

A Scalable and Robust Approach to Collaboration Enforcement in Mobile Ad-Hoc Networks

Ning Jiang, Kien A. Hua, and Danzhou Liu

Abstract: Mobile ad-hoc networks (MANETs) have attracted great research interest in recent years. Among many issues, lack of motivation for participating nodes to collaborate forms a major obstacle to the adoption of MANETs. Many contemporary collaboration enforcement techniques employ reputation mechanisms for nodes to avoid and penalize malicious participants. Reputation information is propagated among participants and updated based on complicated trust relationships to thwart false accusation of benign nodes. The aforementioned strategy suffers from low scalability and is likely to be exploited by adversaries. In this paper, we propose a novel approach to address these problems. With the proposed technique, no reputation information is propagated in the network and malicious nodes cannot cause false penalty to benign hosts. Nodes classify their one-hop neighbors through direct observation and misbehaving nodes are penalized within their localities. Data packets are dynamically rerouted to circumvent selfish nodes. As a result, overall network performance is greatly enhanced. This approach significantly simplifies the collaboration enforcement process, incurs low overhead, and is robust against various malicious behaviors. Simulation results based on different system configurations indicate that the proposed technique can significantly improve network performance with very low communication cost.

Index Terms: Cooperation enforcement, selfish node, wireless ad-hoc network.

I. INTRODUCTION

A mobile ad-hoc network (MANET) is a self-organized, highly distributed, and easy-to-configure network formed by devices (often referred to as nodes) equipped with wireless network interface cards. MANETs do not require dedicated infrastructure and can cope with nodes with different mobile capabilities, which opens a whole new avenue for various military, emergency, airport, and conferences applications.

The flexibility and proper operation of MANETs rely heavily on the collaboration of all participating nodes. Essentially, each node within a MANET is obligated to forward data for others. On the other hand, in many practical scenarios, nodes are restricted in power supply and are thus very sensitive to energy-swallowing operations such as packet forwarding. Obviously, the above two factors form a fundamental conflict, which motivates various selfish behaviors. It is acknowledged by many works [1]–[10] that selfish behaviors (mainly deliberately discarding packets of other nodes) can significantly de-

grade MANET performance. In fact, lack of motivation for participating nodes to collaborate is very likely to be the major obstacle to the adoption of mobile ad-hoc networks.

Many techniques have been proposed to enforce collaboration. Most of these techniques employ reputation mechanisms for nodes to avoid and penalize malicious participants. Reputation information is propagated among network participants and updated based on complicated trust relationships to thwart false accusation of benign nodes. The aforementioned strategy suffers from low scalability, high communication overhead, and is likely to be exploited by adversaries. In this paper, we propose a novel approach to address these drawbacks. With the proposed technique, no reputation information is propagated in the network and malicious nodes cannot cause false penalty to benign hosts. Nodes classify their one-hop neighbors through direct observation and misbehaving nodes are penalized within their localities. More importantly, data packets are dynamically rerouted to circumvent selfish nodes. As a result, overall network performance is greatly enhanced. This approach significantly simplifies the collaboration enforcement process, incurs low overhead, and is robust against various malicious behaviors. Simulation results based on different system configurations indicate that the proposed technique can significantly improve network performance.

The remainder of this paper is organized as follows. We discuss related works in Section II. In Section III, we introduce the selfish/malicious node detection and avoidance mechanism, and also present techniques to enforce the adaptive rerouting mechanism. Experimental results are given in Section IV. Finally, we conclude the paper in Section V.

II. RELATED WORK

In this section, we first present the existing collaboration enforcement techniques. We then introduce the dynamic source routing protocol [11] as background information.

A. Existing Collaboration Enforcement Techniques

The current state of the art in enforcing collaboration in mobile ad-hoc networks can be categorized into three groups, namely incentive motivation approaches, game theory based approaches, and misbehavior penalty approaches.

We consider incentive motivation techniques first. The authors of [12]–[14] proposed to use virtual currency to stimulate incentives for nodes to cooperate with each other. In their techniques, each node maintains a “wallet” of nuggets. A node has to possess enough nuggets to reward other nodes for relaying its packets. The only way a node can gather enough nuggets is to forward packets for other nodes. This technique relies on

Manuscript received October 8, 2005; approved for publication by Ekram Hossain, Division II Editor, June 4, 2006.

N. Jiang is with the Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA, email: njjiang@microsoft.com.

K. A. Hua and D. Liu are with the School of EECS, University of Central Florida, Orlando, FL 32816, USA, email: {kienhua, dzliu}@cs.ucf.edu.

a tamper proof security module and cryptographic techniques to prevent possible abuse of the protocol. The authors of [15] also propose a virtual currency technique that can stimulate node collaboration and defeat colluding malicious users without employment of the tamper resistant module. This scheme requires that each node reports a signature of each packet it forwards to a *central clearance service*. The practicability and performance of this approach remains unclear.

Another class of schemes [16]–[19] utilizes game theory [20] to model the cooperation enforcement problem in MANETs. Essentially, the purpose of these techniques is to derive strategies that consist of Nash equilibrium. Under the Nash equilibrium, no player (nodes) can benefit from violating the proposed strategy. In [16], a virtual currency approach based on a mechanism design technique [20] is presented. The goal of a mechanism design technique is to define a game played by independent agents according to the rules set by the mechanism designer such that the desired outcome, called the social optimum, can be achieved. This technique guarantees that nodes cannot gain anything through cheating during application data delivery. In [18] and [19], a similar technique is proposed to support multicast in MANETs. In [17], an algorithm based on the generous TIT-FOR-TAT (GTFT) strategy [21] is proposed. The authors prove the Nash equilibrium of the strategy. In general, the game theory based approaches assume nodes are rational (i.e., their behaviors are determined by their self interests) and are usually not robust to malicious participants (i.e., nodes willing to sacrifice their own benefits to cause devastating results to MANETs).

Our research, on the other hand, falls in the third category. The main idea is to detect, penalize, and avoid malicious and selfish hosts in MANETs. In [22], the authors use intrusion detection techniques to locate misbehaving nodes. A watchdog and a path rater approach is proposed in [23] to detect and circumvent selfish nodes. The main drawback of this approach is that it does not punish malicious nodes. This problem is addressed in [1], [2], [24], and [25]. The approach, called CONFIDANT, introduces a reputation system whereby each node keeps a list of the reputations of others. Malicious and selfish nodes are detected and reputation information is propagated to “friend” nodes, which update their reputation lists based on certain trust relationships. During route discovery, nodes try to avoid routes that contain nodes with bad reputations. Meanwhile, no data forwarding service is provided for low reputation nodes as a punishment. Another reputation-based technique, called CORE, is proposed in [26]. In this scheme, only positive reputations are disseminated. A formal analysis of CORE is given in [27]. A trust evaluation technique is proposed in [28]. In [29], the authors attack the problem of defending application data transmission against Byzantine errors (i.e., dropping, modifying, and rerouting packets). In their approach, each node maintains a weight list of other nodes. Malicious nodes are located by an on-demand detection process and their weights are increased consequently. A routing protocol is designed to select the least-weight path between two nodes. This approach is also based on per-node reputation lists. In addition, the detection process requires that each intermediate node transmit an acknowledgement packet to the source node. In [30], reputation information is propagated locally and one-way hash functions are employed

to secure reputation propagation. In [31], the authors propose an approach that does not assume any a priori trust relationship between nodes in MANETS. Each node has to obtain a token jointly issued by its neighbors in order to be admitted to the network. In [32], the authors propose to use “self-healing communities” to mitigate selfish nodes. The approach requires modification of underlying routing protocol and overhead to maintain the communities. In [33], a finite-state-model technique is introduced. The technique requires that nodes install tamper-proof modules. Reputation packets are only broadcast locally.

