Mitigating routing vulnerabilities in *ad hoc* networks using reputations

Prashant Dewan*

Corporate Technology Group
Intel Corporation
2111 N.E. 25th Avenue
Hillsboro, OR 97124–5961, USA
E-mail: prashant.dewan@intel.com
*Corresponding author

Partha Dasgupta

Department of Computer Science
Arizona State University
Room 428
Brickyard (on Mill Avenue)
Tempe, AZ 85287–8809, USA
E-mail: partha@asu.edu

Amiya Bhattacharya

Department of Computer Science
New Mexico State University
Las Cruces, NM 88003, USA
E-mail: amiya@nmsu.edu

Abstract: Nodes in mobile *ad hoc* networks have limited transmission ranges that necessitate multihop communication. Hence the nodes expect their neighbours to relay the packets meant for nodes out of the transmission range of the source. *Ad hoc* networks are based on the fundamental assumption that if a node promises to relay a packet, it will relay the packet and will not cheat. This assumption becomes invalid when the nodes in the network have contradictory goals. As a result, routing protocols for *ad hoc* networks become vulnerable to rogue nodes. The reputations of the intermediate nodes, based on their past history of relaying packets, can be used by their neighbours to ensure that the packet will be relayed by the intermediate nodes. This paper introduces a reputation scheme for *ad hoc* networks that can motivate the intermediate nodes to relay packets. The source performs a *route discovery* (using *Ad hoc* on Demand Distance Vector Routing Protocol (AODV)) and finds a set of routes to the destination. Instead of choosing the shortest route to the destination, the source node chooses a path whose next hop node has the highest reputation. This policy, when used recursively, in the presence of 40% rogue nodes, improves the throughput of the system to 65%, from the 22% throughput provided by AODV with same number of rogue nodes. This improvement is obtained at the cost of a higher number of route discoveries with a minimal increase in the average hop length.

Keywords: network security; ad hoc networks; reputations.

Reference to this paper should be made as follows: Dewan, P., Dasgupta, P. and Bhattacharya, A. (2009) 'Mitigating routing vulnerabilities in *ad hoc* networks using reputations', *Int. J. Information and Computer Security*, Vol. 3, No. 2, pp.150–172.

Biographical notes: Prashant Dewan is a Research Scientist at Intel Labs, USA. His research interests are network and platform security, virtualisation and decentralised networks. He has a PhD in Computer Science from Arizona State University and has been working at Intel since 2004. Dewan has published in various academic journals and has more than 20 patents pending at the USPTO.

Partha Dasgupta is on the faculty of Arizona State University, USA. His core areas of expertise are in operating systems, computer security and distributed computing. His current research focus is the uses of cryptography and secure software systems to provide security and dependability of consumer computing. Dr. Dasgupta joined ASU in 1991 and has held faculty positions at Georgia Tech and New York University. His research funding has primarily been from NSF and DARPA with smaller grants from Intel, Microsoft and the Consortium for Embedded Systems. He has 20 years of experience with operating systems and 8 years of experience with security systems. He has a PhD in Computer Science from Stony Brook University.

Amiya Bhattacharya received his Bachelor's degree in Computer Science and Engineering in 1987 from the Indian Institute of Technology at Kharagpur, India, and his Master's degree in Computer Science in 1991 from the University of California, San Diego, USA. He obtained his PhD in Computer Science and Engineering in 2002 from the University of Texas at Arlington, where he was a recipient of the Texas Telecommunication Engineering Consortium Fellowship and the 2002 Outstanding Doctoral Research Award. Prior to joining NMSU, he conducted research at Nokia Research Center at Irving and at the Arizona State University. His primary research interests are mobile computing and communication systems, wireless networks, internet protocols, performance analysis, modelling and simulation.

1 Introduction

Security is an important issue in wireless *ad hoc* networks. The substitution of wired channels with wireless channels, their deployment in military and civilian areas, battlefields, *etc*., make them an easy target of attack by an adversary. Due to the semi-autonomous nature of the nodes comprising the network, the availability of the network, confidentiality and integrity of information in the communication channels, authentication and non-repudiation of the network nodes is an important challenge. In addition, the network has to be protected from an adversary who implants nodes in the network for malicious ends.

Secure routing in *ad hoc* networks is challenging as the nodes are dependent on each other for routing. Most mobile *ad hoc* networks are designed as self-configuring, adaptive networks, which can be deployed in areas deprived of any existing network infrastructure. Due to the limited transmission range of a node in an *ad hoc* network, it

has to rely on the neighbouring nodes in the network to route the packet to the packet's destination node. The routing protocols used in the current generation of mobile *ad hoc* networks, like *Dynamic Source Routing* (DSR) (Sanzgiri *et al*., 2002), and *Ad hoc On Demand Distance Vector Routing Protocol* (AODV) (Perkins and Royer, 1999), are based on the principle that all nodes will cooperate with each other. Certain nodes in an *ad hoc* network might become antagonistic to other nodes and, hence refuse to cooperate with others. Besides, an *ad hoc* network consisting of semi-autonomous nodes owned by different entities might not share a common goal, and hence the nodes might not cooperate, even after promising to do so. Such nodes are termed as *rogue nodes.*

Our simulation results show that for an *ad hoc* network with static nodes, the network throughput[1] drops by more than half, when 40% of the nodes are malicious. This throughput further reduces, with an increase in the number of rogue nodes, or when the nodes become mobile. Depending on the location of the rogue nodes in the network and the network topology, some nodes experience worse throughput than others.

In this paper, the reputation of nodes in an *ad hoc* network is used to detect and subsequently circumvent the rogue nodes. The reputation of a node is a function of the number of data packets that have been relayed by the node in the past and is not based on any other attribute or activity of the node. The nodes achieve high reputation by correctly routing packets for other nodes. If a node fails to route the packet even after promising to do so, or routes a packet incorrectly, it gets a low reputation and hence is subsequently weeded out from the *ad hoc* network.

In the proposed reputation scheme, the source node finds a set of paths to the destination, by using the broadcast based route discovery method of the AODV protocol. The source selects the route (to the destination) offered by its neighbour with the highest reputation. Once the route to the destination is available, the source node sends the data packet to the first hop (neighbour with the highest reputation). Then the first hop forwards the packet to the next hop with the highest reputation and the process is repeated till the packet reaches its destination. The destination acknowledges the packet to the source via the same path in reverse. The source node updates its reputation table by incrementing the reputation of the first hop by 1. All the intermediate nodes in the route increment the reputation of their respective next hop by 1. If there is a malicious node in the route, the data packet does not reach its destination. As a result, the source does not receive any acknowledgement for the data packet before timeout. The source node reduces the reputation, for the first hop in the route, by 1. The intermediate nodes propagate this recommendation downstream, in the route up to the node that dropped the packet. In other words, all the nodes between the malicious node and the sender, including the malicious node, get their reputation decremented by 1 by their respective upstream previous hops. In due time, the reputation of the rogue goes below the threshold, T_r, and after that it is never used as an intermediate hop for any destination. The good nodes do not route packets originating at rogue nodes. As a result therefore the rogue nodes are weeded out of the network and cannot cause any further damage.

