A Reputation Based Scheme for Stimulating Cooperation in MANETs

Aruna Balasubramanian, Joy Ghosh and Xin Wang

University at Buffalo (SUNY)
201, Bell Hall, Buffalo, NY 14260-2000
Email:{ab42,joyghosh,xwang8}@cse.buffalo.edu

Abstract. A Mobile Ad hoc NETwork (MANET) is formed among a set of wireless nodes without infrastructure support, where network services such as routing are provided by the nodes themselves. In such a scenario, if the wireless devices refuse to provide network services (and thus not cooperate), they cannot be forced to do so because of the lack of a controlling authority. Non cooperation leads to a significant reduction in throughput and the network utilization reduces to below optimum value. In this work, we provide a decentralized reputation based scheme for stimulating cooperation among nodes. Every node is assigned a reputation, calculated by all neighbors of the node, and the reputation is high if the node cooperates and low if the node refuses to cooperate. By providing better services (as an incentive) for nodes with high reputation and no service (as punishment) for those with low reputation, a node can be forced to cooperate, to obtain better services and to avoid punishment. We provide security against tampering of reputation and a mechanism to identify nodes that move to a new neighborhood to avoid punishment. The scheme is also resilient to malicious nodes that try to generate false reputation values for its neighbors and is robust in the presence of mobility. The scheme has been simulated using GloMoSim and the results indicate that in the presence of non cooperative nodes, a significantly higher throughput can be obtained when using our reputation scheme.

1   Introduction

Mobile Ad hoc NETworks (MANETs) are self organizing networks formed among wireless mobile devices with minimum infrastructure requirement. Nodes that are within each others transmission range communicate directly, while those at a multi-hop distance rely on intermediate nodes to route their packets [1], [2]. Such networks not only lend themselves to critical situations such as disaster relief and the military, but are also suitable for other commercial applications.

As no central control may be assumed in MANETs, network services such as routing are provided based on an implicit trust among nodes. However, nodes may exhibit non cooperation by refusing to route packets due to several reasons such as power and other resource constraints, or an intent to deliberately disrupt the system. This is of concern in multi-hop communication and a large percentage of nodes refusing to cooperate may have a negative impact on the network throughput. It is thus important to stimulate cooperation among such nodes, to ensure optimum network utilization.

Literature has provided us with several approaches for stimulating cooperation. These approaches are mostly Incentive-based or Punishment-based. Incentive-based schemes ([3],[4],[5]) define *Incentive* to be a positive reenforcement granted to cooperating nodes. Incentives are

normally implemented using credits that are given to nodes that cooperate and forward packets. In turn, network services such as routing is provided to only those nodes that have good credit. However, in an incentive based solutions, a node at an unfavorable location may not get enough packets to forward, and thus may never be able to gain credits to forward its own packet. Also, in the absence of a central authority, ensuring tamper-proof manipulation of the crediting system may be complicated. On the other hand, Punishment based schemes identify and punish nodes that exhibit non cooperative behavior ([6],[7],[8]). These schemes define a metric called *Reputation*, which is the goodness of a node as perceived by its peers, and the reputation is reduced on evidence of non cooperation. A node with reputation lower than a threshold is punished. Cooperation is stimulated because of the node's desire to avoid a bad reputation and the subsequent punishment. However, no incentive is given to nodes with high reputation, which may tempt a node to cooperate just enough to maintain the reputation above the threshold and avoid punishment. Consequently, the network throughput may suffer.

In this work, we present a novel concept for stimulating cooperation by defining a reputation system that not only punishes misbehaving nodes but also rewards consistent good behavior. Our approach ensures that a node is allowed to use the network infrastructure even if it is not able to forward packets by virtue of its location. We have assumed a rational malicious threat model that is common in several scenarios where nodes do not belong to the same authority and thus, cannot be programmed to cooperate, but may disrupt the system to continue their non cooperation. Under this threat model, our reputation based cooperation scheme is secure against attacks including including distributing false reputation or moving to a new neighborhood to avoid punishment. We also provide a methodology for well behaved nodes to carry their reputation when they move to a new neighborhood. The protocol is developed comprehensively to address the concerns raised in [9] including the problems of building up credit and the ineffectiveness of global reputation.