Table 1 offers a qualitative comparison of various collaboration enforcement techniques. In particular, most of existing detection and reaction techniques based on reputation dissemination mechanisms suffer from the following drawbacks:

- Reputation-propagation-based schemes have low scalability. Generally, quite a few reputation packets need to be propagated before “bad citizens” of MANETs can be captured, avoided, and punished. As a result, the cost of cooperation enforcement is quite high.
- Reputation-propagation-based schemes offer incentives to various attacks. Most prominently, malicious users can “oison” the reputation lists by disseminating incorrect reputation information. Such packets can be spoofed with other nodes’ addresses to hide the identity of the attacker or to pretend to be a “friend” of the receiver. In [29], digital signatures and message authentication codes [34] are employed to defeat packet spoofing. However, if a host is possessed (or physically captured) by a malicious user, cryptographic information of the particular node can be extracted and reputation poison attacks can still be mounted. In [25], the same authors of the CONFIDANT protocol present a scheme based on the Bayesian inference model to reduce false accusations. This scheme achieved significant reduction in false accusations for some types of reputation poisoning strategies but failed in some others. However, the scheme still relies on flooding reputation information and does not address the scalability concern.

We try to address the aforementioned problems by using only first-hand experience at each individual node instead of using second-hand reputation. This strategy is both effective and efficient.

B. DSR Overview

Before presenting the proposed technique, we briefly review the DSR protocol to make the paper self-contained. DSR consists of two phases: *Route discovery* and *route maintenance*. In the route discovery phase, the source node broadcasts a route request (RREQ) packet to all its neighbors if it does not have a route to the destination. Each neighboring node appends its address to the packet and broadcasts it if the node is not the intended destination and it has not seen the RREQ packet before. The route request is flooded in the network until it reaches a node that knows of a route to the destination. The node then originates a route reply (RREP) packet, which includes the complete route. The RREP packet will either be transmitted along the recorded route in reverse order or be included in a RREQ packet and broadcasted back to the source node. Generally,

Table 1. Comparison of collaboration enforcement techniques.

Technique	Scalability	Security	Require additional hardware
Incentive motivation approach	High	Low	Yes
Game theory based approach	High	Medium	Yes
Misbehavior penalty approach	Low	Low	No
Proposed approach	High	High	No

the source node may receive multiple routes, from which it selects the best one (by default, the shortest route) for data transmission. To reduce the number of route discoveries, each node maintains a route cache that stores all the routes it knows. Nodes obtain routes through route discovery, or by extracting paths from snooped RREP and data packets. In the route maintenance phase, when a node identifies a failed link, it sends a route error (RERR) packet to the source of the route, which removes the route from its cache. Furthermore, it tries to salvage the packet by looking for a route to the destination in its own route cache. In DSR, when a source node transmits data, it attaches the route to the destination node in each data packet. Intermediate nodes relay packets according to the embedded source routes.

III. THE PROPOSED SCHEME

In this section, we introduce the proposed techniques. Our approach is based on the following fundamental characteristics of MANETs:

- Each packet transmitted by a node A to a destination node more than one hop away must go through one of A 's neighboring nodes.
- A 's neighboring nodes can overhear its packet transmission.

Given a selfish node M , its un-collaborative behavior can be captured by most, if not all, of its neighboring nodes. Each of these nodes will then penalize M by rejecting all its packets. As a result, M will not be able to send any data to nodes more than one hop away. For a benign node B , if B is relaying packets for a source node S and is aware that the next hop node H is a selfish node, B can redirect the packets to avoid H . Note that the rerouting operation requires collaboration from B for S . We also present techniques to enforce such collaboration.

Fig. 1 illustrates the interaction between various components. Basically, the misbehaving node detection module is responsible for detecting un-collaborative nodes on behalf of the routing protocol. Once a misbehaving node is detected, it is reported to the routing module so that routes with the misbehaving node are purged. In addition, the adaptive rerouting module is informed to bypass the misbehaving node. Finally, the misbehaving node will be penalized by the misbehaving node penalty module.

A. Node Configuration

Our technique is based on nodes with the following configuration. First, nodes are equipped with omni-directional antennas and wireless interface cards that can be switched to promiscuous mode to "hear" data transmission in their proximities. Second, we base our discussion on the dynamic source routing protocol, as it is one of the most frequently used routing protocols

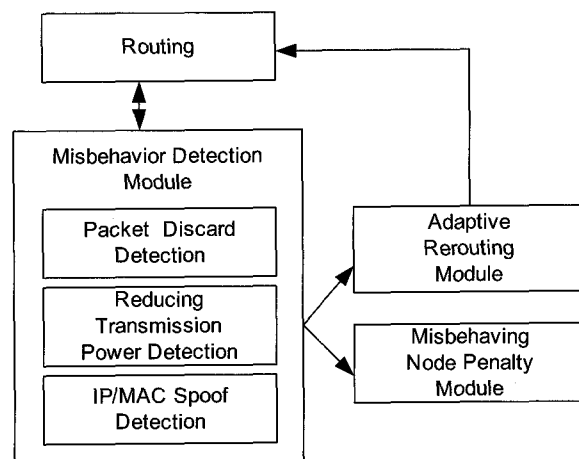


Fig. 1. Interaction between modules.

in the literature. The technique can be extended to accommodate other reactive routing protocols. Third, 802.11 [35], [36] is employed at the MAC layer. Finally, nodes have knowledge of their one-hop neighboring nodes. This can be achieved by either employing a HELLO protocol or by overhearing packets transmitted within the locality.

B. Selfish and Malicious Behaviors Considered

A selfish node can avoid the responsibility of forwarding data in two ways. First, by not participating in route discovery, a node can greatly reduce the chance of being selected to forward data packets by other nodes. Second, a selfish host can cooperate in route discovery, but subsequently discards data packets to save energy. We focus on the second type of selfishness in this paper as it is pointed out in [37] that such misbehavior has more negative impact on overall network throughput. In addition to selfish nodes, we are also concerned about malicious nodes in this paper. Our technique can also detect malicious nodes mounting denial of service attacks by disrupting link-level packet delivery. Only some of these problems have been studied in the literature [38]. Such attacks are immune to many existing collaboration enforcement techniques such as the watchdog module proposed in [23] and the CONFIDANT protocol presented in [1], [2], and [25].

C. The Detection Mechanism

In this section, we first present the detection mechanism. We illustrate through examples that the proposed detection mechanism can not only identify the second type of selfish behavior

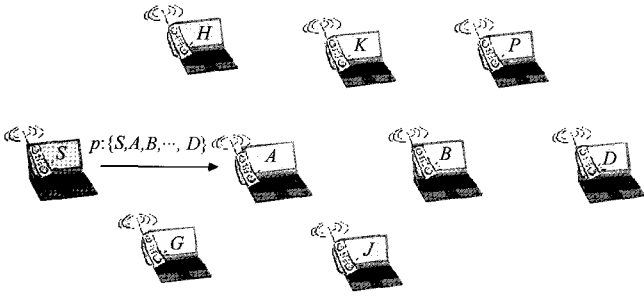


Fig. 2. Detection example.

