protocols can be categorised into two types [1]: Proactive and Reactive.

Proactive routing protocols exchange route or link tables on a periodic basis and hence maintain up to date working routes at all instants of time. However, in doing so, these protocols consume additional battery power as a result of regular exchange of connectivity information. In contrast, Reactive routing protocols only discover routes in the network when no prior route to a particular node is known. These route discoveries are initiated in an on-demand manner and so help in conserving battery power, which is considered a vital resource in ad-hoc networks. Information retrieved through these route discoveries is retained and exploited to its utmost with an aim to minimise subsequent route discoveries.

In wired networks, inter-router protocols essentially have information about their neighbouring routers. These routers perform the routing functions in a rather static manner with

route changeover, being a rather rare phenomenon, performed only upon link severing, load balancing or quality of service requirements. On the contrary, the node neighbourhood in an ad-hoc network changes rapidly along with the link connectivity depending upon the wireless range and mobility of the network. Consequently, the routing information also needs to be precisely updated at a similar pace taking into account the present neighbourhood information. Thus the accurate functioning of an ad-hoc network is directly bound with the sincere and skilful execution of these routing protocols by the participating nodes.

However, the physical vulnerability of the nodes coupled with the readily accessible wireless medium makes enforcement of rigid node memberships extremely difficult. Thus malevolent nodes manage to join these networks with an intention to divert, disturb or disrupt the network flow. These nodes can be categorised into two main types: malicious and compromised. Malicious nodes are those that launch selfish, fabrication, modification or impersonation attacks [2] against the network. Among these attacks, the most commonly observed are the selfish attacks in which malicious nodes intend taking advantage of the network services but are unwilling to sacrifice their own computing or battery resources for the benefit of the network [3]. Compromised nodes are basically benevolent nodes, which have been physically or logically captured to perform in a malicious manner [4].

Extensive research has been carried out to secure ad-hoc networks using cryptographic tools [5]. However, the use of these tools generally imposes a number of prerequisites during both the establishment and operation phases of the network. The customary deployment of these tools also requires a centralised or distributed trusted third party in the network creating a somewhat managed rather than a pure ad-hoc network [6].

AODV, DSR and TORA are three well known reactive routing protocols which are undergoing wide-ranging active research. These protocols have been developed for networks where all nodes can faithfully execute them in a cooperative manner. However, in real life such an altruistic stance is difficult to achieve and so these protocols are more often executed by nodes that divert from the basic requirements of participation.

This paper is the first to provide a realistic comparative

analysis of these three protocols in an attacked network. In this paper, we evaluate the performance of these reactive routing protocols in a pure ad-hoc network which is under selfish attacks. We imitate these attacks in the network and monitor the inherent robustness of the three routing protocols against them. With the help of extensive simulations we present results showing the performance of each protocol in an attacked network under varying traffic loads.

We describe the working of the three routing protocols in Section II. In Section III we first explain the simulation environment, attack pattern and the monitored metrics and then discuss the simulation results. The rest of this paper consists of an analysis in Section IV with concluding remarks in Section V.

II. REACTIVE ROUTING PROTOCOLS

A. AODV

Ad-hoc On-Demand Distance Vector Routing Protocol (AODV) [7] is inherently a distance vector routing protocol that has been optimised for ad-hoc wireless networks. It is an on-demand protocol as it finds the routes only when required and is hence also reactive in nature. AODV borrows basic route establishment and maintenance mechanisms from the Dynamic Source Routing (DSR) protocol [8] and hop-to-hop routing vectors from the Destination-Sequenced Distance-Vector (DSDV) routing protocol [9]. To avoid the problem of routing loops, AODV makes extensive use of sequence numbers in control packets.

