On Using Reputations in Ad hoc Networks to Counter Malicious Nodes

Prashant Dewan, Partha Dasgupta and Amiya Bhattacharya
Department of Computer Science and Engineering
Arizona State University
{dewan, partha, amiya}@asu.edu

Abstract

Nodes in mobile ad hoc networks have a limited transmission range. Hence the nodes expect their neighbors to relay packets meant for far off destinations. These networks are based on the fundamental assumption that if a node promises to relay a packet, it will relay it and will not cheat. This assumption becomes invalid when the nodes in the network have tangential or contradictory goals.

The reputations of the nodes, based on their past history of relaying packets, can be used by their neighbors to ensure that the packet will be relayed by the node. This paper introduces a reputation scheme for ad hoc networks. Instead of choosing the shortest path 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% malicious nodes, improves the throughput of the system to 65%, from 22% throughput provided by AODV. This improvement is obtained at the cost of a higher number of route discoveries with a minimal increase in the average hop length. [1]

1. Introduction

Most mobile ad hoc networks are intended to be 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 neighboring nodes in the network to route the packet to its destination node. The routing protocols used in the current generation of mobile ad hoc networks, like *Dynamic Source Routing* (DSR)[16], and *Ad hoc On Demand Distance Vector Routing Protocol* (AODV)[15], are based on the principle that all nodes will cooperate. Certain nodes in an ad hoc network might become antagonistic and hence refuse to cooperate with each other. 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 malicious nodes.

The simulation results show that for an ad hoc network with static nodes, the network throughput drops by more than half when 40% of the nodes are malicious. This throughput further reduces, with an increase in the number of malicious nodes, or when the nodes are made mobile. Depending on the location of the malicious nodes in the network and the network topology, some nodes experience worse throughput than what is mentioned above.

In this paper, the reputation of nodes in an ad hoc network is used to identify and subsequently circumvent the malicious nodes. The reputation of a node is not contextual and is a function of the number of packets forwarded by a node. In other words, the reputation of a node is a function of only the number of data packets that have been previously relayed by the node. It does not encompass any other attribute.[2] 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, 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 a routing protocol for ad hoc networks. The source node sends the data packet to the first hop with the 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 that updates its reputation table by giving a recommendation of +1 to the first hop. All the intermediate nodes in the route give a recommendation of +1 to their respective next hop in the route and update their local reputation tables. If there is a malicious node in the route, the data

[1] This work is partially supported by DARPA/Spawar, AFOSR and NSF.

[2] The quality of the radio links between two neighboring nodes can be included in the calculation of the reputation for the next hop but in this paper we assume that if two nodes are neighbors they have good quality links. We are currently working on inclusion of quality of inter-neighbor radio links for calculation of reputation of the neighbors of the node.

packet does not reach its destination. As a result, the source does not receive any acknowledgment for the data packet in the stipulated time. The source gives a recommendation of -1 to the first hop on the route. The intermediate nodes propagate this recommendation in the route upto the node that dropped the packet. In other words, all the nodes between the malicious node and the sender, including the malicious node, get a recommendation of -1.

The salient features of the proposed reputation system are: 1) circumvention of malicious nodes, 2) injection of motivation to cooperate among nodes, 3) decentralized collection and storage of reputations and 4) 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 having a significant impact on the routing performance. Unlike the previous research, OCEAN [1], CONFIDANT [2, 3], CORE[14], Nuggets [4] and SPRITE[19] which are built on DSR and require the nodes to snoop into the activities of their neighbors, the proposed reputation system is built on AODV, and the nodes in the network do not have to monitor their neighbor's activities. On the positive side, the proposed approach is resilient to collusion of malicious nodes (unlike OCEAN[1]), and does not need a central server (required by SPRITE[19]). In addition, the proposed approach does not need any tamper proof hardware (required by Nuggets[4]) and is not based on preexisting relationships between nodes (unlike CONFIDANT[2]) . On the negative side, the proposed system mandates that the destination should send an acknowledgment (for a set of packets received) and the acknowledgment should reach the source, before the good nodes get their due credit (for routing the set of packets)

The remainder of this paper is organized as follows. Section 2 presents the model of the ad hoc network used for the reputation scheme, and provides the corresponding background information. Section 3 explains the reputation-based scheme presented in this paper. Section 4 discusses in detail the simulation environment and parameters; and Section 5 presents an analysis of the simulation results. Section 6 discusses the pros and cons of the proposed approach, while Section 7 presents the related work. Section 8 provides the conclusion and discusses future work.