(i.e., discarding data packets of other users), but also capture many malicious attacks.

C.1 Selfish Node Detection

Each node maintains a list of its neighboring nodes and tracks their actions. Nodes make no assumption of other hosts beyond their direct observable regions. We note that users are motivated to monitor their locality as they will benefit from identifying and circumventing selfish neighboring nodes. Furthermore, our detection mechanism fits naturally into DSR since in DSR nodes constantly sense the media and extract routes from overheard packets.

We present the detection mechanism through an example depicted in Fig. 2. It shows a node S transmitting data to a node D using a route $\{S, A, B, D\}$. Suppose node A is a selfish node and does not forward data packets to save energy. Assume nodes H and G are neighboring to both S and A , and nodes K and J are neighboring nodes of both A and B . Each node allocates a memory buffer to store packets transmitted by its neighboring nodes. Let us consider node S first. After S transmits a data packet to A , it 1) records the packet in its local buffer, 2) waits for a certain time interval, and 3) validates whether A has properly forwarded the packet by checking the memory buffer.

Whenever S observes a packet dropped by A (say, at time t), it checks a set Ω of all the packets it has transmitted through node A over a time window defined by $[t - W_{\text{UPPER}}, t - W_{\text{LOWER}}]$. If the cardinality of Ω is greater than a threshold T_{SUM} , S computes the packet drop ratio for node A on Ω . If this ratio is beyond a given threshold T_{SELFISH} , S tallies A as a selfish node; otherwise, S deems A as benign. The purpose of the W_{LOWER} parameter is to make a detecting node ignore packets dropped most recently (perhaps due to link breakage or unexpected network congestion). On the other hand, appropriate W_{UPPER} and T_{SUM} parameters ensure that a detecting node bases its decision on a large enough number of packets and a long enough timeframe. Essentially, selfish intention is sustained if and only if a node has been observed to drop a significant number of packets over a long enough timeframe. With this mechanism, our detection procedure can distinguish link breakage and temporary network congestion from deliberate packet discarding, and effectively reduces false classifications.

In our technique, data transmission is monitored by not only the source and intermediate nodes (i.e., nodes on the selected route), but also their neighboring nodes. Consider nodes G and H in Fig. 2. They, as neighboring nodes of S , overhear all data packets sent by S . Moreover, both G and H learn about the next

hop (A in this example) of each data packet p by extracting the source route option field of p 's IP header. As G and H are both neighboring to A , they will further detect whether A relays the packet using the aforementioned detection technique. In this example, both G and H will eventually identify A as a selfish node based on their own observations. On the other hand, although K and J are also neighbors of node A , they will not be able to detect A 's misbehavior since they have no access to the packets sent by the previous hop to A (S in this example). We refer to this scenario as “asymmetric sensing.” Our experimental results show that the effect of asymmetric sensing is limited. In most of cases selfish nodes suffer much lower performance than benign nodes.

C.2 Denial of Service Attack Detection

The above mechanism can also detect denial of service attacks mounted by malicious nodes using techniques discussed in [23]. In Fig. 2, a malicious node A does relay data packets. However, it either controls its transmission power to prevent data packets from reaching its next hop B , or intentionally causes collisions at node B to achieve the same effect. In either case, nodes S , G , and H will consider node A as a collaborative node whereas B never successfully receives any packet. The watchdog approach [23] fails under these situations. In our approach, however, nodes K and J can detect such attacks by examining the MAC-layer frames. In 802.11, the MAC layer of a node acknowledges the sender for each data frame successfully received. In our example, nodes K and J will not observe acknowledgement frames from B and will thus mark A as malicious instead of falsely accusing B . We note that malicious users can exploit this mechanism to cause false penalties. In Fig. 2, suppose node A is benign and node B is malicious. Node B intentionally refrains from acknowledging packets received from node A , hoping to trick neighboring nodes such as J and K to falsely recognize node A as a selfish node. A key observation to defeat such attacks is that a collaborative node A will retransmit the pending packets if it does not receive acknowledgements from B ; whereas no retransmission attempts will be made by a selfish node. Thus, by verifying whether a node conforms to the MAC layer protocol, we can successfully avoid false accusation of node A .

C.3 Collusion

We compare our technique with existing techniques regarding collusion robustness. In money-incentive models, significant effort needs to be invested (i.e., tamper-proof module) to prevent participants from gaining monetary benefit through colluding. In reputation-based schemes, colluding is attractive to both selfish and malicious users. On one hand, colluding selfish users can successfully cover each other and escape penalty. On the other hand, malicious participants can collaboratively cause various undesirable effects to benign users. In our technique, each node determines the reputation of its neighboring nodes through first-hand experiences, not through “rumor” or “propagated information.” As a result, colluding becomes much harder in this new environment.

C.4 IP/MAC Address Spoofing Detection

A more sophisticated malicious node might seek to spoof its own IP address and/or MAC address to impersonate a neighboring node to either bypass the detection mechanism or cause false penalty. Such address spoofing can be detected by considering the sequence control values of the MAC frames, as pointed out in [31]. The basic idea is that each node in the network keeps track of the MAC address and the sequence control field of all its neighboring nodes. In general, since adversaries are not able to compromise the firmware of network interface cards to manipulate the sequence control field, IP/MAC address spoofing can be successfully identified.

D. The Penalty Mechanism

Punishment of selfish/malicious nodes is achieved as follows. A node dedicates a *detection_time* field for each of its neighboring nodes. Suppose a node H identifies a selfish or malicious node A at time t . It records t in the *detection_time* field corresponding to A . Meanwhile, H keeps monitoring A and updates the *detection_time* field if A does not cease its misbehavior. H drops packets *originated* by A as a penalty. More specifically, H 's decision on whether to forward a data packet p for node A is based on

$$\Delta = t_p - A.\textit{detection_time}$$

where t_p is the time H receives p and $A.\textit{detection_time}$ is the *detection_time* field corresponding to node A on H . If Δ falls within a threshold defined as *penalty interval* τ , H will reject the packet. Consequently, the penalty will last as long as A continues to misbehave. In other words, the actual penalty time is proportional to the length of A 's misbehavior.

One concern of the penalty mechanism is that a benign node might be misclassified when it is penalizing its neighboring misbehaving nodes. We address the problem by slightly modifying the detection mechanism. In particular, a detecting node does not count packets dropped by its neighboring nodes due to selfish node penalty. In Fig. 2, suppose node A is a selfish node and it discards data packets from node S . As explained before, A will be detected by nodes S , H , and G . Consequently, node H and node G will not penalize node S when S rejects packets originated by node A and vice versa. Moreover, we recall that in the detection mechanism, a detecting node forms its decision based on a long enough time window and sufficient packet count. Since a benign node always relays packets for other (benign) nodes, it is unlikely that its packet drop ratio within a reasonable time window will exceed the T_{SELFISH} threshold. Therefore, the chance of false accusation is slim. Our experimental results also confirm that benign nodes in general do not suffer from false penalties.