The solution has been implemented using GloMoSim [10] and results indicate that the throughput is significantly increased when using the proposed cooperation scheme at an acceptable overhead.

2   System Model and Protocol Preliminaries

In this section, we will discuss the assumptions made in our protocol. We assume that the misbehaving nodes are malicious but rational. Thus, a node may be selfish and conserve resources at the expense of others, but will deliberately disrupt the system only if the malicious act has direct benefit to itself, in terms of allowing it to continue its selfish behavior or in improving performance.

We consider selfish behavior to be manifested by dropping packets, but if other manifestations are considered, the same reputation scheme can be extended, by appropriately modifying the monitoring mechanism. We assume that omni-directional antennas are used and all links have bi-directional symmetry.

The notations used in our protocol is described in Table 1 and the timers are described in Table 2.

3   Reputation based Trust System

Reputation, in the context of cooperation schemes, is defined as the goodness of a node perceived by other nodes. In our solution, reputation is a real number, and a higher value of reputation indicates that the node is cooperative while a smaller value of reputation indicates that the node

N_t	Neutral reputation
U_t	Upper threshold for reputation
L_t	Tolerant threshold for reputation
α	Rate of increase of reputation
β	Rate of decrease of reputation
$Rep_{i,j}^n$	Reputation of $Node\ i$, maintained by neighbor $Node\ j$ at time instance n
T_f	Tolerance factor
I_f	Incentive factor
K	Maximum number of malicious nodes
N	Number of nodes in the network

Table 1: Protocol Preliminaries

T_{valid}	Validity period of reputation
T_{Wait}	Time interval for overhearing a neighbor for transmission of a particular packet
T_{Report}	Frequency of propagating reputation reports
$T_{backoff}$	Back off time, when a node is congested
T_{Idle}	Time period to wait before joining the network infrastructure, in the $Idle\ mode$
T_{Redeem}	Time period within which reputation should be improved, in the $Redeem$ mode

Table 2: Notations used for Timers

is misbehaving. The reputation of a node is maintained by its neighbors, who monitor the node's behavior and update its reputation accordingly. We define good behavior as correct forwarding of packets and misbehavior as dropping packets. The monitoring scheme is described in more detail in Section 4, where every node monitors and is in turn monitored by its neighbors. Like in the real world, every new node that joins the network is neither trusted or mistrusted, but is assigned a neutral reputation N_t. All reputations are valid for a time period T_{valid}. The upper threshold of reputation is U_t and the lower threshold is L_t. $L_t < N_t < U_t$, and are all real numbers.

3.1 Rate of increase and decrease of reputation

In our model we have chosen the reputation to increase at the rate of α and decrease at the rate of β, where $\alpha, \beta < 1$, and are real numbers. Careful considerations need to be made when choosing α, the rate of increase of reputation. If α is very large when compared to β, a node may cooperate and build up a high reputation in a short span of time, and continue to drop packets for a long time. Additionally, if a cooperating node is able to reach the upper reputation threshold U_t fast because of the high rate of increase, it may not have enough motivation to continue cooperation. The rate of decrease of a node's reputation also needs to be chosen appropriately. If the reputation is reduced at a low rate, a node can stay in a neighborhood long enough to exploit the network infrastructure. However, if the reputation is decreased at a higher rate, a genuine node that misbehaves temporarily because of network congestion, loss of connectivity or other reasons, may be punished unfairly.

In addition to choosing α and β more carefully, a neighbor does not decrease a node's reputation linearly, but as a function of the node's current reputation and the number of packets dropped. Thus, the reputation of $Node\ i$ is decreased by $Node\ j$ in a particular time period, when $f_{i,j}(x)$ in Eqn. 1 is greater than 0.

$$
\begin{aligned}
f_{i,j}(x) =\ & 0 && if\ x/Rep_{i,j} \leq T_f \\
=\ & x/(Rep_{i,j}) && Otherwise,
\end{aligned}
\tag{1}
$$

where x is the number of packets of $Node\ j$ dropped by $Node\ i$ in the time period, $Rep_{i,j}$ is the reputation of $Node\ i$ maintained by $Node\ j$ at the beginning of the time period, and T_f is the tolerance of the network, which is the number of packets per reputation value that may be dropped, before the reputation of a node is reduced.