2. Model & Attacks

The network model comprises of semi-autonomous nodes (like laptops or PDA's) capable of setting up 802.11 connections with other nodes in their neighborhood. The public keys of the nodes are used to uniquely identify the nodes. The identities may be centrally issued to the nodes, or may be generated using self-certification[7]. Self-certified identities can be generated by using the MAC address of the node as the seed for generating its public key. Self-certified identities are susceptible to Sybil attacks[8] in the absence of any other constraint hence we advocate the use of identities endorsed by a trusted Certificate Authority (CA).We also assume that the source node has the identity certificate of the destination node.

The nodes within the transmission range of a given node are termed as the neighbors of the node. The sender of the packet receives an acknowledgment for each data packet. The acknowledgment packet reaches the sender, following the reverse path followed by the data packet. The destination might explicitly generate an acknowledgment in the network layer, or the sender can use the TCP acknowledgment to ascertain that a given packet has reached its destination. For every packet that reaches its destination, the nodes in the respective route get +1 from the respective node, that preceeds 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.

If the reputation of a node falls below the threshold reputation R, it is considered to be a malicious node. The threshold value R is not a global parameter;it is pre-decided by each node for itself. The nodes can change their threshold parameters depending on the percentage of packets successfully routed in the past.The development of quantitative models for calculating the thresholds is an area of future research. 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 good node (a node with reputation greater than 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 apriori 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 [9]. 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 neighbors of the sender receive the RREQ. If any of the neighbors 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 neighbor does not have the route to the destination, it rebroadcasts the RREQ. Finally, the RREQ reaches the destination, which sends a reply, RREP, back to the sender, or the RREQ reaches another node which has a route to the destination and which sends an RREP to the sender or the request times out and the sender sends another RREQ after some time.

Once the sender has the route to the destination, it sends the data packet towards the destination, on the 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. 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. This attack can be foiled in reputation routing. 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, it will be considered as malicious and will eventually be ostracized.

   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 lowermost priority to the packets originating from rouge nodes. Hence rouge 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 RREP's 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 neighbor 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.

3. **Lavish Behavior:** 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.

The neighbors of a node store its reputation locally. In the current model, the nodes do not exchange the reputation of their respective neighbors. Although the exchange of reputation information of the neighbors among the nodes, will make the system more robust, it is not incorporated in this model. This is because ad hoc networks generally have a few nodes hence each node can find the malicious nodes by relying on its own experience rather than the experience of its peers. In addition, if the nodes exchange the reputations of other nodes, 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 [3]. Two neighbors of the same node might have different reputation information of that node. Such a scenario occurs when one of the neighbors has previously sent packets via the node while the other has not.

This approach (of not sharing the reputations among nodes) is resilient to collusion of malicious nodes. Malicious nodes in the network can give recommendations to each other and increase each other's reputation. The malicious nodes do not even have to route packets to increase each other's reputation, they can exchange false recommendations among themselves. As the good nodes only use the reputation of the other nodes accumulated from their own experience (number of packets routed), malicious nodes will not be able to cheat them by supplying incorrect reputation values. The downside of the approach is that a malicious node can move around the network and selectively drop packets from different neighbors, without getting caught for a long time. Eventually the malicious node will be caught.

The proposed reputation system is divided into three phases. The first phase is the Route Lookup Phase, followed by the Data Transfer Phase and then the Reputation Phase. We describe all the three phases below:

1. **Route Lookup Phase:** Consider a source node, S that has packets for the destination node, D. The routing module of the source node broadcasts a request(RREQ) for a route from node S to node D. All the neighbors of node S receive the RREQ and check their local routing tables for a path to D. If any of them has a route to D, it sends (unicast RREP) the route back to node S. If multi-

ple neighbors have routes to node D, they all reply back to node S. Node S chooses the route from the neighbor with the highest reputation. If two neighbors 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 neighbor does not have the route to node D, it broadcasts the request to its neighbors, and the neighbors broadcast the request to their neighbors. 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. 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, the nodes chooses a route from the neighbor 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.