When a source node intends communicating with a destination node whose route is not known, it broadcasts a ROUTE REQUEST packet. Each ROUTE REQUEST packet contains an ID, source and the destination node IP addresses and sequence numbers together with a hop count and control flags. The ID field uniquely identifies the ROUTE REQUEST packet; the sequence numbers indicate the freshness of control packets and the hop-count maintains the number of nodes between the source and the destination. Each recipient of the ROUTE REQUEST packet that has not seen the Source IP and ID pair or doesn't maintain a fresher (with larger sequence number) route to the destination rebroadcasts the same packet after incrementing the hop-count. Such intermediate nodes also create and preserve a REVERSE ROUTE to the source node for a certain interval of time. When the ROUTE REQUEST packet reaches the destination node or any node that has a fresher route to the destination a ROUTE REPLY packet is generated and unicast back to the source of the ROUTE REQUEST packet. Each ROUTE REPLY packet contains the destination sequence number, the source and the destination IP addresses, route lifetime together with a hop count and control flags. Each intermediate node that receives the ROUTE REPLY packet, increments the hop-count, establishes a FORWARD ROUTE to the source of the packet and transmits the packet on the REVERSE ROUTE. For preserving connectivity information, each node executing AODV makes use of periodic HELLO messages to detect link breakages to nodes that it considers as its immediate neighbours. In case a link break is detected

for a next hop of an active route a ROUTE ERROR message is sent to its active neighbours that were using that particular route.

B. DSR

The Dynamic Source Routing (DSR) protocol [8] is a reactive routing protocol. As the name suggests it uses IP source routing. All data packets that are sent using the DSR protocol contain the complete list of nodes that the packet has to traverse. During route discovery, the source node broadcasts a ROUTE REQUEST packet with a unique identification number. The ROUTE REQUEST packet contains the address of the target node to which a route is desired. All nodes that have no information regarding the target node or have not seen the same ROUTE REQUEST packet before, append their IP addresses to the ROUTE REQUEST packet and re-broadcast it. In order to control the spread of the ROUTE REQUEST packets, the broadcast is done in a non-propagating manner with the IP TTL field being incremented in each route discovery. The ROUTE REQUEST packets keep on spreading until they reach the target node or any other node that has a route to the target node. The recipient node creates a ROUTE REPLY packet, which contains the complete list of nodes that the ROUTE REQUEST packet had traversed. Based upon implementation, the target node may respond to one or more incoming ROUTE REQUEST packets. Similarly, the source node may accept one or more ROUTE REPLY packets for a single target node. The selection of the ROUTE REPLY can be made both on minimal hop count or latency. In this paper, we have used a multi-path version [10] of the DSR protocol in which each ROUTE REQUEST packet received by the destination is responded to by an independent ROUTE REPLY packet. For optimisation reasons, nodes maintain a PATH CACHE or a LINK CACHE scheme [11]. All nodes either forwarding or overhearing data and control packets, add all useful information to their respective route cache. This information is used to limit the spread of control packets for subsequent route discoveries.

C. TORA

The Temporally Ordered Routing Algorithm (TORA) [12] is a distributed routing protocol for multi-hop networks. The unique feature of this protocol is that it endeavours to localize the spread of routing control packets. The protocol is basically an optimised hybrid of the Gafni Bertsekas (GB) protocol [13] and the Lightweight Mobile Routing (LMR) protocol [14]. It guarantees loop freedom, multiple routes and minimal communication overhead even in highly dynamic environments. The protocol attempts to minimise routing discovery overhead and in doing so prefers instant routes over optimal routes. The protocol supports source-initiated on-demand routing for networks with a high rate of mobility as well as destination oriented proactive routing for networks with lesser mobility. TORA maintains state on a per-destination basis and runs a logically separate instance of the algorithm for each destination. TORA assigns directional

heights to links so as to direct the flow of traffic from a higher source node to a lower destination. The significance of these heights, which are assigned based on the direction of a link towards the destination, is that a node may only forward packets downstream but not upstream, i.e. to another node that has a higher, undefined or unknown height. The height is represented by a quintuple $(\tau, oid, r, \delta, i)$ where the first three values represent a reference level and the last two represent the change with respect to the reference level. Each time a node loses its downstream link due to a link failure, a new reference level is computed using either a partial or full link reversal mechanism. The values in the height quintuple indicate the following:

- τ Logical time of a link failure
- oid Unique ID of the router defining the reference level
- r Reflection indicator bit
- δ Propagation ordering parameter
- i Unique ID of the router