It is certainly possible that the packets enroute are not dropped maliciously by a node, but are dropped due to TX errors in the channel or due to congested channels. In such situations, the current node does not receive a MAC layer acknowledgement (802.11 ack), from the next hop. Hence the current node uses standard TCP congestion control algorithms: slow start, congestion avoidance, fast retransmit or fast recovery (Allman *et al*., 1999). In such a situation, no recommendations are generated and the reputation(s)

of the nodes do not change. *If the current node fails to get the acknowledgement even after multiple retires and is forced to use the next optimal route, then the current node and the subsequent nodes will reduce the reputation of the nodes downstream.*

The salient features of the proposed reputation system are:

- circumvention of rogue nodes
- injection of motivation to cooperate among nodes
- decentralised collection and storage of reputations
- subsequent increase in the average throughput of the *ad hoc* network.

In addition, the nodes in the network are able to quickly use the reputation information to make routing decisions without any significant impact on the routing performance.

The remainder of this paper is organised as follows. Section 2 summarises the work done on reputations and *ad hoc* networks; Section 3 presents the model of the *ad hoc* network used for the reputation scheme. In addition, Section 3 provides the corresponding background information and explains the reputation-based scheme presented. Section 5 discusses in detail the simulation environment and parameters and presents an analysis of the simulation results. Section 6 discusses the pros and cons of the proposed approach, while Section 7 provides the conclusion and discusses future work.

2 Related work

We outline the research done in using reputations for P2P networks. In addition, we present the security and reputation methodologies used in *ad hoc* networks, that are closely related to the research presented in this paper.

2.1 Peer-to-peer networks

Reputations in peer-to-peer networks have been investigated by many researchers looking to secure peer-to-peer networks and motivate the peers in the P2P network to abstain from cheating while punishing chronic cheaters (De Capitani di Vimercati Damiani and Paraboschi, 2003; Dewan and Dasgupta, 2004; Kamvar *et al*., 2003; Marti and Garcia-Molina, 2003; Marti *et al*., 2000; Wang, 2003). Simply said, the peers get recommendations for their actions or service provided and peers with low recommendations are declared to be *bad peers* by the peer-to-peer community and ostracised from the community. As *ad hoc* networks are based on the peer-to-peer model, research done for reputations in peer-to-peer networks has been borrowed for secure routing in *ad hoc* networks. Reputations in *ad hoc* networks are based on the same paradigm as that in peer-to-peer networks. The network nodes are evaluated and subsequently recommended on the basis of the *routing service* provided by the nodes. The highly reputed nodes route more packets and thereby raise their reputation. If the resources of the node are not sufficient, then it loses a fraction of its reputation and hence receives lower volume of traffic.

2.2 Ad hoc networks

Considerable volume of literature is available on securing *ad hoc* networks (Stajano and Anderson, 1999; Lidong Zhou, 1999; Ramanujan and Kudige, 2003). This body of literature can be divided into two parts. The first part consists of work performed for authentication, confidentiality, integrity, non repudiation of nodes to protect against various active and passive sniffing and spoofing, *worm hole*, *black hole*, *grey hole* and DoS attacks. The second part encompasses work done for secure routing by using incentive based mechanisms for motivating the nodes in the *ad hoc* network to be *good* nodes (Obreiter *et al.*, 2003; Bansal and Baker, 2003; Buchegger and Boudec, 2002a–b; Buttyán and Hubaux, 2000).

Stajano and Anderson (1999) have enumerated attacks on *ad hoc* networks. These attacks include but are not limited to radio jamming and denial of service attack leading to battery exhaustion. In addition, authentication of nodes by others in the network and establishment of trust association among nodes are a formidable challenge for *ad hoc* networks. Stajano *et al.* have proposed the use of *resurrecting duckling,* which is based on the principle that the entity that gets the control of the node first becomes the master of the node, till either the master releases the slave or an event occurs which forces the slave to change its master.

This paper belongs to the second category (Secure Routing). The other research done in the field of secure routing using reputations in *ad hoc* networks is as follows:

- Techniques for Intrusion-Resistant *Ad-hoc* Routing Algorithms (TIARA) (Ramanujan and Kudige, 2003) is a routing independent approach for countering Denial of Service Attacks on *ad hoc* networks. It restricts the attack traffic to the immediate neighbourhood of the adversary. The overlay routes are reconfigured to circumvent rogue nodes.

- Security-Aware *Ad-hoc* Routing (SAR) (Seyung *et al*., 2001) uses a novel metric for evaluation of the security of a route. It assumes pre-configured hierarchical trust relationships among nodes and predistribution of keys among them for establishment of secure associations. Nodes in SAR do not 'process' control packets (for route establishment), if they do not possess the necessary trust needed for processing the control packet, thereby resulting in a route in which all the nodes have a trust level at least equal to the threshold level specified in the packet.

- Secure Routing for Mobile *Ad hoc* Networks (SRP), proposed by Papadimitratos and Haas (2002) also uses pre-existing security association between the source and the destination and separates the malicious control packets from the genuine ones by performing cryptographic validation on the control traffic.

- Authenticated Routing for *Ad hoc* Network (ARAN), proposed by Sanzgiri *et al.* (2002), uses a trusted certificate server for node authentication. The nodes sign individual control packets in order to counter spoofed packets and attach their certificates to the packets signed by them for verification by subsequent hops. Either short lived certificates are used to do away with revocation or revocation lists signed by the CA are propagated in the network.

- Ariadne, proposed by Hu *et al.* (2002), foils spoofing attacks using Message Authentication Codes (MAC) and shared keys between the source and the destination. The target node authenticates the initiator using the shared key and the initiator can authenticate each entry in the path to the target node. The integrity of the route is protected using one-way hash functions.
- Inter Node Cooperation – Obreiter *et al.* (2003) classify inter-node noncooperation into *venial noncooperation* where the node just does not have sufficient resources to cooperate and *malicious noncooperation* where the nodes do not cooperate for malicious reasons. They further classify the remuneration schemes for motivation of nodes into *account based* and *reputation based* schemes. They say that account based schemes either need tamper proof hardware or a central banker. As most *ad hoc* networks have neither of the two, reputation based schemes are more appropriate for secure routing in *ad hoc* networks.
- Observation Based Cooperation Enforcement in *Ad hoc* Networks (OCEAN), proposed by Bansal and Baker (2003), is based on *localised reputation* values. The nodes in the *ad hoc* network evaluate the routing behaviour of the other nodes themselves and do not use third party reputations. The nodes snoop on the neighbours in order to evaluate their routing behaviour. The sender stores a checksum of the packet sent to the next hop and waits for the next hop to send a packet with the same checksum to the following node. If the neighbour does not send the packet with the same checksum, the sender gives a negative recommendation to the next hop; else it gives a positive recommendation. Every node maintains a neighbour rating and the absolute negative is higher than the absolute positive in order to check selective forwarding. The source node generates an *avoid list* and the subsequent nodes avoid the nodes in the list for carving a DSR based route from the source to the destination.
- The Terminodes protocol as proposed by Buttyán and Hubaux (2000) uses a digital money model called *Packet-Purse.* Each node has certain volume of *nuggets*, which are either obtained in exchange for real money or by routing packets for other nodes. In order to send a packet, the source node loads the packet with some nuggets and the intermediate node either takes out one nugget out of the packet or takes out all the nuggets and reloads the packet with fewer nuggets before sending it to the next hop. The nodes have the motivation to be rich because a node without nuggets will not be able to forward any packets and hence would be thrown out of the network. It implicitly assumes that the nodes have tamper-proof hardware and they cannot 'steal' nuggets from a packet.
- The CONFIDANT protocol, as proposed by Buchegger and Boudec (2002a–b) is also based on the snoop-thy-neighbour paradigm. Unlike OCEAN, CONFIDANT uses a *global reputation* scheme in which the nodes exchange the information about the reputation of other nodes using alarms. The nodes do give higher weightage to the reputation information gathered locally, as compared to information obtained from third parties. Sharing global information facilitates faster identification of rogue nodes. In addition, they use threshold cryptography in order to sift out liars who propagate false third party reputations.