2. **Data Transfer Phase:** Once node S has a route to destination D, it initiates the Data Transfer Phase. If the packet has originated from a malicious node, the packet is put back to the end of the queue (of incoming packets) of the current node. If the packet has originated from a good node, the current node sends the data packet to the next hop in the route, discovered in the previous phase. In addition, it stores the corresponding IP-ID of the packet, the previous hop (=NULL) and the next hop in the local Neighbor-Packet Table. It starts a timer before it should receive an acknowledgment for the packet, from the destination.

   An intermediate node 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 neighbor-packet table. The intermediate nodes also start a timer, before which it should receive the acknowledgment for the packet from the destination. Once the packet reaches its destination, the destination node D sends a signed acknowledgment packet to the source S. The acknowledgment packet traverses the same route as the data packet, but in the opposite direction.

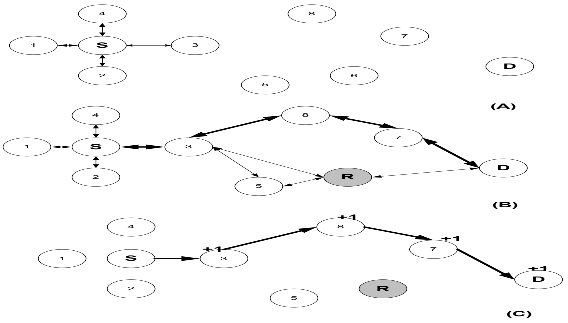

Figure 1. ***In (A), nodes 1, 2, 3 and 4 are neighbors of node S. Node S wants to find a route to node D. In (B) node S finds a route via S-3-8-7-D, which is not the shortest route, but the route which does not have a malicious node. The shortest route S-3-R-D has a malicious node so it is not used. The data and the acknowledgment 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***

3. **Reputation Phase:** When an intermediate node receives an acknowledgment packet, 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 acknowledgment 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 neighbor packet table. Once the acknowledgment packet reaches node S, it deletes the entry for the IP-ID from the neighbor-packet table and gives a recommendation of +1 to the neighbor that delivered the acknowledgment.

4. **Timeout:** Once 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 neighbor packet table. If an entry is found, the node gives a negative recommendation (-1) to the next-hop node (retrieved from the neighbor packet table) for the IP-ID and deletes the entry from the neighbor packet table. If the reputation of the next-hop node goes below the threshold, 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 neighbor-packet, table the node ignores the time out.

3. Simulation

In the following sections we present the simulation scenario and the analysis of the results.

3.1. Simulation Scenario

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 minutes. The simulated network consists of 50 uniformly allocated nodes in a space of 900*900 square-meters. The propagation limit of each node is set to -120dBm. The *Initial Reputation* is set to 500 and the *Threshold Reputation* is set to 300. The number of malicious nodes are varied with each iteration.

In the application layer, a Constant Bit Rate generator (CBR), is used for 9 distinct routes. The results obtained by using only AODV and only 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 phases. In the first phase, the nodes are static; i.e., the X, Y and Z coordinates of the nodes do not change with time. In the second phase, 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 stop for 60 seconds at their destination, and then move on to the next destination. In the third phase, a random way point mobility model is also used, but the pause time for the nodes is reduced to zero. The values of the following parameters are collected:

1. **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. Throughput (%) = [Packets Sent - (Packets Dropped by Malicious nodes)]*100/ [Packets Sent].
2. **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.
3. **Average Number of Requests:** The average number of requests is the number of route requests (RREQ) initiated per source-destination pair. The drop in throughput increases with an increase in the number of malicious nodes in the network.

3.2. Simulation Analysis

In phase 1 the throughput of the network is reduced to 22.22%, with AODV, when 50% of the nodes are malicious (Figure 2). When half of the nodes in the network are malicious, the reputation routing improves the throughput to 65%. The cost borne for this improvement is that the average number of hops increases from 0.8 to 1.6 and 1700 extra route requests are issued for delivering 6000 packets.