E. Dynamic Redirection

In reputation-based mechanism, two scenarios will cause a source node to reroute data packets over a particular node. First, when an intermediate node detects a selfish or malicious node, it informs other nodes (including the source node of the session) through *reputation packets* so that they can choose a "clean"

route to circumvent the selfish node. Second, RERR packets are transmitted to the source node when broken links are encountered.¹ In both cases, source nodes are responsible for rerouting the data. In the proposed technique, we allow neither of the above packets to be propagated. An obvious question is: Who should reroute the data packets to bypass both irresponsible nodes and broken links?

Our solution is that each node shares the responsibility of rerouting packets. Again, we use Fig. 2 to illustrate the idea. We assume that node S is sending data to node D through a path $\{S, A, B, D\}$. Suppose the link between node A and node B is a *malfunction link* (i.e., either broken or node B is selfish). Without loss of generality, we assume that node B is a selfish node. After relaying a certain number of packets, node A will realize that B is a selfish node. We refer to node A as a *proxy* of source node S .² In our approach, A first purges all paths containing node B as an intermediate node from its route cache. Next, when A receives subsequent data packets from S , it broadcasts a route redirect (RRDIR) packet, indicating node B as a *bypassing target*. We note that a RRDIR packet serves as an indication of the beginning of the reroute process and the target node to be bypassed. It is by no means a reputation broadcast. In other words, neighboring nodes will not update their views of other nodes based on the RRDIR packets they receive. Continue the above discussion, the proxy node A then reroutes the packets by obtaining an alternative clean route to node D from its route cache. If such a route does not exist in its cache, A will buffer the data packets and instantiate a route discovery process to locate a clean path to D . In Fig. 2, A will discover a new clean route $\{K, P, D\}$, revise the embedded route of each data packet, and relay them to the destination. In this case, the actual route data packets traverse from S to D is $\{S, A, K, P, D\}$. It is possible for several proxy nodes to adaptively reroute data packets to avoid multiple selfish nodes along the chosen route. If A cannot find a route to D after a certain number of retries, it informs S through a RERR packet.

The proper functioning of the proposed selfish and malicious node circumvention scheme relies on the collaboration of proxy nodes. Unfortunately, proxy nodes can act maliciously to either avoid the reroute task or mount denial of service attacks. Continue the above example. When node A receives a data packet from S , it has the following options.

- Node A can mount a denial of service attack to S by deliberately forwarding packets to B even though it is aware that B is a selfish node. Nodes K and J will detect such attack as follows. First, both nodes will identify node B as a misbehaving node and they will assume that A has reached the same conclusion. Next, as A makes no effort to bypass B , both K and J will mark A as a malicious node and starts to penalize it.
- A does not reroute the packet and simply reports a RERR back to the source. In this case, all its neighboring nodes (S , G , H , J , and K) hear the RERR packets whereas none of them is aware of any route discovery attempt made by A .

¹Selfish nodes can falsely claim broken links in order to be excluded from packet transmission sessions.

²The proxy of a source node can be the source node itself when its next hop is selfish.

Thus, all of them will deem A as a selfish node.

- A broadcasts a RRDIR packet and then starts a route discovery process. Nevertheless, A reports a RERR to the source regardless of whether it receives RREP packets from the destination. The countermeasure we design involves utilizing some context information. After A sends a RREQ packet to look for a route to D , all its neighboring nodes will wait for the RREP packet to come back. Suppose node K relays the replying RREP packet to A and assume node H also hears the packet. Both H and K will expect to see node A transmit data to node D . However, as A sends a RERR packet, both nodes will recognize A as misbehaving. Furthermore, other neighboring nodes (S , G , and J) will deduct certain number of points for node A (say, equivalent to one third of those deducted for packet dropping). In other words, failure to reroute data packets is deemed as low-weight misbehavior. The purpose of this design is to discourage un-collaborative behavior. Benign nodes always relay data packets and will not suffer from such deduction.
- A broadcasts a RRDIR packet and reroutes data through a fabricated path. This attack has very limited effect in that benign nodes along the faked route will reroute the data packets and node A still has to relay data.

A last concern is that malicious nodes might attempt to disrupt data transmission by rerouting data packets. For instance, in Fig. 2, suppose A is a malicious node. When it receives a data packet from S that it should forward to a benign node B , it redirects the packet to a different (fabricated) route, hoping that other nodes along the redirected route will drop the packet. We note that this problem also exists in other schemes and is not introduced by our technique. More importantly, our technique facilitates the detection of such attacks. With our redirection mechanism, A has to broadcast a RRDIR packet to announce the rerouting operation. Otherwise, its neighboring nodes (S , H , and G) will identify it as a malicious node. In the RRDIR packet, A has to declare the correct next hop (B in this case) that it intends to bypass. Otherwise, it will be captured by S , H , and G . After receiving A 's RRDIR packet, node B will be aware of A 's attempt to deviate packets from a valid route and penalize A . Nodes K and J will also penalize A as they both recognize B as a benign node through their own observations. Finally, nodes that reroute packets for an excessive number of sessions within a certain time period will be considered as malicious and penalized by their neighbors.

IV. EXPERIMENTAL STUDY

We conducted various experiments to evaluate the effectiveness of the proposed technique in enforcing collaboration for MANETs. In this section, we first introduce the simulation setup and parameters. We then discuss the proposed technique based on various performance metrics.

A. Schemes Implemented

We implemented four schemes, namely the reference scheme, the defenseless scheme, the reputation-based scheme, and the proposed experience-based scheme, for performance evaluation. In the reference scheme, all the nodes act collaboratively and

relay data for each other. The defenseless scheme was implemented similar to those in [2] and [23]. A certain fraction of nodes are selfish as they promise to forward data for other nodes but fail to do so. In other words, these nodes forward routing packets, but discard any data packet not destined at them. No detection or prevention mechanism is implemented so that the network is totally “defenseless.” Next, we implemented a reputation-based system. In this scheme, each node maintains global reputation of other nodes. Nodes update reputation of others as follows. First, nodes monitor and form their opinion about the reputation of neighboring nodes using the same detection mechanism as presented in Section III-C. Nodes always trust their first-hand experiences with other nodes and ignore any reputation information against their own belief. Next, when a node detects a selfish node, it informs the source node of the communication session through a reputation packet. In response, the source node selects a “clean” route to transmit the remaining data if necessary. Nodes also update reputation of other nodes based on promiscuously learned reputation packets. Finally, each node periodically broadcasts reputation of other nodes in its locality. We implemented three types of nodes in this scheme, namely benign node, selfish node, and cheating node. A benign node always truthfully broadcasts the reputation information it has observed first hand, and honestly forwards the reputation information from neighboring nodes. A selfish node does not participate in data packet forwarding but cooperates in disseminating reputation information (i.e., it generates and relays reputation packets and never lies about other nodes). A cheating node relays both data and reputation packets for others. During reputation broadcast, however, it always lies about the reputation of nodes that it has direct experiences with. For all other nodes it is aware of, the cheating node simply reports them as selfish.

B. Simulation Setup

All the experiments were based on GlomoSim [32], a packet-level simulation package for wireless ad-hoc networks. The simulations were run on a Pentium-4 2.5 GHz PC with 1 GB of memory.