- A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile *Ad hoc* Networks (CORE) is proposed by Michiardi and Molva in Michiardi (2002). CORE uses the global reputation mechanism with a twist. In CORE the *requester* requests the execution of a function from the provider and aims its *watchdog* on the *provider*. If the provider's reputation table shows that the requester is a good requester it executes the function and sends the result along with the list of other members who cooperated for the evaluation of the function, to the requester. Otherwise, the provider does not respond to the request. Once the requester obtains the result, it disarms the watchdog and the updates the reputations of the cooperating nodes in its local reputation table. If the provider cheats, the watchdog barks and the requester reduces the reputation of the provider. The twist is, although the requester accepts third party recommendations, it only considers positive recommendations in order to foil *bad mouthing*.[2]

- A Simple Cheat-proof, Credit-based System for Mobile *Ad hoc* Networks (SPRITE) is proposed by Zhong *et al.* (2002). Unlike Terminodes, SPRITE does not need any tamper-proof hardware but is based on a central trusted server named, *Credit Clearance Service* (CCS). The CCS accounts for all packets received, transmitted and dropped in the network. The source node 'loses' credit when it initiates sending a packet and the source node gains credit when it routes packets for other nodes. The credit is 'encashed' by submitting the packet receipts to the CCS off line. The CCS verifies the receipts with its packet info database and issues credits.

- Pathrater and Watchdog – Marti *et al*. (2000) have proposed a reputation scheme which uses a *watchdog* to snoop onto the neighbours and a *pathrater* to rate paths on the basis of presence of rogue nodes in them. In Marti *et al*. (2000), the nodes use a promiscuous mode to listen to the incoming and the outgoing traffic of their neighbouring nodes.

  Liu and Yang (2003) have shown in that if global reputations are used then the nodes will converge to a common reputation value for any specific node, irrespective of the number of liars in the system. The reputation information should be shared frequently enough among the nodes in the network for them to converge to common reputation values.

In all the existing systems (Buchegger and Boudec, 2002a; Michiardi, 2002; Kevin Lai *et al*., 2003; Zhong *et al*., 2002; Buttyán and Hubaux, 2000) the nodes have to watch their neighbourhood which not only necessitates promiscuous modes of operation, but also overloads the nodes with busy neighbourhoods. As a result, nodes can only ascertain if their neighbours forwarded the packets that they receive, but cannot ascertain if the forwarded packet reaches the destination or even the next hop. The proposed infrastructure does not use the 'watch-thy-neighbour' technique but relies on destination acknowledgement. The existing systems have been developed on DSR, while the proposed infrastructure has been implemented on AODV. A salient difference between their approach and the proposed approach is: while in Kevin Lai *et al*. (2003), a node selects the route to the destination, by considering the reputations of all the nodes in the route, the proposed scheme (Dewan *et al*., 2004) only considers the reputation of the next hop in the route.

3 Network model

The proposed network model comprises of semi-autonomous nodes capable of setting up 802.11 or similar wireless connections with other nodes within their transmission range. The nodes within the transmission range of a given node are the *neighbours* of the node. The sender of the packet receives an acknowledgement for each data packet or a set of data packets. The acknowledgement packet follows the reverse path of the original packet, towards the sender. The destination explicitly generates an acknowledgement to notify the sender that the data packet has reached its destination. The acknowledgement is cryptographically signed by the destination node in order to counter spoofed acknowledgements. For every packet that reaches its destination, the nodes in the respective route get a recommendation of +1 from the respective node that precedes the given node in the route. For every packet that is dropped en route to its destination, all the nodes in the route, before the malicious node that dropped the packet receive a recommendation of −1. Neither the source, nor the destination knows the specific node that dropped the packet.

Node identification is important issue as the reputation of the node is assigned to its identifier. A node should not be able to change its identifier and should not be able to spoof other nodes. PKI identifier based on identity certificate provided by a central authority fits the bill but PKI is computationally expensive and hence may not be a good solution for nodes with low volume of resources. Other approaches that can be used are, unique identifiers embedded in tamper proof hardware (a MAC address) or preconfigured symmetric trust relationships between nodes. For node identification, one solution does not fit all and hence will be dependent on the deployment scenarios.

In this paper we assume that the nodes in the network are identified by their public keys. Whenever a node initiates a route request it sends its certificate along with the RREQ packet. The public keys of the nodes are used to uniquely identify the nodes. The identities are *allotted* by a central Certificate Authority (CA).[5] The role of the CA finishes after the identities are allotted. The CA is needed to ensure that one node does not possess more than one identity, because if that happens the nodes would be able to raise their own reputations without routing packets for other nodes. As a result it would significantly reduce the utility of any reputation system.

If the reputation of a node falls below the *threshold reputation*, T_r, it is considered to be a malicious node. T_r is a global parameter which is configurable at the time of deployment. A malicious node will drop the data packets it receives for relaying. However it will not drop the control packets (RREQ, RREP), because if it drops the control packets, then it will no longer be a part of the network and hence will not be able to inflict any damage by dropping data packets. A rogue node can modify the control (request or response) packets in order to divert specific traffic towards itself or other nodes (*e.g.*, wormhole and black hole attacks). A *good node* (a node with reputation greater than T_r) will not forward a data packet to a malicious node and will try to find an alternative route to the destination. The nodes in the model are not a priori aware of which nodes are malicious. Initially, all the nodes hold the same reputation, *i.e.*, all the nodes are considered to be good nodes and none of the nodes is expected to be malicious.

A thorough explanation of the AODV protocol can be found in Perkins and Royer (1999). In AODV, once a sender has a packet for a destination, it checks its routing table to determine if it has a route to the destination. If it does not have a route (or has an

inactive route), it initiates a route discovery by broadcasting a RREQ. All the neighbours of the sender receive the RREQ. If any of the neighbours has a route to the destination, it sends a reply back to the sender in the form of a RREP. The sender updates its routing table with the route. If a neighbour does not have the route to the destination, it re-broadcasts the RREQ and the following cases can happen:

- If the RREQ reaches the destination, the neighbour sends a reply, RREP, back to the sender.
- If the RREQ reaches another node which has a route to the destination the neighbour again sends an RREP back to the sender.
- If the request times out another RREQ is sent out.

Once the sender has the route to the destination, it sends the data packet towards the destination, on known route. If the intermediate node is not able to forward the data packet to the next hop it sends a RERR to the sender, to inform all of its upstream nodes that might be interested in the broken route, or it performs a local repair of the broken part of the route.

4 Threat model and countermeasures

A malicious node can launch the following attacks in an *ad hoc* network, where the proposed reputation mechanism is not used:

1 *Incorrect routing information*: A malicious node might make a false claim to know the route to a destination and generate a RREP for a destination, for which it does not have a route. The motivation for this attack is to obtain a strategic position in the network such that a large volume of traffic is passed via the attacker. In order to execute this attack, the malicious node can claim to know the shortest path to the destination, or by modifying the sequence number in the AODV control packets. There can be three possible outcomes of this attack:

   - The packet does not reach its destination,
   - The packet reaches its destination without any modification but via a suboptimal path (*Wormhole Attack*).
   - The packet reaches its destination but is maliciously modified enroute.