In the on-demand mode, TORA algorithm performs four routing functions: Route Creation, Route Maintenance, Route Erasure and Route Optimisation. To accomplish these functions it uses four distinct control packets: Query (QRY), Update (UPD), Clear (CLR) and Optimisation (OPT). During route discovery, a source node requiring a route to a destination, broadcasts a QRY packet containing the destination address. The QRY packet is propagated through the network until it reaches the destination or any intermediate node possessing a route to the intended destination. The recipient of the QRY packet broadcasts an UPD packet that lists its height with respect to the destination. If the destination itself replies to a QRY packet it sets the height to zero in the UPD packet. Each node that receives the UPD packet sets its own height greater than that in the UPD packet. This results in creation of a directed acyclic graph (DAG) with all links pointing in the direction of the destination as the root. In the proactive mode, routes are created using the OPT packet that is sent out by the destination. The OPT packet, which is similar to the UPD packet, also consists of a sequence number for duplication avoidance. Each recipient nodes adjusts its height data structure and sends out a OPT packet to neighbouring nodes.

As each node in TORA maintains multiple DAGs to the destination so in any network with an average n number of nodes each with $\frac{n}{2}$ downstream neighbours, a node could still effectively communicate with the destination node upon link failure of $\frac{n}{2} - 1$ nodes. However, to sustain this redundancy, each node maintains a height data structure, link status along with a number of state and auxiliary variables for each destination node.

TORA is not a standalone routing protocol but requires the services of the Internet MANET Encapsulation Protocol (IMEP) [15]. IMEP has been designed as a network layer protocol that provides link status, reliable in-order delivery of control packets, neighbour connectivity information, address resolution and other services to Upper Layer Protocols (ULP).

IMEP endeavours to reduce overhead by aggregating a number of control packets into blocks. These blocks are transmitted with a sequence number and a list of nodes who have not yet acknowledged the receipt of the block. All such nodes, upon receipt of the same block, acknowledge the receipt. However, if after a certain number of retries the block is still not acknowledged, the link to that node is considered down and this information is conveyed to the ULP.

IMEP uses BEACON packets to ascertain the connection status between adjacent nodes. Each node periodically broadcasts a BEACON that contains a Router Identification number. In response to the BEACON, every recipient node broadcasts an ECHO packet that contains its IP address. These packets ensure that all nodes maintain local neighbourhood connectivity information at all times. IMEP declares a link to be broken if no BEACON's are exchanged within the maximum beacon interval [16].

III. SIMULATION

A. Setup

The NS-2 [17] simulator was used to evaluate the performance of the three routing protocols under identical attack conditions. The simulation parameters are listed in Table 1.

TABLE I
TABLE 1: SIMULATION PARAMETERS

Examined Protocol	AODV, DSR and TORA
Simulation time	900 seconds
Simulation area	1000 x 1000 m
Number of nodes	50
Propagation Model	Free Space
Transmission range	250 m
Transmission Power	281.8 mW
Reception Power	281.8 mW
Movement model	Random waypoint
Maximum speed	20 m/s
Pause time	10 seconds
Traffic type	CBR (UDP)
Maximum Connections	10, 20 & 30
Payload size	512 bytes
Packet rate	4 pkt/sec
Maximum malicious nodes	20

B. Mobility Model

We implement the random way point movement model for the simulation, in which a node first waits for the pause interval and then moves to a randomly chosen position with a velocity chosen between 0 m/s to the maximum speed, waits there for the pause time, and then moves on to another random position.

C. Communications Model

We use the IEEE standard 802.11 Distributed Coordination Function (DCF) [18] as the MAC layer for the three routing protocols. All ROUTE REQUEST and QRY packets are broadcasted using the un-slotted Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA). In CSMA/CA each broadcasting node waits for a vacant channel by sensing the medium. If the channel is vacant, it makes the transmission.

In the event of a collision, the colliding stations defer using the Ethernet binary exponential back off algorithm. To unicast packets, the node first reserves the channel by transmitting a short Ready-to-Send (RTS) frame. The intended recipient node, in response, sends a Clear-to-Send frame to the RTS sender. All nodes overhearing the RTS or CTS frames desist from transmitting for the Network Allocation Vector (NAV) interval. Upon receipt of the CTS, the packet is transmitted which is acknowledged by the recipient [19].