Similarly, the reputation of $Node\ i$ is increased by $Node\ j$ in a particular time period, when $g_{i,j}(y)$ in in Eqn. 2 is greater than 0.

$$
\begin{aligned}
g_{i,j}(y) = &\ 0 && if\ y/Rep_{i,j} \leq I_f \\
= &\ y/(Rep_{i,j}) && Otherwise,
\end{aligned} \tag{2}
$$

where y is the number of packets forwarded by $Node\ i$ for $Node\ j$ in the time period. I_f is the Incentive factor of the network, which is the number of packets per reputation value that needs to be forwarded, before the node is given incentive.

Thus, if the node has high reputation, more packets need to be dropped before the reputation is reduced, while if the node has low reputation, dropping a smaller number of packets may result in decrease in reputation. This in itself is a soft incentive, where the network is more tolerant to nodes with higher reputation. However, it gets progressively difficult for cooperating nodes to increment their reputation. Thus, this algorithm prevents nodes from reaching the maximum limit U_t fast.

The reputation of a $Node\ i$ at time $n+1$ $(Rep_{i,j}^{n+1})$ is calculated recursively by $Node\ j$ based on the reputation at the previous time period n, represented as $Rep_{i,j}^{n}$ and is given by

$$
Rep_{i,j}^{n+1} = Rep_{i,j}^{n} + \alpha * (g_{i,j}(y)) - \beta * (f_{i,j}(x)) \tag{3}
$$

where x and y are the number of packets dropped and forwarded for $Node\ j$ respectively, during the time period [n, n+1].

4   Monitoring

A monitor is an entity that oversees nodes for evidence of misbehavior as well as good behavior, and updates the reputation of the node accordingly. In an ad hoc infrastructure-less network, it may not always be possible to appoint a trusted central agent for monitoring other nodes, as connectivity or availability of a central agent cannot be guaranteed. The authors of [7] proposed a monitoring scheme where the one hop neighbors of a node is responsible for monitoring the node. We adapt this monitoring scheme for our solution, where a neighbor waits for a time T_{wait} to overhear if the packet is transmitted to the next hop. If the packet is transmitted, the reputation of the node is increased, else the reputation of the node is decreased. Note that the reputation is not altered every time a packet is dropped or forwarded, but is recomputed periodically, using Eqn. 3.

4.1   Reputation distribution

Every node maintains a list of reputation of all its neighbors, and updates the reputation based on the neighbor's behavior. Fig. 1 represents a network where M, A and S are neighbors, and M, S and B are neighbors. From the design of our monitoring scheme, A monitors only those packets that it forwards to M and does not monitor the packets M receives from other nodes. Thus, the reputation information of M maintained by A may not be sufficient to make decision about the cooperation level of M. For example, in scenarios where A does not forward many packets to M, M may have an average reputation. However, M may forward packets for other nodes and this information should be known to A before it can make a decision regarding providing network services for M.

Thus, the reputation information is distribution among one-hop nodes to ensure that all nodes have knowledge of the cooperation level of all other nodes and maintain a consistent view
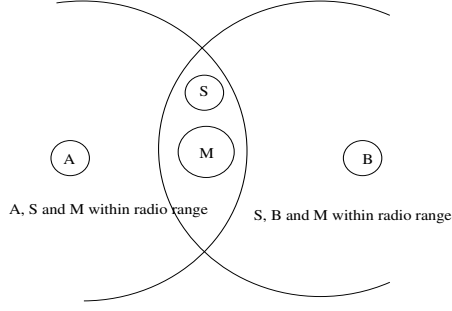
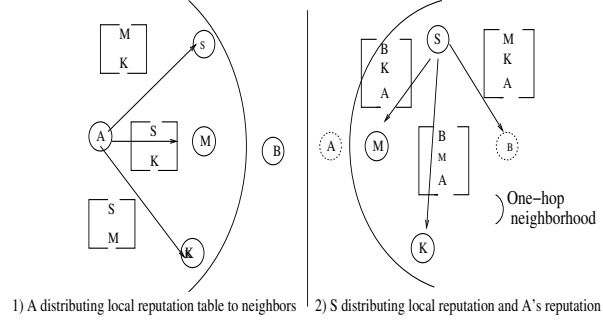Fig. 1: Neighbors monitoring a node

