Mobile Netw Appl (2012) 17:342–352
DOI 10.1007/s11036-010-0243-9

Security Through Collaboration and Trust in MANETs

Wenjia Li · James Parker · Anupam Joshi

Published online: 22 June 2010

Abstract It is well understood that Mobile Ad Hoc Networks (MANETs) are extremely susceptible to a variety of attacks, and traditional security mechanisms do not work well. Many security schemes have been proposed that depend on cooperation amongst the nodes in a MANET for identifying nodes that are exhibiting malicious behaviors such as packet dropping, packet modification, and packet misrouting. We argue that in general, this problem can be viewed as an instance of detecting nodes whose behavior is an outlier when compared to others. In this paper, we propose a collaborative and trust-based outlier detection algorithm that factors in a node's reputation for MANETs. The algorithm leads to a common outlier view amongst distributed nodes with a limited communication overhead. Simulation results demonstrate that the proposed algorithm is efficient and accurate.

Keywords outlier detection · mobile ad hoc network · security · misbehavior · multi-dimensional trust

1 Introduction

A Mobile Ad-hoc NETwork (MANET), as its name implies, is normally composed of a dynamic set of cooperative nodes that are willing to relay packets for other nodes due to the lack of any pre-deployed network infrastructure. Mobile Ad-hoc NETworks (MANETs) have a variety of both civilian and military applications, ranging from emergency disaster rescue personnel coordinating efforts after a hurricane, earthquake or brush fire to soldiers exchanging information for situational awareness on the battlefield [1]. Other possible applications include personal and home area networking, real-time traffic alert propagation via vehicular networks, and Cyber Physical System (CPS).

Several factors make MANETs extremely susceptible to various misbehaviors such as intrusions [2], greyholes [3], and blackholes [4]. First of all, data in MANETs is transmitted via Radio Frequency (RF) broadcasts, which can be easily eavesdropped on or even modified. Second, nodes in MANETs have limited power supply, and consequently their performance is severely degraded when power is exhausted. Third, when they are used for security and military purposes, nodes in MANETs are vulnerable to compromise and manipulation by adversaries. Hence, it is obvious that misbehavior detection should be an indispensable component of any security solution that aims to safeguard the mobile ad hoc networks. The misbehavior typically observed includes dropping of packets, misroutes, false Requests/Clears in the MAC layer etc. However, many of these events can also happen due to environmental and mobility related reasons, not just malicious intent. Most of the current misbehavior detection mechanisms rely on a predefined threshold to decide if a node's behavior is malicious or not. However, it is rather difficult to set an appropriate threshold because the network is quite dynamic and unpredictable, and environmental conditions such as ambient RF noise can vary. In contrast, we do not need to rely on any previous knowledge to find a node that is an outlier with respect to a given observable. Given the fact that a malicious node generally behaves differently when compared to other nodes, we can detect the node misbehaviors by means of outlier detection.

This paper is an extended version of our previous paper that was accepted by CollaborateCom 2008.

W. Li (✉) · J. Parker · A. Joshi
Department of Computer Science and Electrical Engineering,
University of Maryland, Baltimore County (UMBC),
1000 Hilltop Circle,
Baltimore, MD 21250, USA
e-mail: wenjia1@cs.umbc.edu

J. Parker
e-mail: jparke2@cs.umbc.edu

A. Joshi
e-mail: joshi@cs.umbc.edu

Besides misbehavior detection, trust management is another well-studied method that can be used to secure MANETs. The main purpose of trust management is to evaluate the behaviors of other nodes, and thus build a reputation for each node based on the result of behavioral assessment. Most of the trust management schemes in MANETs model the trustworthiness of a node in one *dimension*, i.e., all observations are used to calculate a single scalar trust for each node. However, a single trust metric may not be expressive enough to adequately describe whether a node is trustworthy or not in many complicated scenarios. Figure 1 illustrates an example scenario in which a single scalar trust is not expressive enough.