   Reputation routing counters the scenarios 1a and 1c. After receiving the data packet for the corresponding destination, it will have to drop the data packet. The upstream node in the route will give a negative recommendation to the node. Once the reputation of the node falls below the threshold reputation, T_r, it will be considered as malicious and will eventually be ostracised. Worm-hole attack, scenario 1b, is mitigated using packet leashes (Gupte and Singhal, 2003) that specify the maximum TTL for a packet thereby putting an upper bound on the distance a packet can travel.

   A malicious node might not reveal that it knows the route to the destination. Although the node can save its resources (like energy, processing power, *etc*.) by doing this, it will not be able to inflict any damage to the network, as it will not be able to drop the data packets routed via other paths. In addition, the good nodes

assign lowest priority to the packets originating from low-reputation nodes. Hence rogue nodes will see a considerable increase in network latency, once all the nodes in the route to the packet destination assign lowermost routing priority to the packet.

A malicious node might propagate a false route error (RERR) and advertise the route again on subsequent RREQ from the source. This attack can significantly increase the network latency. The node just before the malicious node in the route detects and foils this attack by maintaining a history of RREPs received from the malicious node.

2 *Drop data packets*: A malicious node will drop all the data packets that it receives. In addition, it will not acknowledge to the sender that it has dropped a data packet. In other words, it will not send a RERR when it drops a packet. Reputation routing foils this attack. In such a scenario, the upstream neighbour of the node will give it a negative recommendation and the reputation of the node will be reduced; eventually the node will be weeded out of the network.

An attack, in which a node selectively drops packets, is difficult to counter. A malicious node can move around the network and selectively drop packets from different neighbours, without getting caught for a long time. Eventually the malicious node is likely to get caught, as its reputation with all the nodes whose packets it drops will reduce, albeit slowly due to its high mobility.

3 *Lavish behaviour*: A malicious node might try to do launch a denial of service attack by sending too many packets. The solution proposed in this paper does not counter a DoS attack. DoS attacks have been well researched and there are lots of techniques available in the literature for preventing such attacks.

4.1 Reputation model

The neighbours of a node store its reputation locally. In the proposed system, the nodes do not exchange the reputation of their respective neighbours. Although the exchange of reputation information of the neighbours among the nodes will make the system more robust, it will expose the system to a collusion of rogue nodes. If third party reputations are used, rogue nodes in the network can give recommendations to each other and increase each other's reputation. The rogue nodes do not even have to route packets to increase each other's reputation; they can exchange false recommendations among themselves. If the nodes use third party reputations, the target (node soliciting reputation of another node) will have to consider the credibility of the information source (node providing reputation of another node). As a result the node reputations will become multi-contextual. This will imply more work for the nodes at the routing layer, and will also increase the volume of the network traffic. More details about the exchange of reputations among semi-autonomous nodes are provided in Dewan and Dasgupta (2004).

We show that even when the third party reputations are not used and the good nodes only use the reputation of the other nodes accumulated from their own experience (number of packets routed), there is a significant reduction in the number of packets dropped by rogues. In addition, the rogue nodes will not be able to cheat other nodes by supplying incorrect reputation values. The downside of the approach is that the network becomes more vulnerable to grey hole attacks.

4.2 Modified AODV protocol

The proposed addition to the AODV protocol is divided into three mandatory phases and one non-mandatory phase. The first phase is the Route Lookup Phase, followed by the Data Transfer Phase and then the Reputation Phase. The Timeout phase is executed when the sender does not receive an acknowledgement before for the packet sent by it, before its timeout. We describe all the phases below:

- *Route lookup phase*: Consider a source node, S that has packets for the destination node, D (see Figure 1A). The routing module of the source node broadcasts a request (RREQ) for a route from node S to node D. All the neighbours of node S: nodes (1, 2, 3, 4) receive the RREQ and check their local routing tables for a path to D. If any of them has an active route to D, it sends (unicast RREP) the route back to node S. If multiple neighbours have routes to node D, they all reply back to node S. Node S chooses the route from the neighbour with the highest reputation (instead of the shortest path as in conventional AODV). If the two neighbours that have the same reputation send the route to node S, it chooses the shorter route, stores it in its routing table, and proceeds to the next phase. If a neighbour does not have the route to node D, it broadcasts the request to its neighbours, and the neighbours broadcast the request to their neighbours. This process continues till the TTL of the RREQ expires or the request reaches a node which has a route to node D or the request reaches node D (see Figure 1B). Each intermediate node updates its routing table with a path to source S, using the previous hop of RREQ as the next hop to node S.

  The destination node or an intermediate node sends a RREP to the source node, via the path followed by the RREQ in the opposite direction. Each intermediate node updates its routing table with a path to destination D using the previous hop of the RREP as the next hop for destination D. This process continues till the RREP reaches node S. Finally, node S inserts a record for destination D in its routing table. In case of multiple replies, a node chooses a route from the neighbour node that has the highest reputation among the candidates. The conflict among nodes that have the same reputation is resolved by selecting the next hop that has a shorter route to the destination. Like in AODV, node sequence numbers and message ids are used to ensure that there are no loops or stale information.

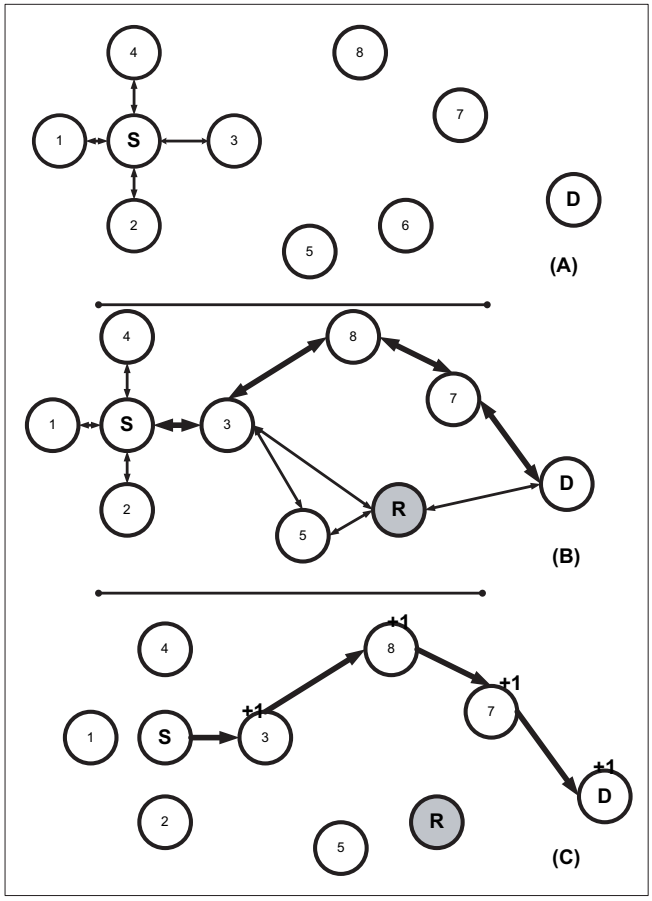
- *Data transfer phase*: Once node S has a route to destination D, it initiates the Data Transfer Phase (see Figure 1C). It searches its local routing table for the next hop to destination D and transmits the packet to the next hop. In addition, it stores the corresponding IP-ID[3] of the packet, the previous hop (=NULL) and the next hop in the local Neighbour-Packet Table. It starts a timer and the node should receive an acknowledgement for the packet from the destination before the expiry of the timer. The next hop checks if the packet has originated from a malicious node. If node S is malicious the next hop puts the packet to the end of its queue of incoming packets. If the node S is a good node, the current node sends the data packet to the next hop in the route, discovered in the previous phase. Similarly all intermediate nodes looks up the next hop on the route to the destination, in the routing table, and stores the IP-ID, the previous hop, and the next hop information in its neighbour-packet table. The intermediate nodes also start a timer, before which they should receive the acknowledgement for the packet from the destination. Once the packet reaches its

destination, the destination node D sends a signed[4] acknowledgement packet to the source S. The acknowledgement packet traverses the same route as the data packet, but in the opposite direction.