D. Attack Pattern

Malicious nodes simulate the following two types of selfish attacks:

1) *Black Hole Attack*: In this attack the malicious node drops all data packets, which it is supposed to forward. However, it participates devotedly in the route discovery process, which is initiated by other nodes so as to remain on the path of the data connections.

2) *Grey Hole Attack*: The grey hole attack is similar to the black hole attack except that the malicious node also selectively forwards data packets at random intervals.

E. Legitimate Packet drop

In addition to malicious packet drop, data packets can also be dropped by any node due to the following reasons:

1) *MAC Layer Collisions*: All three protocols do not guarantee packet delivery and so data packets are not buffered for retransmission. In the event of a collision involving a data packet, the packet is simply considered lost. The responsibility of retransmission of the packet is left to the higher layers in the protocol stack.

2) *Saturation of Interface Queues*: TORA, AODV and DSR implement Network Interface Queues (IFQ) to buffer packets, which are ready to be transmitted and are received by the network protocol stack. These IFQ generally limit the maximum number of packets that can be held in them and may also implement a maximum timeout policy for packets in the IFQ. As a result, any packet awaiting a route in the IFQ for an extended period, may simply be discarded without any notification.

The legitimate packet drops are influenced by the mobility pattern of the network and accordingly influence the different performance metrics of the network. This is confirmed by the fact that the throughput of the three protocols always remains lower than 100% even when no malicious nodes are present in the network. However, the ratio of legitimate packet drop to deliberate drop is negligible, as will be shown in the results.

F. Metrics

To evaluate the performance of the protocols, we use the following metrics:

1) *Throughput*: It is the ratio between the number of data packets received by the application layer of destination nodes to the number of packets sent by the application layer of source nodes.

2) *Routing Packet Overhead*: This is the ratio between the total number of control packets generated (excluding HELLO and BEACON packets) to the total number of data packets received during the simulation time.

3) *Average Latency*: Gives the mean time (in seconds) taken by the data packets to reach their respective destinations.

4) *Path Optimality*: It is the ratio between the number of hops in the optimal path to the number of hops in the path taken by the data packets.

5) *Energy Consumption*: It is the amount of energy (in Joules) consumed per node during the simulation time.

G. Results and Discussion

Each simulation is carried out with a different mobility and connection pattern and the results are obtained by ensemble averaging [20] over 100 runs. The test was carried out under multiple traffic loads with the number of connections set to 10, 20 and 30. Accordingly, AODV with ten sources is represented by AODV-10, DSR with twenty sources by DSR-20 and TORA with thirty sources by TORA-30 respectively.

Fig. 1 depicts the performance results for the AODV, DSR and TORA protocols in the presence of malicious nodes. The results indicate that the throughput of all three protocols rapidly drops with the increase in the number of malicious nodes. The rate of this drop is the highest for AODV which degrades from 97% with no malicious nodes to about 33% with 40% malicious nodes. The throughput drops in DSR from 95% to 32% for a similar increase in the number of malicious nodes. The TORA-10 protocol degrades from an 85% throughput to about 40%. However, the increase in the number of sources causes TORA-20 and TORA-30 to undergo a congestive collapse [16]. This is essentially due to the positive feedback loop created in TORA/IMEP due to the increased number of MAC layer collisions. These collisions, incorrectly make IMEP believe that the links to adjacent nodes are severed. In response TORA generates more UPD packets, which closes a serviceable link that is temporarily congested. This leads to generation of further QRY packets to find alternate routes despite the availability of working routes. This increased control packet overhead, essentially closes the feedback loop causing further congestion. On the other hand, the malicious nodes in the network, which drop the network traffic, aid TORA in recovering from this phenomenon by reducing the traffic load and thereby the number of MAC layer collisions. This indirectly improves the performance of the TORA-20 and TORA-30 protocol from 15% to approximately 38% with 40% malicious nodes.