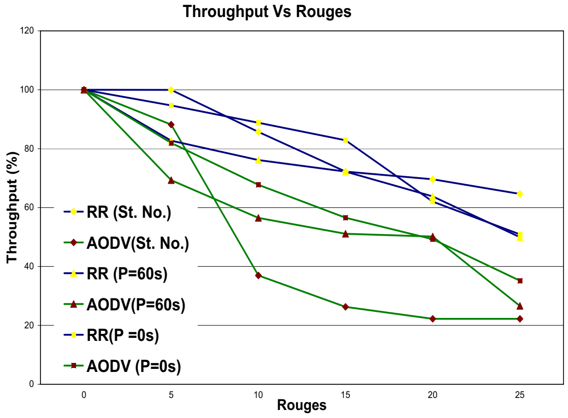

Figure 2. ***The drop in throughput with an increase in the number of malicious nodes in the network. RR=Reputation Routing. St. No=Static Nodes P=Pause Time***

The increase in the throughput is attributed to the new malicious-node-free route found by the source, when the currently known route has a malicious node.When the known route to destination is infested with a malicious node the number of route requests (RREQ) increases because of the extra route requests issued. Ideally, the source should receive a new route for every route request (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 neighboring nodes are close to each other, then the first node, which gets an opportunity to transmit in the MAC layer, reaches the destination first. Assuming equal distance between the destination node and the two neighbors (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. In the second phase, the nodes are mobile with a pause of 60 sec at their destination. In this phase the reputation scheme improves the throughput to 50%, in the presence of 50% malicious 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, while the average hop length 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 route requests increases with an increase in mobility. There is only a minor change in the

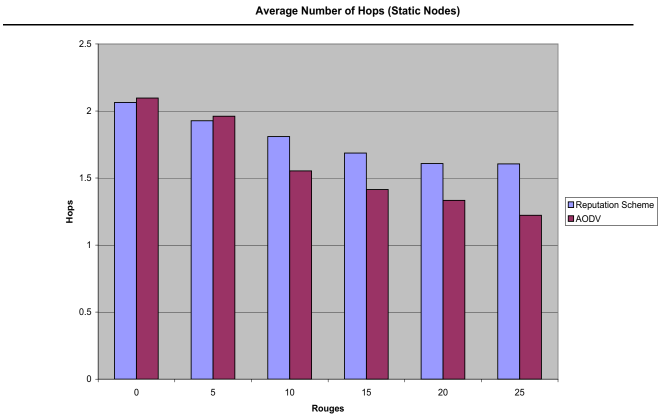

Figure 3. ***Average Hop Length vs. Number of Malicious nodes. RR=Reputation Routing. St. No=Static Nodes P=Pause Time***

average hop length. As the routes break when the neighbors of a node change, the number of route requests increases due to the increased number of routes needed between a source-destination pair.

In the final phase, 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 route requests. The difference between the average hop length is less than one.

4. Discussion

The nodes in an ad hoc network are semi-autonomous. Hence the reputation scheme motivates them to allocate their resources to other nodes in the network. As the sender relays the packet only to highly reputed neighbors, it reduces the risk that its neighbors will intentionally drop the packet. The neighbors in turn forward the packets to nodes that have a high reputation with them. As a result, the number of packets intentionally dropped is reduced and the throughput of the system rises. The malicious nodes, i.e., the nodes that have a low reputation do not receive any packet, and hence cannot inflict any damage by dropping packets.

1. Good Nodes become a bottleneck: In the current reputation scheme, the node with the highest reputation is selected as the next hop by its neighbor. As a result, the good nodes (nodes with higher reputations) become overloaded, while the other nodes become totally free. 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 can be reduced by selecting a set of reputed nodes and distributing the load among them. The criteria for the selection and the policy to be used for the distribution of packets are future work.

2. Why should a node forward the packet? It is possible that a good node might realize 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 neighborhood. In other words, a good node disregards the existence of a malicious neighbor by purging its entry out of the neighbor table. Therefore, it never sends a RREQ to the malicious node. Hence the likely hood that the packet will be dropped by a malicious node, after a good node has forwarded it is even lower. Some malicious nodes might decide to drop the acknowledgment 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.

3. 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 acknowledgment increases the network traffic. Alternatively, the sender can intercept the returned TCP acknowledgment 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.

4. Poor Nodes are penalized: 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 such nodes drop packets due to the shortage of resources. Eventually, their reputation goes below the threshold, and these nodes are considered as malicious nodes.