Our experiments were based on a MANET of 50 nodes within a 700×700 -square-meter 2-dimensional space. The simulation duration for each run was 10 minutes. All the nodes employ 802.11 [39] at the MAC layer. At the beginning of each simulation run, nodes were uniformly placed in the area. The random waypoint model was used to model host mobility. In this model, each node moves in a straight line towards a randomly selected destination location at a speed uniformly distributed between 0 m/s and some maximum speed. After the node reaches the destination location, it pauses for a specified period of time and then repeats the movement. In our experiments, the maximum speed of a node was limited to 20 m/s. We experimented with 0, 5, and 10 selfish nodes, accounting for 0%, 10%, and 20% of total number of nodes, respectively. Selfish nodes are randomly generated for all the simulation schemes. The number of selfish nodes is denoted as m . For each value of m , we tested two mobility scenarios, with pause times (denoted as p) of 120 second and 300 second, respectively. We employed the selfish node

Table 2. Fixed detection parameters.

Parameter	Value
T_{SELFISH}	0.8
Penalty interval τ	180 seconds
Detection buffer size	2 MB

Table 3. Simulation parameters.

Parameter	Value
Number of nodes	50
Area	700 m \times 700 m
Speed	Between 0 m/s and 20 m/s
Radio range	250 m
Placement	Uniform
Movement	Random waypoint model
MAC	802.11
Sending capacity	2 Mbps
Application	CBR
Number of applications	10
Simulation time	10 minutes

detection algorithm discussed in Section III-C for both the proposed scheme and the reputation-based scheme, with different W_{LOWER} and W_{UPPER} values. We picked 0, 4 seconds, and 8 seconds for W_{LOWER} and 15 seconds, 30 seconds, and 60 seconds for W_{UPPER} , resulting in a total of 9 different $[W_{\text{LOWER}}, W_{\text{UPPER}}]$ pairs. Each node allocates a buffer to store packets forwarded by its neighboring nodes in order to detect selfish nodes. A node can handle a maximum of 50 neighboring nodes and for each neighboring node a maximum of 20 packets are stored. The size of an 802.11 frame is limited to around 2 kB. Therefore, the size of the detection buffer is about 2 MB for each node. Table 2 lists parameters fixed throughout the experiments. We tested the reputation-based system with 0 and 5 randomly selected cheating nodes. In the experiments, the reputation broadcast interval was set to 10 seconds. Each configuration was executed under 5 different random seeds and the average values of the metric variables are reported. Constant bit rate (CBR) applications were used in this study. For each simulation run, we randomly generated a total of 10 CBR client/server sessions. In particular, we generated three selfish sessions (i.e., sessions originated by selfish nodes) and seven benign sessions (i.e., sessions started by benign nodes). The data packet size of each CBR session was chosen to be 552 bytes and packet transmission interval was set to 0.2 second. Table 3 lists all the simulation parameters.

C. Metrics

In the experiments, we evaluated the proposed scheme based on the following metrics:

- **Goodput of benign sessions (G_B):** For benign sessions, we denote the total number of bytes successfully received by CBR server applications as B_S and the overall bytes sent by CBR client applications as B_C . Then,

$$G_B = B_S / B_C.$$

This metric is a good indicator of the degree of collaboration among the nodes. Successful detection and circumvention of selfish nodes will result in significantly higher goodput.

- **Goodput of selfish sessions (G_S):** For malicious sessions, we denote the overall bytes sent by selfish source nodes as B'_C and the total number of bytes successfully received by the corresponding CBR server nodes as B'_S . Then,

$$G_S = B'_S / B'_C.$$

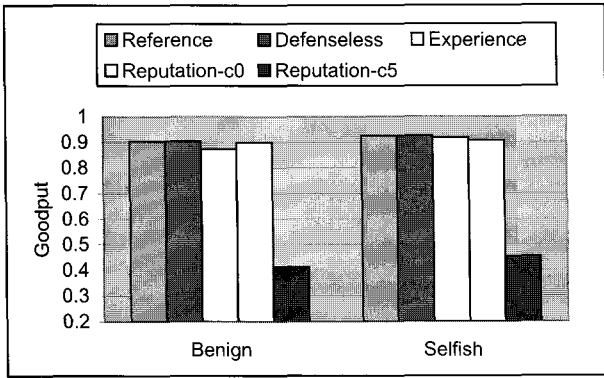
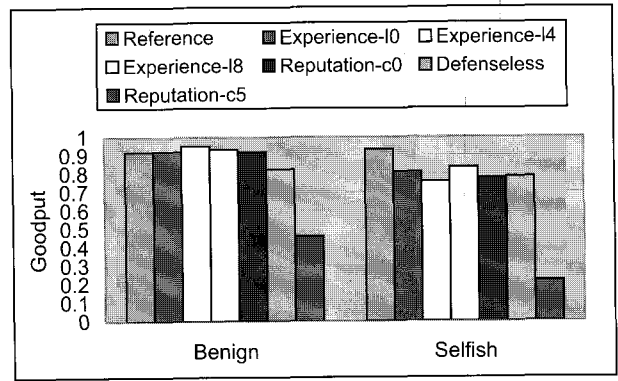
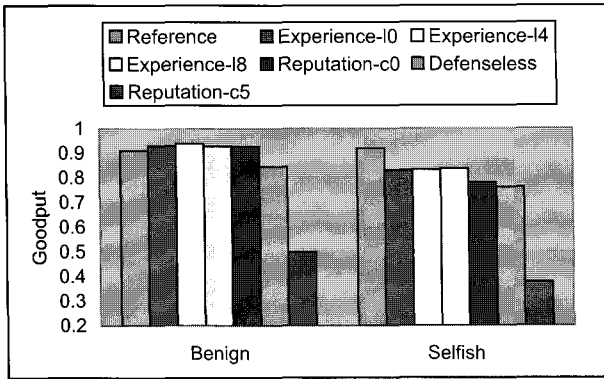
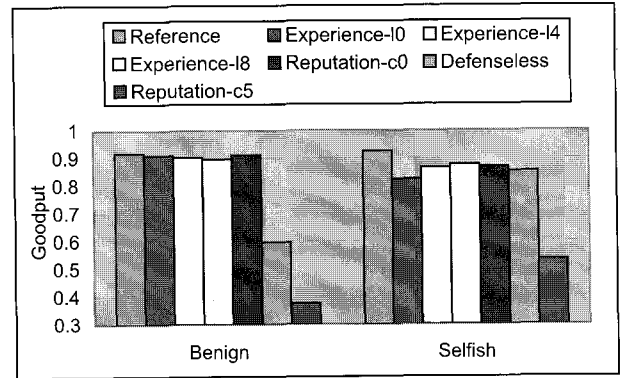
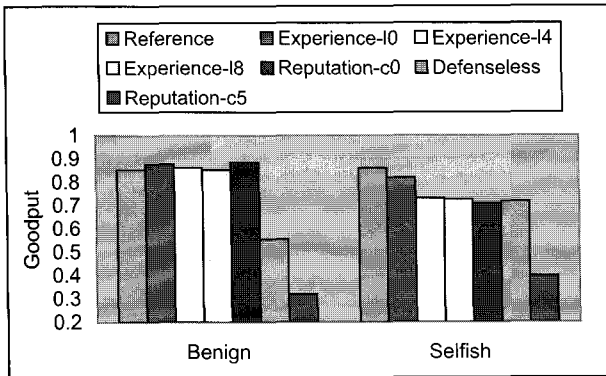
This measures the effectiveness of the proposed technique in terms of penalizing misbehaving nodes. A good collaboration enforcement technique should ensure a low G_S to discourage misbehaviors.