The routing overhead of all three protocols remains significantly constant with the increase in the number of malicious nodes. AODV, on average, generates one control packet for each received data packet, DSR one for 10 received data packets while TORA-10 generates one control packet for every four received data packets. The increased control packet overhead in AODV and TORA is primarily due to their route discovery mechanism that requires the ROUTE REQUEST or UPD packet to be broadcasted over the network. Whereas

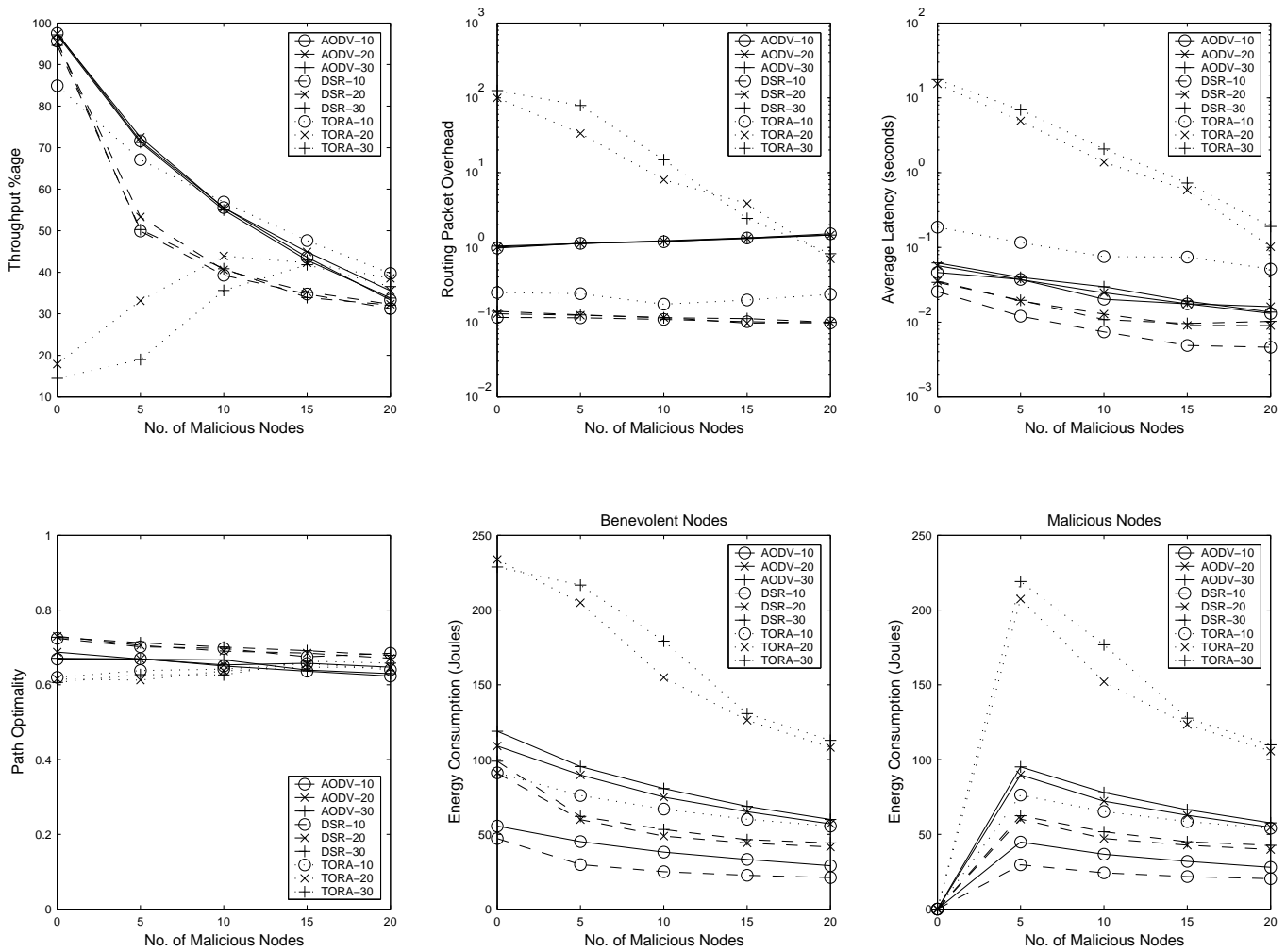


Fig. 1. Simulation with varying number of Malicious Nodes

the DSR protocol makes effective use of its inherent caching strategy to limit the routing overhead. The control packet overhead of TORA-20 and TORA-30 remains exponentially high when no malicious nodes are present and drops when the malicious nodes in the network inadvertently support the protocol by lowering the amount of traffic in the network.