- *Reputation phase*: When an intermediate node receives an acknowledgement packet (see Figure 1C), it retrieves the record (inserted in the data transfer phase) corresponding to the IP-ID of the packet. The record contains the previous-hop and the next-hop nodes of the IP-ID. It forwards the acknowledgement to the previous-hop node and increments the reputation of the next-hop node. In addition, it deletes the entry for the IP-ID from the neighbour packet table and gives a recommendation of +1 to the node that delivered the acknowledgement. Once the acknowledgement packet reaches node S, it deletes the entry for the IP-ID from the neighbour-packet table and gives a recommendation of +1 to the neighbour that delivered the acknowledgement.
- *Timeout*: If the timer for a given packet expires at a node, the node retrieves the entry, corresponding to the IP-ID returned by the timer, from the neighbour packet table. If an entry is found, the node gives a negative recommendation (–1) to the next-hop node (retrieved from the neighbour packet table) for the IP-ID and deletes the entry from the neighbour packet table. If the reputation of the next-hop node goes below the threshold, T_r, the current node either deactivates the route in the routing table and sends an error message (RERR) to the upstream nodes in the route or performs a local repair by initiating another RREQ for the destination. If a record for the IP-ID is not found in the neighbour-packet implying it was deleted in the Reputation Phase, table the node ignores the time out.

  An important thing to note here is that an RERR sent by a node will not decrease its reputation with its neighbours. This is because a node sends an RERR only when, either it does not have a route to the destination or the next hop in the route to destination is rogue.

Figure 1 In (A) nodes 1, 2, 3 and 4 are neighbours of node S. Node S wants to find a route to node D. In (B) node S finds a route via S-3-8-7-D, which is not the shortest route, but the route which does not have a malicious node. The shortest route S-3-R-D has a malicious node so it is not used. The data and the acknowledgement are passed via the same route (D-7-8-3-S), in opposite directions. In (C), all the nodes in the route give +1 recommendation the next hop



5 Simulation

In the following sections we present the simulation scenario and analyse the results obtained from the simulation.

5.1 Simulation scenarios

The proposed scheme is simulated on an Intel 2.4 GHz machine using Linux Red Hat 8.0 with 512 MB RAM and the network simulator, *Glomosim*. Each iteration of the simulation runs for 300 min (real time). The simulated network consists of 50 uniformly allocated nodes in a space of 900 × 900 sq m (almost 150 soccer fields). The propagation limit of each node is set to –120 dBm, node transmission power to 15 dBm, and receiver

sensitivity to –91 dBm and antenna gain to 0. The *Initial Reputation* is set to 500 and the *Threshold Reputation*, T_r is set to 300. The number of rogue nodes is varied with each iteration.

In the application layer, a Constant Bit Rate (CBR) generator is used for nine distinct source-destination pairs. The results obtained by using only AODV and the proposed scheme are compared. The inter-node bandwidth is 250 Kbps and the MAC layer communication is done using 802.11. The simulation is divided into three scenarios. In the first scenario, the nodes are static, *i.e.*, the X, Y and Z coordinates of the nodes do not change with time. In the second scenario, a *random way point mobility* model is used. The nodes move at a speed between 10 m/s and 20 m/s till they reach their destination. They pause for 60 sec at their destination, and then move on to the next destination. In the third scenario, again a random way point mobility model is, but the pause time for the nodes is reduced to zero. The values of the following performance parameters are collected:

- *Network throughput*: The network throughput is the ratio of the total number of packets that reach their destination, to the total number of packets sent by the source(s) for all source-destination pairs:

  Throughput(%) = 100*[Packets sent – Packets dropped]/[Packets sent]

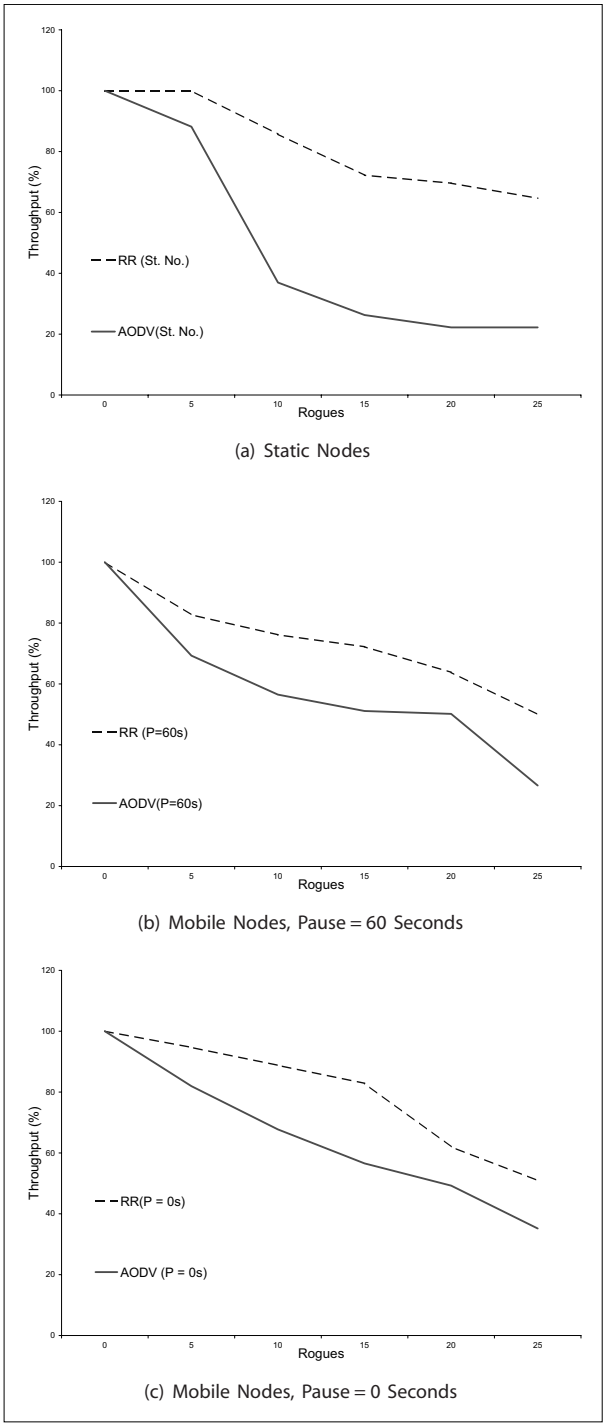
- *Average number of hops*: The average number of hops is the ratio of the total number of hops traversed by all the data packets to the total number of packets sent.