Fig. 2: Exchanging reputation tables

of the neighbors. The reputation table is shared with neighbors periodically (after time interval $T_{Report} + Random\ time$) as shown in Fig 2. In our future work, we will investigate the effects of distributing the reputation information to two-hop neighbors, in order to calculate a more accurate global reputation.

Weights are given to the neighbors reputation reports, and the weights are proportional to how much the neighbor is trusted. This is in turn proportional to the reputation of the neighbor. The node places greatest weight to its own observation, which is the upper threshold U_t. By providing weights to the global reputation calculation, the authenticity of the global reputation value increases. For any node i and j in the network, assume $Node\ j$ has calculated the reputation of $Node\ i$ as $Rep_{i,j}$, using Eqn 3. This reputation is recomputed based on the reputation reports about i obtained by j as shown in Eqn. 4.

$$
\begin{aligned}
Rep_{i,j} = Rep_{i,j} * (U_t/Sum_{(Rep)}) + \\
\sum_{m \in reporter, m \notin i} RR_{i,m} * Rep_{m,j}/Sum_{(Rep)} \\
Sum_{(Rep)} = \sum_{m \in reporter, m \notin i} (Rep_{m,j}) + U_t,
\end{aligned}
\tag{4}
$$

where $RR_{m,j}$ is the reputation report sent by neighbor node m to j.

By using the reputation distribution and re-calculation algorithm, all nodes have a consistent view of all its neighbors. This is illustrated in Section 7 via an implementation of the weighted reputation calculation algorithm.

4.2   Problems of Incorrect Monitoring

Certain problems have been identified with the monitoring scheme described in [7] including incorrectly penalizing nodes when packets are dropped due to congestion or collision and propagating false reports about a node. We will address the issue of dropping packets due to congestion and collision in our future work.

Propogation of false reputation report is an important concern for any reputation based cooperation scheme. A selfish node can send false reputation about other nodes and partition the network by claiming some nodes following it in the path are misbehaving. By assigning weights to reputation reports proportional to the reputation, the effect of a false report is significantly reduced. For example, in Fig 2, if A sends a false reputation report about M and A is selfish, its neighbors K and S maintain a low reputation value for A. Consequently, the weights given to A's reports are low and the false reports generated by A will not be significant.

This is further validated in our analysis presented in Section 7. However, this may lead to a different problem, where a node with a low reputation will not be considered truthful even if it is in fact propagating a correct report about a misbehaving node. In such cases, we assume that monitoring by other good nodes in the neighborhood will identify the misbehaving node.

5 Stimulating cooperation

5.1 Incentive strategy

Our scheme provides incentives for nodes that cooperate by prioritizing the data traffic with respect to the reputation of the source (and destination) node. IEEE 802.11e implements priority forwarding at the MAC layer based on the priority of the packet. In this solution, we assign priorities to packets based on the reputation of the source and destination, and the MAC layer performs per-packet priority forwarding.

An intermediate node may not have knowledge of the reputation of the source or the destination node, and may not be able to assign priority to the packet. The source thus needs to obtain a certificate of its good reputation, that can be presented to the intermediate nodes. We have designed a certificate generation algorithm that is truly decentralized and can be verified universally. The algorithm is secure and ensures that selfish nodes cannot obtain false certificates, and will be described in more detail in Section 6.