The path optimality of the network with all three protocols remains steady with the increase in the number of malicious nodes. The data packets, which finally do reach their intended destinations in the presence of malicious nodes, have traversed the path that contains no black hole but may or may not contain a grey hole. This path essentially represents the least hop path chosen by the sending node at that particular instant and so also has lower latency. TORA-20 and TORA-30 depict increased latency due to the creation of the positive feedback loop when the network contains a low number of malicious nodes. However, as the data traffic gets truncated by the malicious nodes the latency with TORA also improves. The DSR protocol, under all traffic loads makes effective use of available cache information and thereby offers the least latency in the network despite the increase in the number of malicious

nodes.

The energy consumption of the nodes is primarily dependent upon the throughput and the control packet overhead. Nodes executing the DSR protocol consume less energy as compared to those executing the AODV and TORA protocol. The increase in the number of malicious nodes directly reduces the throughput of the network which in turn lowers the energy consumption by the nodes. The malicious nodes are able to save upon a few Joules by not implementing the forwarding mechanism as compared to the benevolent nodes.

IV. ANALYSIS

The simulation results indicate that the TORA protocol performs exceptionally well in the presence of malicious nodes. However, due to the inherent feedback loop problem of the TORA protocol it fails to sustain the same performance under higher traffic loads. The multi-path feature of the TORA protocol permits intermediary nodes to make localized routing decisions during breakage of existing links. These decisions help to improve the overall throughput of the network. TORA has relatively higher packet overhead due to its IMEP reliable

delivery mechanism and thus induces higher energy consumption for both benevolent and malicious nodes. Malicious nodes that do not accurately perform the packet forwarding function are able to conserve some of their energy. The path optimality of TORA remains significantly constant even with increased number of malicious nodes. This is due to the fact that the packets that do reach their intended destinations have traversed the least hop path at that particular instant.

Nodes executing the DSR protocol consume the least energy due to the minimal routing overhead generated by the DSR protocol. The DSR protocol makes maximum use of its caching scheme to limit the number of search requests for known nodes. However, the overall throughput of the DSR protocol in the presence of malicious nodes, remains lower than that of the TORA and AODV protocols. This is attributed to the fact that a malicious node present on a link in the LINK CACHE scheme of the DSR protocol will be used until the time it stays on the least hop path and the link remains serviceable. Thus a sending node will continue sending packets on the path containing a malicious node until the link expires. Only upon the expiry of that link, will an alternate route be found to the same destination, either from the cache or through a new route discovery. Whereas in TORA and AODV, the point to point routing process inadvertently helps to use both benevolent and malicious nodes in the data connections, and so no one type of node remains involved in a route for an extended duration.

AODV exhibits an improved throughput over the other two protocols under different traffic loads with varying number of malicious nodes. AODV and TORA, functionally operate in a similar pattern where the intermediary nodes pass the data packets to the next hop on the path to the destination. However, the intermediary nodes in TORA take advantage of its multipath feature and are able to convey packets even upon the link severing of the primary link. On the other hand, upon severing of a link, a node executing AODV has to drop the packet and propagate a ROUTE ERROR packet. These ROUTE ERROR packets require the source node to initiate further route discoveries increasing the overall packet overhead.