- *Average number of requests*: The average number of requests is the number of RREQs initiated per source-destination pair.

5.2 Simulation analysis

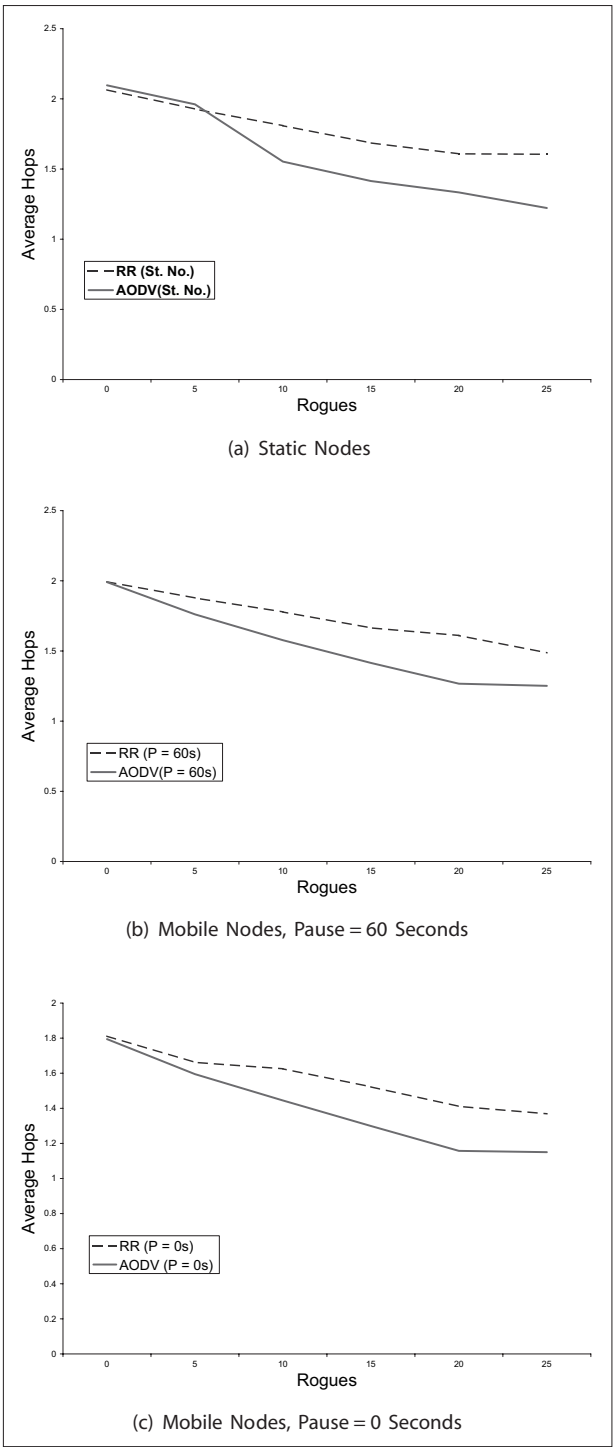
In scenario 1, when classic AODV is used and 50% of the nodes (25 nodes) are malicious, the throughput of the network is 22.22% (Figure 2). The use of reputation routing improves the throughput to 65% in the presence of same number (50%) of rogue nodes. The cost borne for this improvement is that the average number of hops increases from 0.8 to 1.6 hops and 1700 extra RREQs are issued for delivering 6000 packets to their respective destinations for 9 distinct source-destination pairs. The increase in the throughput is attributed to the new malicious-node-free route(s) found by the source, when the route found by vanilla AODV has a malicious node. When the known route to destination is infested with a malicious node, the number of RREQ increases because of the extra RREQs issued. Ideally, the source should receive a new route for every RREQ issued. However, this does not happen in the simulation because the destination node replies to the first RREQ received from a given broadcast. It ignores the RREQs received from other nodes for the same broadcast (identified by broadcast ID). If the two neighbouring nodes are close to each other, then the first node, which gets the channel, *i.e.*, an opportunity to transmit in the MAC layer, reaches the destination first. Assuming equal distance between the destination node and the two neighbours (of the source node), the source node gets the same route for the new RREQ issued till the other node is the first relaying node. This contention in the MAC layer increases the number of RREQs issued. This can be further reduced if the destination node replies to all RREQs and the source node decides the best route. This reduction in RREQs would be at the cost of more RREP traffic.

Figure 2 The drop in throughput with an increase in the number of malicious nodes in the network



Notes: RR = Reputation Routing, St. No = Static Nodes, P = Pause time.

Figure 3 Average hop length versus number of malicious nodes



Notes: RR = Reputation Routing, St. No = Static Nodes, P = Pause time.

In the second scenario, the nodes are mobile (20 m/s > speed > 10 m/s) with a pause of 60 sec at their destination. In this scenario the reputation scheme improves the throughput to 50%, in the presence of 50% rogue nodes; an improvement of 24% over the throughput shown by AODV. The difference in the number of requests issued in the two protocols is 1000 (more in reputation routing), while the average hop lengths increases from 1.25 to 1.48 (Figure 3). As illustrated in (Figure 2), the throughput is reduced when the nodes start moving and the number of RREQs increases with an increase in mobility. There is only a minor change in the average hop length. As the routes break when the neighbours of a node move out of its transmission range, the number of RREQs increases due to the increased number of routes needed between a source-destination pair.

In the final scenario Figure 2, the throughput of the system is reduced to 50% when 40% of the nodes are malicious (when AODV is used). The reputation scheme improves the throughput to 62% at a cost of 900 additional RREQs. The difference between the average hop lengths is less than one. *The throughput of the system drops from 65% from static nodes to 50% for mobile nodes in scenario 2. Due to the mobility, the neighbours of the nodes change. As a result, the nodes might encounter higher number of dropped packets before they can assign a reputation to their neighbours (some of them rogues) whom they have never seen before.*

6 Discussion

In this section we discuss the special cases encountered in *ad hoc* networks using reputation routing and the side effects of the use of reputation routing. In addition, this section presents suggestions for optimisation of the protocol:

- *Good nodes become a bottleneck*: In the current reputation scheme, the node with the highest reputation is selected as the next hop by its neighbour. As a result, the good nodes (nodes with higher reputations) become overloaded, while the other nodes become totally free. Good nodes in congested areas are more likely to get overloaded. Once the load on the good nodes is more than what the resources can manage, they start dropping packets and start loosing reputation. As a result, their incoming traffic is reduced to a level at which they can forward all the packets they receive for relaying. Subsequently, they start accumulating good reputation again. The oscillation of the reputation value of a node is reduced by selecting a set of reputed nodes and distributing the load among them. This might not be possible in a network with sparse topology where there are not many routes between the source and the destination. In such networks the oscillation is likely to be a function of number of alternative routes available for a given, source to the destination.
- *All nodes in the route get penalised*: If the route from the source to the destination contains a rogue node that drops the packet, all nodes (before the malicious node) in the route are penalised by their upstream neighbours. It might seem unfair, that a good node should get a negative recommendation just because there is a rogue in the route. Penalisation of all nodes injects the motivation in the nodes, to only select highly reputed node(s) as the next hop(s) for a given route. If only the rogue node is penalised the other nodes in the route will have no motivation to disallow the rogue node in the route. As a result the rogue node would be able to maintain two routes,

first route with a good next hop for its own packets, and the second route with another rogue node as the next hop, for forwarding packets originating from other nodes.