The source node uses the certificate generation algorithm to obtain the certificate and attaches it to the data packet. In certain routing algorithms that obtain the route based on the reply from the destination node ([1], [2]), the destination node can also obtain a certificate and send it along with the *Route Reply* packet. The priority for each packet is calculated based on the reputation of the source and destination and the MAC layer routes the packets based on their priorities. Note, by using the traffic priority as the incentive scheme, the tolerance of the network towards misbehavior varies with the network load. In other words, the magnitude of the reward increases when the network gets more congested. In a lightly loaded network, all nodes will get serviced irrespective of their reputation, unless their reputation is below the lower threshold L_t.

5.2 Punishment strategy

A node that does not cooperate in routing has a low reputation and is tolerated until reputation reaches a lower tolerance threshold L_t. A node whose reputation is below this threshold is said to be in the *Punishment* mode, and is punished as follows.

– A node in the punishment mode cannot send its own packets
– No packets are routed via a node in the punishment mode

We have only considered dropping data packets as a form of misbehavior. Selfish nodes may also deliberately include misleading information in route replies, so that a route containing the selfish node is not chosen for forwarding. Though our incentive strategy solves this problem to a certain extent, in our future work, we will develop other techniques to check routing anomalies and calculate the reputation accordingly.

5.3 Joining the network infrastructure after punishment

Since a non cooperative node is purged from all route paths, they cannot forward packets for others and consequently cannot improve their reputation with neighbors. Thus, a punished

node may never be able to able to participate in the network activities. We have designed two alternate protocols that can be used by a node in the *Punishment* mode, to rejoin the network infrastructure, namely the *Idle* and the *Redeem* protocol.

If a node chooses to follow the *Idle* protocol, it does not participate in any network activity for time period T_{Idle} and if the node is in the *Redeem* mode, the node is allowed to forward packets for others after a redeem period T_{redeem} but cannot forward its own packets until is reputation reaches the neutral reputation N_t.

6 Security

Two important security considerations in using the cooperation scheme are - 1) Prevent nodes from moving from one neighborhood to another to avoid punishment 2) Obtain tamper-proof certificates to certify the reputation of a node

6.1 Maintaining reputation across neighborhood

When using the reputation of a node to lock it out of network services such as routing, nodes can simply move out of one neighborhood to a region where their misbehavior has not been reported. Since the reputation of a node is only computed and distributed locally, the problem is more significant. Assume that a selfish node moves out to a new neighborhood. The algorithm to identify and punish the node is as follows.

1. If the selfish node is not known to any node in the new neighborhood, it is considered to be a new node and is assigned neutral reputation N_t
2. The selfish node is allowed to participate in the network infrastructure However, when it sends out its own packet, a *new* flag is set in the *Route Request* by its current neighbors.
3. If the *Route Request* with the *new* flag is cited by anyone in the selfish node's old neighborhoods, it raises an *Alarm* in the *Route Reply*. The *Alarm* contains the reputation information of the selfish node, including the punishment time
4. The new neighborhood obtains all such *Alarms* and chooses the one with maximum punishment time, and if the *Alarms* have equal punishment times, chooses the one with minimum reputation
5. The selfish node is assigned this reputation by its new neighbors

A selfish node is at a risk of being identified by the nodes in its previous neighborhood, and in a dynamic environment, it is highly likely that one of the node's previous neighbors would receive the *Route Request* packet with the *new* flag set. To avoid being more severely punished, a node with low reputation is encouraged to carry its certified reputation when it moves to a new neighborhood.

6.2 Certificate of reputation

Certificates are obtained by nodes to certify their reputation. The certificate may be used by a source node that needs to send its packets with higher priority. Certificates may also be used by a node with high reputation, which needs to carry its reputation when moving to a new neighborhood. In this solution, we use a distributed certification mechanism suitable for ad hoc networks, for obtaining certificates.