- **Communication cost:** The communication cost (hereafter also referred to as “cost” in short) of the proposed scheme O_E is calculated as the ratio between the number of all the control packets (i.e., RREQ, RREP, RERR, and RRDIR) originated and forwarded by nodes in the network and the total number of data packets successfully delivered to the destination nodes. More specifically, $O_E = \frac{C_E}{D_E}$, where C_E is the number of control packets originated and forwarded by nodes in the network and D_E is the number of data packets received by destination nodes. Similarly, the communication cost of the reputation-based scheme is computed as $O_R = \frac{C_R}{D_R}$, where C_R is the number of control packets (i.e., RREQ, RREP, RERR, and reputation packets³) originated and forwarded by nodes in the network, and D_R is the number of data packets successfully received by destination nodes. We note that the size of a data packet is generally much larger than the size of a control packet. Nevertheless, the ratio measures the average cost it takes the target scheme to successfully transmit a data packet.

D. Experimental Results

We present simulation results of various network configurations in this section. In the experiments, we set T_{SUM} to 8 (refer to Section III-C for the explanation of T_{SUM}). We study the impact of T_{SUM} in the next subsection. In all the experiments, we observe that in general, the goodput of both benign sessions and selfish sessions is not affected by W_{UPPER} whenever W_{LOWER} is fixed. This suggests that under all experimental scenarios, there are always enough packets falling in the detecting timeframe for nodes to detect selfish neighbors. Given this observation, in this section, we only present the average goodput of both benign and selfish sessions for a specific W_{LOWER} value for the proposed scheme. These results are always compared with the best performance result pair $\langle G_B, G_S \rangle$ achieved by the reputation-based scheme using the same detection mechanism, and under the same mobility pattern and number of selfish nodes. More specifically, for a particular configuration of the reputation-based scheme, a performance result pair $\langle G_B, G_S \rangle$ achieved under a $\langle W_{\text{LOWER}}, W_{\text{UPPER}} \rangle$ is considered better than another performance pair $\langle G'_B, G'_S \rangle$ achieved under another $\langle W'_{\text{LOWER}}, W'_{\text{UPPER}} \rangle$ iff $G_B - G_S > G'_B - G'_S$. Ties are broken by selecting a $\langle G_B, G_S \rangle$ pair with higher G_B .

³The reputation packets include packets originated by a node that detects a misbehaving node and periodical reputation broadcast transmitted by each node.

Fig. 3. Goodput when $m = 0$.Fig. 6. Goodput when $p = 300, m = 5$.Fig. 4. Goodput when $p = 120, m = 5$.Fig. 7. Goodput when $p = 300, m = 10$.Fig. 5. Goodput when $p = 120, m = 10$.

In all the figures, we refer to the proposed scheme as “Experience- lX ,” where X represents the W_{LOWER} value and the reputation-based scheme as “Reputation- cY ,” where Y indicates the number of cheating nodes.

D.1 Benign Session Goodput

Fig. 3 and the column “benign” of Figs. 4–7 depict the goodput of benign sessions when the number of selfish nodes is 0, 5, and 10.

Fig. 3 illustrates the results when there is no selfish node. For the proposed scheme, as results are very similar for all the $\langle W_{\text{LOWER}}, W_{\text{UPPER}} \rangle$ pairs, we only present the average results. We observe that the overall performance of the proposed

technique is very close to that of the fully collaborative network. This implies that the proposed approach incurs negligible overhead.

By employing the proposed scheme, significantly more data are successfully delivered to the destination nodes than the defenseless scheme since proxy nodes proactively detect and reroute data around misbehaving nodes. We can observe this effect in both Figs. 4 and 6, where there are 5 selfish nodes. The goodput of the experience-based scheme is always around 0.93 in both scenarios. The improvement over a defenseless network is about 12%. As another example, in Figs. 5 and 7, where there are 10 malicious nodes, the proposed technique lifted the goodput from around 0.6 in a defenseless network to higher than 0.85, an improvement of more than 40%. Moreover, the performance is similar under all W_{LOWER} values although $W_{\text{LOWER}} = 4$ achieved slightly higher goodput in most of the cases. In general, a lower W_{LOWER} will cause higher false penalties due to temporary link breakage whereas the detection algorithm with a larger W_{LOWER} tends to ignore many of the recently dropped packets and thus unnecessarily delays the reroute and penalty reaction. In addition, the high average goodput confirms that the benign nodes were in general experiencing almost no false accusation caused by penalizing misbehaving nodes, as explained in Section III-D.

We also notice from Figs. 4–6 that the goodput of benign sessions of both the proposed scheme and the liar free reputation-based scheme consistently exceeds the one in a totally collaborative network. Our explanations are as follows. In both ap-

proaches, data packets originated by selfish nodes are rejected by their benign neighbors as a penalty. Consequently, such benign neighboring nodes are left with more bandwidth to serve other well-behaved participants, thereby lifting the goodput of benign sessions.

We now compare the performance of the proposed technique with the reputation-based scheme. First, similar performance in terms of benign session goodput is observed for our technique and a liar free reputation-based system. This suggests that prompt packet reroute within the locality of intermediate nodes is in general as efficient as rerouting by source nodes. As a result, reputation propagation becomes unnecessary. In all the experiments, the reputation-based scheme suffered from significant performance loss (more than 50%) when only a few cheating nodes were present. In our simulation, as cheating nodes cooperate in data delivery, they will be deemed as benign nodes and their neighboring nodes will readily accept reputation advertisements from the cheating nodes provided that the recipients have no direct experience with the advertised nodes. As a result, the reputation mechanism was corrupted by inaccurate information and denial of service was experienced by most of the participants. The proposed experience-based approach has none of these problems and is therefore more robust in maintaining good performance.

D.2 Goodput of Selfish Sessions

We present the simulation results of goodput of selfish sessions in the “selfish” column of Figs. 4–7.

First, we observe that in most of the cases the goodput of selfish sessions for either the experience-based scheme or the liar free reputation-based scheme is higher than in a completely defenseless configuration. Such improvement is due to the fact that selfish nodes, while not recognized, also detect and actively avoid other uncooperative nodes and therefore also benefit from either the reroute functions in the case of experience-based technique or shared reputation information in the case of reputation-based method.

The experience-based scheme exhibited different behaviors under different W_{LOWER} settings. In general, $W_{\text{LOWER}} = 4$ performed better in terms of penalizing selfish participants as it more effectively detects selfish nodes.

In all cases, the goodput experienced by selfish users is lower than what collaborative users enjoy for the experience-based scheme. As an example, in Fig. 6, the goodput of benign sessions is higher than 0.93 (left column) as opposed to around 0.81 in the case of selfish sessions. Same phenomenon can be observed in other figures. Thus, selfishness will incur service downgrade and becomes less attractive.

In most of scenarios, the penalty capability of the liar free reputation-based scheme is slightly better than the experience-based approach, as selfish nodes become known to more participating nodes through reputation propagation. We now consider the case when a few cheating nodes exist in the reputation-based system. In practice, cheating nodes will most likely propagate negative reputation of others. As a result, liars actually contribute to the penalty of selfish nodes since the reputation they propagate with regard to selfish nodes is true. This effect is clearly presented in the experiments. However, such penalty is

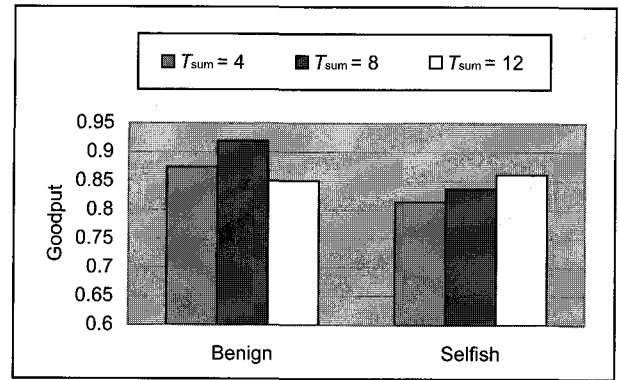


Fig. 8. $p = 120, m = 5$.