   This problem can be solved in two ways. If a node acknowledges dropping a packet to the source, i.e., sends a RERR when it drops a packet, 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

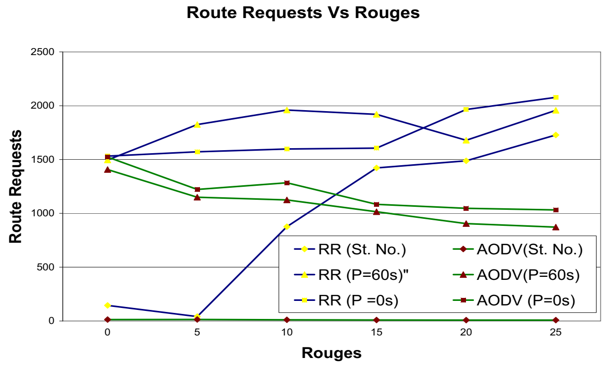

Figure 4. ***Number of RREQs vs. Number of Malicious nodes.RR=Reputation Routing. St. No=Static Nodes P=Pause Time***

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 can also be 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 penalized 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 democratized.

5. Number of RREQs increases: In AODV, the source and 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 malicious nodes.

6. False Positives As expected, when the difference between the initial reputation of the node and the threshold reputation (TR), 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. Once the difference is increased to 40% of the initial reputation, the packet drop rate becomes more or less constant.

5. Related Work

The reputations in decentralized systems have been extensively investigated in [5, 6, 7, 9, 14, 15]. While researchers have investigated the use of reputation peer-to-peer networks, Obreiter et. all have used reputations for secure routing in ad hoc networks in [11]. The authors in [11] have proposed account-based and reputation-based mechanisms for injecting motivation in autonomous or semi-autonomous nodes of a network. Efficient routing (not secure routing) in ad hoc networks is also a well-researched topic. AODV is explained in detail in [12] and DSR in [13]. Both AODV and DSR assume cooperative behavior of the network nodes. ARAN [13]is a secure routing protocol which uses cryptographic techniques like X.509 certificates.

Buttayan and Hubaux have proposed *Nuglets* [4] and Zhong et. all. have proposed *SPRITE* [16]. Both Nuglets and SPRITE are remuneration based systems in which the routing nodes receive payment for each packet routed by them. While Nuglets necessitates a tamper-proof security module, SPRITE is dependent on a centralized server named as Credit Clearing Service (CCS). As evident from the description above, the proposed reputation infrastructure does not need any tamper-proof security module or a CCS. In [8] Lai et. al. have proposed a the use of a WatchDog to detect malicious behavior and a 'PathRater' for ranking of paths. WatchDog and PathRater together improve the throughput of the network in the presence of malicious nodes. Their approach does not penalize malicious nodes and hence the malicious nodes can get away without routing any traffic. In contrast, in the proposed approach, the good nodes penalize the malicious nodes by assigning the lowest priority to the packets that originate at the malicious nodes.

Boudec and Buchegger have proposed a reputation infrastructure CONFIDANT [2, 3] to penalize the malicious nodes by weeding the freeloaders. CONFIDANT uses preexisting trust relationships among nodes to ascertain the credibility of third party reputation information. The proposed technique does not use third-party reputation information and hence is independent of any preexisting trust among nodes in an ad hoc network. Michiardi and Molva have proposed a generic reputation infrastructure, CORE [10]. CORE can be used either in the application or in the network layers, while the proposed infrastructure has been optimized for the routing layer. Besides, CORE categorizes nodes with no reputation and nodes with bad reputation in the same category. In the proposed infrastructure, malicious nodes posses negative or low reputation and all nodes get a residual reputation when they join the network for the first time. Bansal and Baker have proposed OCEAN [1] to avoid the trust-management machinery required for third party reputation exchange. It improves the throughput in the presence of malicious nodes. Unlike the infrastructure presented in this paper, OCEAN is not resilient to the collusion of malicious nodes and can be subverted if an attacker gains control overtwo or mode nodes.

In all the existing systems [2-4, 10, 14, 19] the nodes have to watch their neighborhood which not only necessitates promiscuous modes of operation, but also overloads the nodes with busy neighborhoods. The proposed infrastructure does not use the ’neighborhood watch’ technique. In addition, nodes can only ascertain if their neighbors forwarded the packets that they receive but cannot ascertain if the forwarded packet reached the destination or even the next hop. Besides all the existing systems[2-4, 10, 14, 19] 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 [8] selects the route by considering the reputations of all the nodes in the route, the proposed scheme only considers the next hop reputation.

6. Conclusions & Future Work

The reputation of the nodes, based on their previous relaying history, can not only be used to increase the throughput of an ad hoc network, but also to motivate nodes to cooperate. The reputation scheme improves the throughput to 65% with 40% malicious nodes, in a network where the nodes are static. The cost of this improvement is the increased number of route requests. The throughput can be further improved at the cost of extra messages, by making the nodes exchange their reputation databases using cryptographic protocols for ascertaining the credibility of the source of information and the correctness of the reputation information obtained. Quantitative models for calculating threshold values R will increase the usability of the proposed approach.