In our solution, the signing key or the private key of the central server is distributed to all nodes in the network using K-out-of-N threshold cryptography [11] such that all nodes have

Rounds	Node 1	Node 2	Node 3
1	47.5	49.3	49.2
2	45.8	48.4	48.2
3	44.6	47.4	47.1
4	43.4	46.3	46.0
5	42.3	45.2	44.8

Table 3: Weighted reputation of Node 0 when it drops Node 1's packets

Rounds	Node 2	Node 3
1	49.6	49.5
2	49.1	49.0
3	48.6	48.5
4	48.3	48.1

Table 4: Weighted reputation of Node 0 when falsely accused by Node 1

a key share. Any $K + 1$ nodes may combine their key shares to regain the private key, but no combination of K shares can regenerate the key. A node requests for a certificate of its reputation and the neighbors that are convinced of the node's reputation, sign a partial certificate using their key shares. $K + 1$ such partial certificate shares are combined by the node, to obtain its certificate.

We assume that a maximum of K selfish nodes reside in the neighborhood. K is an important system parameter and determines both the availability and the security of a solution. There needs to be at least $K + 1$ good nodes in the neighborhood, to ensure that a node requesting for a certificate can obtain one. However, we cannot assume that K is small, because this will affect the security of the solution. Selfish nodes cannot obtain a false certificate even if they collude, because there are a maximum of K selfish nodes in the neighborhood and thus cannot obtain $K + 1$ certificate shares. The certification generation scheme is similar in conception to that presented in [12].

7 Results

7.1 Reputation distribution

In this section, we will analyze the reputation distribution and weighted reputation calculation algorithm.

In Table 3, the weighted reputation of $Node$ 0, as calculated by $Nodes$ 1, 2 and 3 is tabulated. $Node$ 0 continuously drops packets it receives from $Node$ 1, and hence $Node$ 1 decreases its reputation by a factor of 5 after every round. The neutral reputation of all nodes is 50 and the reputation of the nodes do not change, except for $Node$ 0. As seen from Table 3, because of the use of weighted reputation, all nodes have a consistent view of $Node$ 0, i.e. the reputation of $Node$ 0 is reduced after every round by every neighbor, though $Node$ 0 misbehaves only with $Node$ 1.

Table 4 tabulates the reputation of $Node$ 0, as calculated by $Nodes$ 2 and 3. Assume that $Node$ 1 reports a false reputation value about $Node$ 0 and reduces its reputation by 10 points after every round. $Node$1 is a selfish node and has a low reputation of 30. From Table 4 it can be seen that, the false report by $Node$ 1 does not effect the reputation of $Node$ 0 significantly, because of the low reputation of $Node$ 1.

7.2 Simulation Setup

We have simulated the reputation scheme using GloMoSim [10] for proof-of-concept and analysis. The cooperation (or the reputation) scheme is implemented over the Dynamic Source Routing (DSR) [2] protocol in the routing layer. To implement the reputation scheme, all nodes are assigned a neutral reputation, $N_t = 50$, and an upper threshold for reputation, $U_t = 100$. The lower threshold L_t is set to 25 and α and β, the increment and decrement factors respectively,

are set to 2 and 5. The total area used for simulation is 1000 x 1000 m and the radio range is 250 m. The node placement is random. 802.11 is used in the MAC layer, the application used is CBR, with packet sizes varying from 64 B to 128 B. The simulation time is 40 minutes and there are 50 nodes in the network.

7.3   Throughput in the presence of malicious nodes

In a fairly high traffic network (with 30 CBR applications), we introduce compromised nodes that do not cooperate in forwarding packets. The compromised nodes are chosen at random. Fig 3 depicts the throughput of the non malicious nodes in the network with and without the reputation scheme. As the number of malicious nodes increases, the throughput decreases significantly, when using both the enhanced and the basic version of the DSR protocol. However, since packets are not routed via malicious nodes when it drops packets over a period of time, the throughput of the enhanced-DSR protocol is always higher than that of the basic version. It can also be observed, from Fig. 3 that the rate of decrease in throughput is increases more rapidly when using the basic version when compared to the enhanced version. This is because, as the number of selfish activities increase, more selfish nodes are identified and more routes are salvaged. When the percentage of malicious nodes is 30 % of the total nodes in the network, the throughput of the enhanced DSR is 50 % more than the basic version.