- *Why should a node forward the packet?* It is possible that a good node might realise that the packet will be dropped by the next hop node which is malicious. Hence the question is why should the good node even bother to forward a packet to the malicious node? The answer is that all nodes try to be in a good neighbourhood. In other words, a good node disregards the existence of a malicious neighbour by purging its entry out of the neighbour table. Therefore, it never sends a RREQ to the malicious node. Hence the likelihood that the packet will be dropped by a malicious node, after a good node has forwarded it is even lower. Some rogue nodes might decide to drop the acknowledgement instead of the packet. Malicious nodes will not benefit from this strategy. If they really want to disrupt the network, they can drop the packet in the first place. The scenario in which a malicious node drops alternate packet and hence keeps its reputation constant, can be circumvented by deducting a higher value (>1) from the reputation of a node for dropping packets.

- *Increased traffic volume*: In the simulation, the destination of the packet acknowledges receiving each packet, to the source of the packet, via the path traversed by the packet. This acknowledgement increases the network traffic. Alternatively, the sender can intercept the returned TCP acknowledgement, to ascertain that the previous packet has reached its destination. This approach will reduce the traffic volume considerably. The drawback of this approach is that it needs access to information across the layers of the network stack.

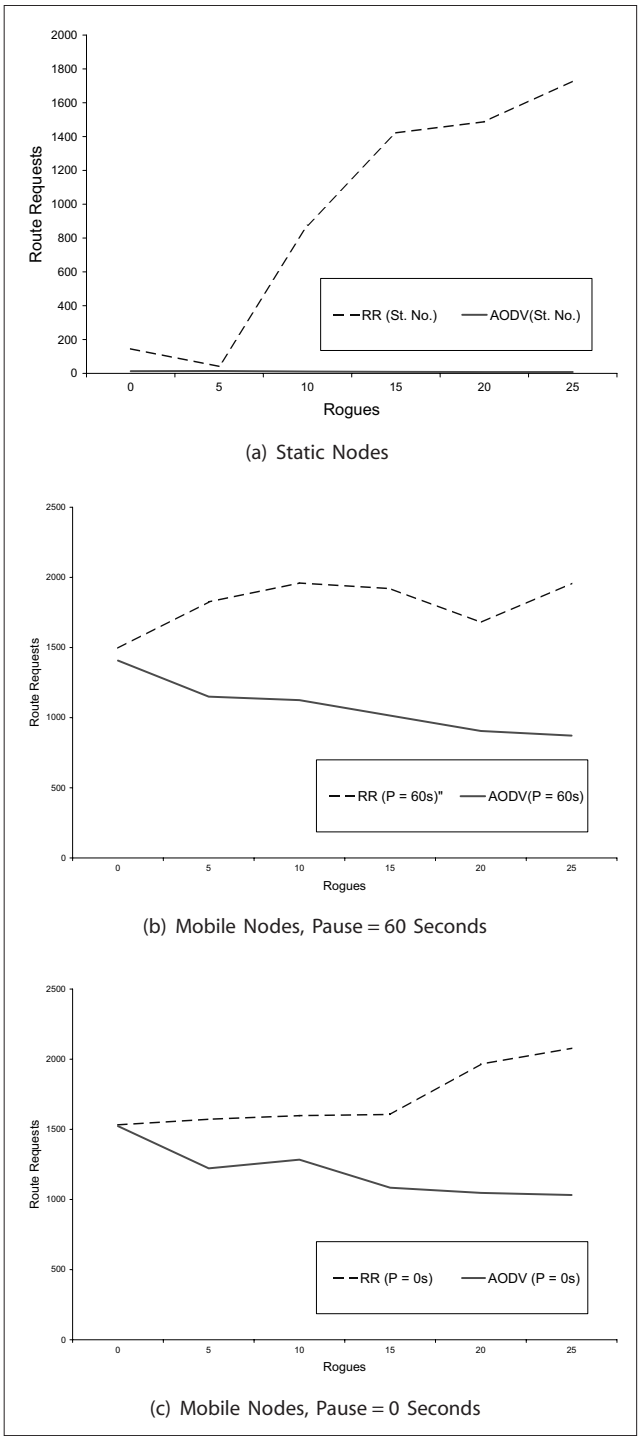
- *Poor nodes are penalised*: Nodes with lower resources, such as PDAs, are unable to route packets for other nodes due to the scarcity of resources. Such nodes loose reputation because of they are forced to drop packets due to the shortage of resources. Eventually, their reputation goes below the threshold, T_r, and these nodes are considered as rogue nodes.

  This problem can be solved in two ways. If a non-malicious node acknowledges dropping a packet to the source, *i.e.*, sends a RERR when it drops a packet,[6] the previous hop on the upstream tries to find an alternative route and circumvents the concerned node. In this fashion, a venial node can save itself from a flood of traffic and still maintain a good reputation. A malicious node can also pose as a node with low resources and get circumvented, thereby saving its resources. Although this would allow the malicious node to save its resources, it would not be able to drop any data packets. The above strategy can only be applied if a venial node has resources to send the error message. If it does not have the required resources, then it looses its reputation, and is eventually considered malicious.

  This problem is also solved by attaching a list of the resources of a node in its identity certificate (assuming that the structure of the identity certificate allows it). This class of nodes is penalised for failing to route packets – but the penalty inflicted on them is only a fraction of what is inflicted on a node with a large volume of resources. For example, consider node A, which possesses half the memory and half the processing power of most of the other nodes in the network. If a node fails to route a packet sent by another node, it gets a recommendation of –0.5, instead of –1. In this way, the system is democratised.

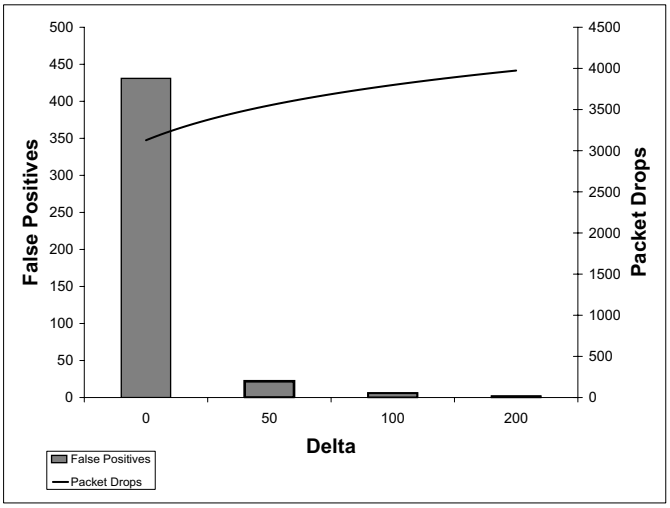
- *Number of RREQs increases*: In AODV, the source and intermediate nodes broadcast the RREQs. The RREP is unicast back to the node from which the first RREQ was received by the destination. This policy enables the source node to locate the shortest route to the destination. In reputation routing, this policy increases the number of RREQs required for a fresh route from the source to the destination because of the contention at the MAC layer as shown in (Figure 4). The RREQ volume can be reduced if a destination can maintain a list of the RREQs received for a given broadcast and randomly select one to reply. As a result, the source node S will get a new route which may or may not have rogue nodes.
- *False positives*: As expected, when the difference between the initial reputation (T_i) of the node and the threshold reputation, T_r, is increased from 0% to 40% of the initial reputation the number of false positives decreases from 411 to 2 for a simulation when 50% of the nodes are actually malicious as shown in Figure 5. If one good node is considered malicious by three different nodes, it results in three false positives. Besides the number of packets dropped increase with the difference between the initial and the threshold reputation, T_r. Once the difference is increased to 40% of the initial reputation, the packet drop rate becomes more or less constant.