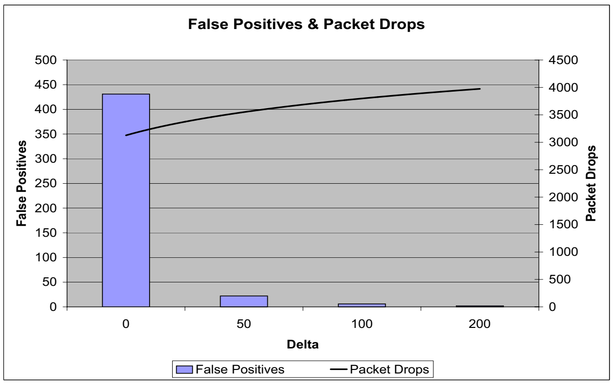

Figure 5. False Positives vs. Packet Drops. Malicious nodes= 27, Total Nodes = 50 Packets Sent=6000 Delta= (Initial Reputation - Threshold Reputation) False Positive: A good node whose reputation goes below threshold

References

[1] S. Bansal and M. Baker. Observation-Based Cooperation Enforcement in Ad hoc Networks. Research Report cs.NI/0307012, Stanford University, 2003.

[2] S. Buchegger and J.-Y. L. Boudec. Nodes Bearing Grudges: Towards Routing Security, Fairness and Robustness in Mobile Ad hoc Networks. In *Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 403–410, Canary Islands, Spain, 2002. IEEE.

[3] S. Buchegger and J.-Y. L. Boudec. Performance Analysis of the CONFIDANT Protocol,. In *MobiHoc*, pages 226 – 236, Lausanne, Switzerland, 2002. IEEE.

[4] L. Buttyán and J.-P. Hubaux. Enforcing Service Availability in Mobile Adhoc WANs. In *ACM international symposium on Mobile ad hoc networking and computing*, pages 87–96, Boston, Massachusetts, 2000. ACM Press.

[5] S. De Capitani di Vimercati Damiani, S.Paraboschi. Managing and Sharing Servents’ Reputations in P2P Systems. In *Knowledge and Data Engineering, IEEE Transactions*, pages 840–54. IEEE, 2003.

[6] P. Dewan and P. Dasgupta. Pride: Peer-to-peer Reputation Infrastructure for Decentralized Environments. Technical report, Arizona State University, November 5, 2003.

[7] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the Twelfth International Conference on World Wide Web*, pages 640–651. ACM Press, 2003. Budapest, Hungary.

[8] I. S. Kevin Lai, M. Feldman and J. Chuang. Incentives for Cooperation in Peer-to-Peer Networks. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[9] S. Marti and H. Garcia-Molina. Identity-crisis: Anonymity vs. Reputation in P2P Systems. In *Third IEEE International Conference on Peer-to-Peer Computing*, 2003.

[10] R. M. P. Michiardi. Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad hoc Networks. In *Communication and Multimedia Security*, Portoroz, Slovenia, 2002. IEEE.

[11] P. Obreiter, B. Koenig-Ries, and M. Klein. Stimulating Cooperative Behavior of Autonomous Devices - an Analysis of Requirements and Existing Approaches. In *Second International Workshop on Wireless Information Systems (WIS2003)*, Angers, France, 2003.

[12] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, 1999.

[13] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. A Secure Routing Protocol for Ad hoc Networks. In *International Conference for Networking Protocols*, 2002.

[14] K. L. Sergio Marti, T. J. Giuli and M. Baker. Mitigating Routing Misbehavior in Mobile Ad hoc Networks. In *ACM/IEEE International Conference on Mobile Computing and Networking*. ACM/IEEE, 2000.

[15] Y. Wang. Bayesian Network-Based Trust Model in Peer-to-Peer Networks. In *Workshop on ”Deception, Fraud and Trust in Agent Societies” at the Autonomous Agents and Multi Agent Systems*, 2003.

[16] S. Zhong, J. Chen, and R. Yang. Sprite: A Simple, Cheat-proof, Credit-based System for Mobile Ad-hoc Networks. In *IEEE INFOCOM*, San Francisco, USA, 2002. IEEE Press.