7.4   Overhead

The cost of the proposed reputation scheme in terms of communication overhead analyzed by studying the overhead in control packets. The reputation control packets are used for the cooperation protocol and includes reputation reports sent to neighbors, request for certificates and certificate replies. Fig. 4 provides the ratio of the reputation control packets to the total control packets for varying node density. It can be observed that the number of control packets sent specifically for the reputation scheme is less than .03 times the total number of control packets sent, even when the number of nodes in the network is 50. This ratio increases with increasing node density, because of the increase in reputation reports and certification services. It can also be observed that the overhead due to the reputation control packets is less when the network activity is higher. This is because, as the network activity increases, the number of reputation control packets is relatively constant, but the total number of control messages increases significantly. In our future work, the reputation scheme will be extensively analyzed using the simulation model.
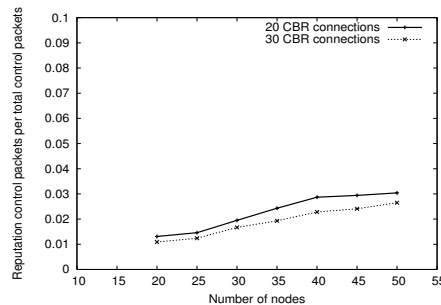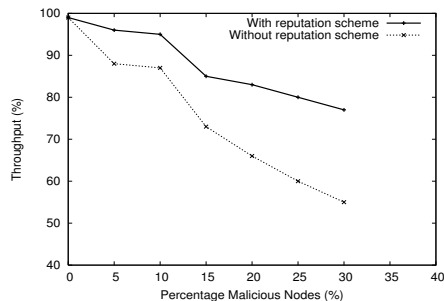

Fig. 3: Network throughput in the presence of malicious nodes

Fig. 4: Overhead of the cooperation scheme

8 Conclusion

This paper discussed a novel reputation-based scheme that provides both incentive and punishment for stimulating cooperation among ad hoc nodes. Incentive, in the form of higher priority during packet forwarding, is provided to well behaved nodes, to ensure continued cooperation. Selfish nodes that drop packets to conserve energy are identified and punished. Cooperation of a node is measured by its neighbors using a reputation metric and a node's reputation is calculated based on its behavior. The reputation system and the monitoring scheme used to observe the node behavior have been developed comprehensively and is well suited for the mobile ad hoc network domain. The solution is secure against malicious nodes that try to tamper with the cooperation scheme. Simulation results indicate that in the presence of non-cooperating nodes, the throughput when using the cooperation scheme is significantly higher than when no cooperation scheme is used. Results also indicate that the increase in control overhead due to the use of the cooperation scheme is minimal.

References

1. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, IEEE Computer Society (1999) 90
2. Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. In Imielinski, Korth, eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996)
3. Zhong, S., Chen, J., Yang, Y.R.: Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: Proceedings of IEEE INFOCOM '03, San Francesco, CA (2003)
4. Buttyan, L., Hubaux, J.P.: Stimulating cooperation in self-organizing mobile ad hoc networks. In: Mobility Networks and Application. (2003) 579–592
5. Buttyan, Hubaux: Enforcing service availability in mobile ad-hoc WANs. In: Proceedings of the First IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), Boston, MA, USA (2000)
6. Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks. In: Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, IEEE Computer Society (2002)
7. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. In: Mobile Computing and Networking. (2000) 255–265
8. Michiardi, P., Molva, R.: Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, Kluwer, B.V. (2002) 107–121
9. Yau, P.W., J.Mitchell, C.: Reputation methods for routing security for mobile ad hoc networks. In: Joint Workshop on Mobile Future and Symposium on Trends in Communications. (2003)
10. Zeng, X., Bagrodia, R., Gerla, M.: GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In: Workshop on Parallel and Distributed Simulation. (1998) 154–161
11. A.Shamir: How to Share a Secret. Communications of the ACM **22(11)** (1979) 612–613
12. Kong, J., Zerfos, P., Luo, H., Lu, S., Zhang, L.: Providing robust and ubiquitous security support for mobile ad hoc networks. In: Proceedings of the Ninth International Conference on Network Protocols (ICNP'01), IEEE Computer Society (2001) 251