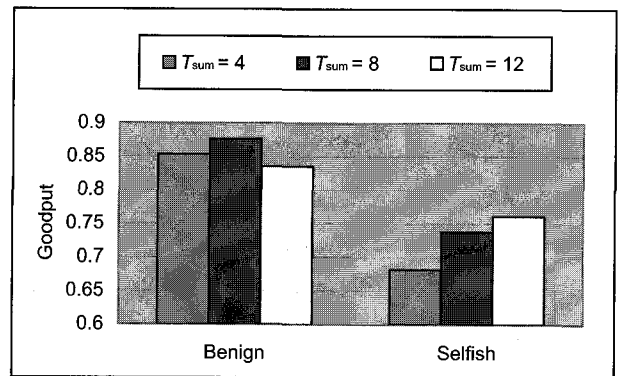
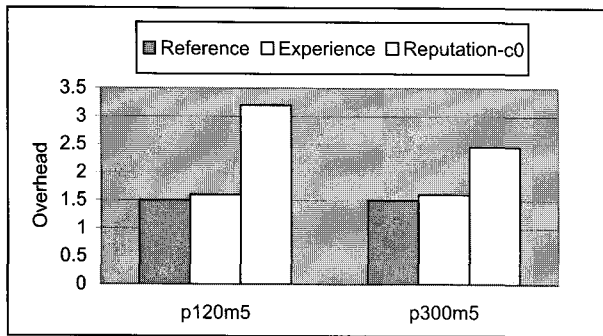
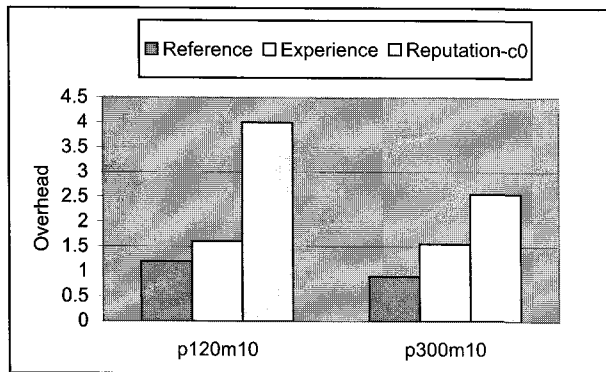


Fig. 9. $p = 120, m = 10$.

in the cost of benign nodes. As depicted in Figs. 4–7, the goodput of benign nodes is significantly hurt. We thus conclude that experience-based scheme is more suitable for MANETs due to its resilience to performance degradation caused by reputation poisoning behaviors.

D.3 The Impact of T_{SUM}

Figs. 8 and 9 illustrate the impact of the T_{SUM} parameter. Basically, T_{SUM} dictates how fast the detection mechanism reacts to packet loss. On one hand, lower T_{SUM} makes the detection mechanism more sensitive to packet drop. As a result, misbehaving nodes are captured quickly after they drop a few packets. However, some benign nodes might be mis-classified when they experience temporary link breakage. We observe in Figs. 8 and 9 that when $T_{\text{SUM}} = 4$, selfish nodes experience the lowest goodput. However, the goodput of benign nodes is lower compared to the results when T_{SUM} is set to 8 as some benign nodes are falsely penalized. On the other hand, higher T_{SUM} favors benign nodes at the cost of slower detection of misbehaving nodes. From Figs. 8 and 9, we observe that when $T_{\text{SUM}} = 12$, less penalty is imposed to selfish nodes. Benign nodes also suffer from reduced goodput since selfish nodes are not detected promptly. From the experiments, we observe that $T_{\text{SUM}} = 8$ seem to be an appropriate setting.

Fig. 10. Communication cost when $m = 5$.Fig. 11. Communication cost when $m = 10$.

D.4 Communication Cost

Figs. 10 and 11 illustrate the communication cost of the proposed scheme and the cheat free reputation-based scheme. For our approach, we show the results when $W_{\text{LOWER}} = 4$ and $W_{\text{UPPER}} = 60$. We also compare results of both schemes with the reference scheme.

From the figures, we can make the following observations. First, the cost of both our approach and the reputation-based scheme is higher than the cost of a completely attacker free environment. This is because both schemes are aware of misbehaving participants and proactively avoid such nodes, thereby incurring higher routing overhead; whereas the total number of successfully delivered data packets is similar. Second, the cost of the reputation-based mechanism is much higher than our scheme (higher than 67% in most cases). This is because our approach requires no reputation propagation; whereas the reputation-based scheme has to flood reputation information throughout the network. Although both schemes can achieve similar goodput for benign sessions (as illustrated by Figs. 3–7), our scheme is significantly more scalable and is thus more desirable for MANETs. Next, consider a fixed number of malicious nodes: The lower the node mobility, the lower the cost of both schemes. Obviously, when mobility is low, less routing packets are initiated. On the other hand, more packets are successfully delivered to the destination nodes, hence the lower communication cost. Finally, for a fixed mobility configuration, the higher the number of misbehaving nodes, the higher the communication cost. This also fits the intuition as nodes have to work more diligently when more un-collaborative participants are present.

V. CONCLUDING REMARKS

In mobile ad-hoc networks, there is no fixed infrastructure readily available to relay packets. Instead, nodes are obligated to cooperate in routing and forwarding packets. However, it might be advantageous for some nodes not to collaborate for reasons such as saving power and launching denial of service attacks. Therefore, enforcing collaboration is essential in mobile ad-hoc networks.

In most existing techniques, collaboration enforcement is achieved by a detect-and-react mechanism. In which, each node maintains global reputation of others in order to avoid and penalize misbehaving nodes. Propagation of reputation information is accomplished through complicated trust relationships. Such techniques incur scalability problems and are vulnerable to various reputation poisoning attacks.

In this paper, we propose a novel approach to enforcing collaboration and security in mobile ad-hoc networks. In our technique, nodes keep local reputation of their neighboring nodes through direct observation. No reputation advertisement is initiated or accepted. Nodes dynamically redirect data packets to avoid recognized adversaries. The redirect operation is also guarded against various evasive attempts. The advantages of this approach are many. First, since it does not rely on propagated reputation information, there is no need to maintain complex trust relationships. Second, since the misbehavior detection mechanism is based on first-hand experience at individual nodes, denial of service attacks are much more difficult to achieve. Colluding among nodes to secretly carry out fraudulent actions becomes much more difficult.

We conducted various experiments to investigate the effectiveness and efficiency of the proposed technique. Simulation results, based on GlomoSim, indicate that this technique is very effective in improving network performance. It also works well in disciplining defecting hosts. More importantly, the success of the proposed technique does not rely on reputation exchange and is thus both scalable and robust.