Figure 4 Number of RREQs versus number of malicious nodes



Notes: RR = Reputation Routing, St. No = Static Nodes, P = Pause time.

Figure 5 False positives and packet drops versus rogues



Notes: Malicious nodes = 27, Total nodes = 50, Packets sent = 6000, Delta = (Initial Reputation (T_i) – Threshold Reputation, T_r), False positive: a good node whose reputation goes below threshold.

7 Conclusions and future work

The reputation of the nodes, based on their previous relaying history, cannot only be used to increase the throughput of an *ad hoc* network with rogue nodes, but also to motivate nodes to cooperate. The reputation scheme, presented in this paper, improves the throughput of the network to 65% when 40% of the nodes in the network is rogues. The cost of this improvement is the increased number of RREQs generated by the source node for finding rogue free routes to the destination. The packet acknowledgement from the destination node obviates any need for promiscuous mode listening. This not only saves the energy of the nodes but also provides higher level of privacy to the nodes. The simulations also show that there are only a minimal number of false positives.

More analysis is needed to evaluate the benefits of the reputation system in various mobility scenarios. Specifically, the utility of this mechanism in inter-vehicular networks and sensor networks is work that will be done in the future.

Acknowledgements

The authors would like to acknowledge the many helpful suggestions of two anonymous reviewers and Editor-in-Chief of the journal.

References

Allman, M., Paxson, V. and Stevens, W. (1999) *RFC-2581 – TCP Congestion Control*, RFC, April.

Bansal, S. and Baker, M. (2003) 'Observation-based cooperation enforcement in ad hoc networks', Research Report cs.NI/0307012, Stanford University, July.

Buchegger, S. and Boudec, J.Y.L. (2002a) 'Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks', *Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, Canary Islands, Spain: IEEE Press, pp.403–410.

Buchegger, S. and Boudec, J.Y.L. (2002b) 'Performance analysis of the confidant protocol', *MobiHoc2002*, Lausanne, Switzerland: IEEE Press, pp.226–236.

Buttyán, L. and Hubaux, J.P. (2000) 'Enforcing service availability in mobile ad-hoc WANs', *1st ACM International Symposium on Mobile Ad hoc Networking and Computing*, Boston, MA: ACM Press, pp.87–96.

De Capitani di Vimercati Damiani, S. and Paraboschi, S. (2003) 'Managing and sharing servents' reputations in p2p systems', *Knowledge and Data Engineering, IEEE Transactions*, IEEE, pp.840–854.

Dewan, P. and Dasgupta, P. (2004) 'PRIDE: Peer-to-peer reputation infrastructure for decentralized environments', *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters WWW*, pp.480–481.

Dewan, P., Dasgupta, P. and Bhattacharya, A. (2004) 'On using reputations in ad hoc networks to counter malicious nodes', *QoS and Dynamic Systems, (In Conjunction with IEEE ICPADS)*, IEEE, July.

Gupte, S. and Singhal, M. (2003) 'Secure routing in mobile wireless ad hoc networks', *Ad hoc Networks*, Vol. 1, pp.151–174.

Hu, Y., Perrig, A. and Johnson, D. (2002) 'Ariadne: a secure on-demand routing protocol for ad hoc networks', citeseer.ist.psu.edu/hu02ariadne.html.

Kamvar, S.D., Schlosser, M.T. and Garcia-Molina, H. (2003) 'The eigentrust algorithm for reputation management in p2p networks', *Proceedings of the Twelfth International Conference on World Wide Web*, Budapest, Hungary: ACM Press, pp.640–651.

Kevin Lai, I.S., Feldman, M. and Chuang, J. (2003) 'Incentives for cooperation in peer-to-peer networks', *Workshop on Economics of Peer-to-Peer Systems*.

Lidong Zhou, Z.J.H. (1999) 'Securing ad hoc networks', Special Issue on Network Security, *IEEE Network*, IEEE, November–December.

Liu, Y. and Yang, Y.R. (2003) 'Reputation propagation and agreement in mobile ad-hoc networks', *Wireless Communications and Networking*, IEEE, Vol. 3, March.

Marti, K.L., Giuli, T.J. and Baker, M. (2000) 'Mitigating routing misbehavior in mobile ad hoc networks', *ACM/IEEE International Conference on Mobile Computing and Networking*, ACM/IEEE.

Marti, S. and Garcia-Molina, H. (2003) 'Identity crisis: anonymity vs. reputation in P2P systems', *Third International Conference on Peer-to-Peer Computing (P2P'03)*, p.134.

Michiardi, R.M.P. (2002) 'Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks', *Communication and Multimedia Security*, Portoroz, Slovenia: IEEE.

Obreiter, P., Koenig-Ries, B. and Klein, M. (2003) 'Stimulating cooperative behavior of autonomous devices – an analysis of requirements and existing approaches', *Second International Workshop on Wireless Information Systems (WIS2003)*, Angers, France.

Papadimitratos, P. and Haas, Z.J. (2002) 'Secure routing for mobile ad hoc networks', *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, January.

Perkins, C.E. and Royer, E.M. (1999) 'Ad-hoc on-demand distance vector routing', *2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans: IEEE.

Ramanujan, S.N.T. and Kudige, R. (2003) 'Techniques for Intrusion-Resistant Ad hoc Routing Algorithms (TIARA)', *DARPA Information Survivability Conference and Exposition*, IEEE, April, pp.98–100.

Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C. and Belding-Royer, E.M. (2002) 'A secure routing protocol for ad hoc networks', *International Conference for Networking Protocols*.

Seyung, Y., Naldurg, P. and Kravets, R. (2001) 'Security-aware ad-hoc routing for wireless networks', Technical report, University of Illinois at Urbana Champaign, August.

Stajano, F. and Anderson, R. (1999) 'The resurrecting duckling: security issues for ad-hoc wireless networks', *Security Protocols, 7th International Workshop*, LNCS VERLAG.

Wang, Y. (2003) 'Bayesian network-based trust model in peer-to-peer networks', *Proceedings of the Autonomous Agents and Multi Agent Systems 2003 Conference, AAMAS03*, Melbourne, 14–18 July.

Zhong, S., Chen, J. and Yang, R. (2002) 'Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks', *IEEE INFOCOM*, San Francisco: IEEE Press.

Notes

1 Unlike the conventional definition of the throughput that measures packets transferred per second, we measure number of packets dropped/number of packets sent besides measuring the number of hops between the source and the destination, as we assume consistent inter-hop delay.

2 Bad mouthing is when a peer malicious incriminates another peer of being malicious.

3 IP-ID is a 16 bit packet identifier which is unique for every packet in the network.

4 Signed acknowledgements are transmitted in plain text.

5 The CA is trusted by all nodes and may be internal or external to the *ad hoc* network.

6 This is the default behaviour of nodes in AODV.