V. CONCLUSION

Ad-hoc networks are generally established in improvised environments with minimal or no restrictions on node memberships. To sustain connectivity in a dynamic topology, all participating nodes are expected to sincerely execute the routing protocols. However, there may be circumstances in which malevolent nodes may also participate and attack the network. In order to determine the precise impact of such nodes, in this paper, we have examined the performance of three common reactive routing protocols in an attacked network. We simulated up to 40% malicious nodes in the network and monitored the routing protocols under varying traffic conditions. The results from the simulations indicate that the performance of these protocols varies significantly under similar attack conditions. TORA, at lower traffic loads, performs better than the other two protocols in the presence of malicious nodes. AODV performs second best but surpasses TORA at higher traffic

loads. DSR, although provides lower throughput as compared to either TORA or AODV, generates the least routing overhead, has the lowest latency and consumes minimal energy.

ACKNOWLEDGEMENTS

This work was supported by the Australian International Postgraduate Research Scholarship and the University of Western Australia Postgraduate Award.

REFERENCES

- [1] E. M. Royer and C. K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications Magazine*, vol. 6, no. 2, pp. 46–55, 1999.
- [2] B. Dahill, B. N. Levine, E. Royer, and C. Shields, "A secure routing protocol for ad hoc networks," *Proceedings of the International Conference on Network Protocols (ICNP)*, pp. 78–87, 2002.
- [3] P. Michiardi and R. Molva, "Simulation-based analysis of security exposures in mobile ad hoc networks," *Proceedings of the European Wireless Conference*, 2002.
- [4] S. Carter and A. Yasinsac, "Secure position aided ad hoc routing protocol," *Proceedings of the IASTED Conference on Communications and Computer Networks (CCN)*, pp. 329–334, 2002.
- [5] A. A. Pirzada and C. McDonald, "Secure routing protocols for mobile ad-hoc wireless networks," in *Advanced Wired and Wireless Networks*, T. A. Wysocki, A. Dadej, and B. J. Wysocki, Eds. Springer, 2004.
- [6] —, "Establishing trust in pure ad-hoc networks," *Proceedings of the 27th Australasian Computer Science Conference (ACSC)*, vol. 26, no. 1, pp. 47–54, 2004.
- [7] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," *IETF RFC 3591*, 2003.
- [8] D. B. Johnson, D. A. Maltz, and Y. Hu, "The dynamic source routing protocol for mobile ad hoc networks (dsr)," *IETF MANET, Internet Draft (work in progress)*, 2003.
- [9] C. E. Perkins and P. Bhagwat, "Dynamic destination-sequenced distance-vector routing (dsv) for mobile computers," *Proceedings of the SIGCOMM Conference on Communications, Architectures, Protocols and Applications*, pp. 234–244, 1994.
- [10] A. Nasipuri and S. Das, "On-demand multipath routing for mobile ad hoc networks," *Proceedings of the Eight International Conference on Computer Communications and Networks*, pp. 64–70, 1999.
- [11] Y. C. Hu and D. B. Johnson, "Caching strategies in on-demand routing protocols for wireless ad hoc networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 231–242, 2000.
- [12] V. Park and S. Corson, "Temporally ordered routing algorithm (tora) version 1 functional specification," *IETF MANET, Internet Draft (work in progress)*, 2001.
- [13] E. Gafni and D. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology," *IEEE Transactions on Communications*, vol. 29, no. 1, pp. 11–18, 1981.
- [14] M. S. Corson and A. Ephremides, "Lightweight mobile routing protocol (lmr), a distributed routing algorithm for mobile wireless networks," *Wireless Networks*, 1995.
- [15] S. Corson, S. Papademetriou, P. Papadopoulos, V. Park, and A. Qayyum, "Internet manet encapsulation protocol (imep) specification," *IETF MANET, Internet Draft (work in progress)*, 1999.
- [16] 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," *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 85–97, 1998.
- [17] NS, "The network simulator," <http://www.isi.edu/nsnam/ns/>, 1989.
- [18] IEEE-Standard, "Wireless lan medium access control (mac) and physical layer (phy) specifications 802.11," 1997.
- [19] A. S. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall, 2002.
- [20] W. H. Yuen and R. D. Yates, "Inter-relationships of performance metrics and system parameters in mobile ad hoc networks," *Proceedings of the IEEE MILCOM*, vol. 1, pp. 519–524, 2002.