REFERENCES

- [1] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in wireless ad-hoc networks," in *Proc. IEEE INFOCOM*, 2003.
- [2] P. Michiardi and R. Molva, "Prevention of denial of service attacks and selfishness in mobile ad-hoc networks," Res. Rep. RR-02-63, Jan. 2002.
- [3] R. B. Myerson, *Game Theory: Analysis of Conflict*. Cambridge, Mass.: Harvard University Press, 1991.
- [4] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *Proc. ACM WiSe*, 2002.
- [5] R. Axelrod, *The Evolution of Cooperation*. New York: Basic Books, 1984.
- [6] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad-hoc networks," in *Proc. CNDS*, Jan. 2002.
- [7] Y. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad-hoc networks," in *Proc. WMCSA*, June 2002.
- [8] J. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad-hoc networks," in *Proc. IEEE/ACM MobiHoc*, 2001.
- [9] V. Kärpistö, "Security in ad-hoc networks," in *Proc. the Helsinki University of Technology, Seminar on Network Security*, 2000.
- [10] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *Proc. ACM WiSe*, 2004.
- [11] S. Buchegger and J. L. Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation of nodes—fairness in dynamic ad-hoc networks," in *Proc. IEEE/ACM MobiHoc*, June 2002.

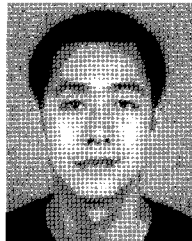
- [12] S. Buchegger and J. L. Boudec, "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad-hoc networks," in *Proc. the Tenth Euromicro Workshop on Parallel, Distributed, and Network-based Processing*, Jan. 2002, pp. 403–410.
- [13] S. Buchegger and J. L. Boudec, "IBM research report: The selfish node: Increasing routing security in mobile ad-hoc networks," Res. Rep. RR-3354, 2001.
- [14] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A library for parallel simulation of large-scale wireless networks," in *Proc. PADS*, May 26–29, 1998.
- [15] Y. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad-hoc networks," Dept. Computer Sci., Rice Univ., Tech. Rep. TR01-383, Dec. 2001.
- [16] S. Buchegger and J. L. Boudec, "Coping with false accusations in misbehavior reputation systems for mobile ad-hoc networks," EPFL, Tech. Rep. IC/2003/31.
- [17] H. Yang, X. Meng, and S. Lu, "Self-organized network-layer security in mobile ad-hoc networks," in *Proc. ACM WiSe*, 2002.
- [18] Information Technology Laboratory, National Institute of Standards and Technology, "The keyed-hash message authentication code (HMAC)."
- [19] P. Michiardi and R. Molva, "A game theoretical approach to evaluate cooperation enforcement mechanisms in mobile ad-hoc networks," in *Proc. WiOpt: Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks*, 2003.
- [20] P. Papadimitratos and Z. J. Haas, "Secure data transmission in mobile ad-hoc networks," in *Proc. ACM WiSe*, 2003.
- [21] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proc. IEEE INFOCOM*, 2003.
- [22] L. Zhou and Z. Haas, "Securing ad-hoc networks," *IEEE Network Mag., Special Issue on Networking Security*, vol. 13, no. 6, Nov./Dec., pp. 24–30, 1999.
- [23] N. B. Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson, "A charging and rewarding scheme for packet forwarding," in *Proc. IEEE/ACM MobiHoc*, 2003.
- [24] P. Michiardi and R. Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad-hoc networks," in *Proc. CMS*, 2002.
- [25] L. Buttyan and J. Hubaux, "Enforcing service availability in mobile ad-hoc WANS," in *Proc. IEEE/ACM MobiHoc*, Aug. 2000.
- [26] L. Buttyan and J. Hubaux, "Stimulating cooperation in self-organizing mobile ad-hoc networks," EPFL-DI-ICA, Tech. Rep. DSC/2001/046, Aug. 2001.
- [27] N. Jiang, S. Sheu, K. A. Hua, and O. Ozyer, "A finite-state-model scheme for efficient cooperation enforcement in mobile ad-hoc networks," in *Proc. ICPADS*, 2005.
- [28] S. Eidenbenz and L. Anderegg, "Ad-hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad-hoc networks with selfish agents," in *Proc. MobiCom*, Sept. 2003.
- [29] J. Kong, X. Hong, Y. Yi, J. S. Park, and M. Gerla, "A secure ad-hoc routing approach using localized self-healing communities," in *Proc. IEEE/ACM MobiHoc*, 2005.
- [30] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad-hoc networks," in *Proc. ICNP*, Nov. 2002.
- [31] ANSI/IEEE Standard 802.11 (1999). [Online]. Available: <http://standards.ieee.org/catalog/olis/lanman.html>
- [32] J. Wright, GCIH, and CCNA, "Detecting wireless LAN MAC address spoofing. [Online]. Available: <http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf>
- [33] I. S. Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification." IEEE 802.11 Standard, ISBN 1-55937-935-9, 1997.
- [34] P. Michiardi and R. Molva, "Simulation-based analysis of security exposures in mobile ad-hoc networks," in *Proc. European Wireless Conf.*, 2002.
- [35] E. M. Royer and C. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," *IEEE Pers. Commun.*, Apr. 1999.
- [36] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad-hoc networks," in *Proc. MobiCom*, 2000, pp. 255–265.
- [37] Q. He, D. Wu, P. Khosla, "SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proc. WCNC*, 2004.
- [38] D. Johnson and D. A. Maltz, "Dynamic source routing in ad-hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Ed., Dordrecht, The Netherlands: Kluwer Academic Publishers, 1996, pp. 153–181.
- [39] W. Zhao, X. Li, and Y. Wang, "Truthful multicast routing in selfish wireless networks," in *Proc. MobiCom*, 2004, pp. 245–259.
- [40] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proc. MobiCom*, 2000, pp. 275–283.
- [41] W. Zhao, X. Li, and Y. Wang, "Design multicast protocols for non-cooperative networks," in *Proc. IEEE INFOCOM*, 2005.



Ning Jiang received his B.S. degree of Computer Science from Shanghai Jiao Tong University, China in 1997 and his Ph.D. degree of Computer Science from University of Central Florida, USA in 2006. He joined the Mobile Information Worker Group of Microsoft Corporation after receiving his Ph.D. degree. His research interests include computer network security, data mining, and multimedia communications. He received the IEEE Orlando Regional Outstanding Graduate Student of the Year Award in 2003.



Kien A. Hua received the B.S. degree in Computer Science, M.S. and Ph.D. degrees in Electrical Engineering, all from the University of Illinois at Urbana-Champaign, in 1982, 1984, and 1987, respectively. From 1987 to 1990, he was with IBM Corporation. He joined the University of Central Florida in 1990, and is currently a professor in the School of Computer Science. Dr. Hua has published widely including three articles recognized as best papers and one as a top paper at international conferences. He has served as vice-chair, associate chair, demo chair, and program committee member for numerous ACM and IEEE conferences. Currently, he is on the editorial boards of the IEEE Transactions on Knowledge and Data Engineering and Journal of Multimedia Tools and Applications.



Danzhou Liu received the B.E. degree in information engineering from the Beijing University of Aeronautics and Astronautics in 1994, the M.E. degree in computer engineering from the Nanyang Technological University in 2002, and the M.S. degree in computer science from the University of Central Florida in 2005. He is currently a Ph.D. candidate in the Data System Group at the University of Central Florida. His research interests include multimedia retrieval, multimedia communications, and data mining and machine learning.