In this example scenario, the observer observes and records the same amount of Request-To-Send (RTS) flooding, packet modification and fake opinion spreading behaviors for node 1, 2, and 3 in the first stage. Assume that these three misbehaviors are punished at the same rate, and a single scalar trust is evaluated for these nodes. Then, all these three nodes are equally trustworthy from the observer's perspective. Nevertheless, it is obvious that node 2 is less trustworthy than the other two nodes for relaying packets; whereas the opinions from node 3 should be questioned more because it is more likely to spread rumors. Therefore, a single trust scheme is neither accurate nor effective in the complicated scenarios.

In this paper, we propose and evaluate a collaborative, multi-dimensional-trust based outlier detection mechanism for mobile ad hoc networks. Compared to the traditional *single*-dimensional trust management mechanisms such as those that have been discussed in [30–33], the trustworthiness of a node is judged from different *perspectives* (i.e., *dimensions*), and each dimension of the trustworthiness is derived from various sets of misbehaviors according to the nature of those misbehaviors. In our approach, as in many others [19, 21, 25, 29], all the nodes in MANETs first observe and record a group of abnormal behaviors that are conducted by their neighbours. Unlike most of the existing approaches, each node then locally detects the misbehaviors and infers the trust (in terms of different dimensions) of its neighbors from its own observations in the second step. Next, the local observations are exchanged amongst the neighbouring nodes, and the local views of misbehavior as well as trust will be updated accordingly only when some brand new observations are offered by a trustworthy neighbour. The observation exchange process will last until there is not any view update for all the nodes.

The major contribution of this paper is to explore how misbehaviors can be correctly identified by means of outlier detection, and how trustworthiness of each node can be properly evaluated via multi-dimensional trust management. More specifically, the key novel features of our proposed mechanism are: (1) the misbehavior detection scheme by means of outlier detection, in which neither pre-deployment training procedure nor pre-defined threshold for node misbehaviors will be required; (2) the multi-dimensional trust management scheme in which the notion of trustworthiness is further divided into several attributes (i.e., dimensions) so that each attribute is able to precisely indicate whether or not a node is trustworthy in terms of one specific feature that its behavior should have, such as cooperative, well-behaved, and honest.

2 Related work

2.1 Outlier detection

Outlier detection is a hot topic in the data mining research, and various definitions of outliers have been proposed in the literature. Outliers are generally defined as data points that are very different from the rest of the data with respect to some measure [5]. Outlier detection can be used for two purposes: either to eliminate outliers and thus potentially reduce the; or to expose the outliers for further analysis, such as in intrusion detection [6, 7], fraud analysis [8] and habitat monitoring for endangered species [9]. Our proposed algorithm takes two popular distance-based definitions into account: (1) distance to the nearest neighbour (*NN*), and (2) average distance to k nearest neighbors (k-*NN*) [10].

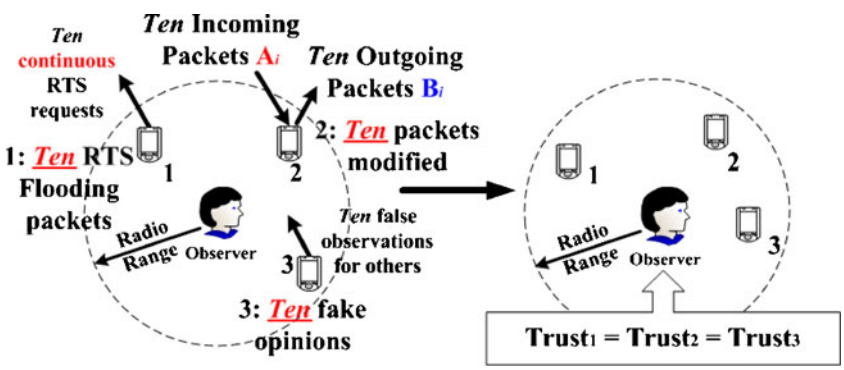

Fig. 1 An example where a single scalar trust is NOT expressive enough

One major motivation of outlier detection research is to efficiently identify outliers in a large-scale database [10–12]. Nevertheless, the situation in mobile ad hoc networks is significantly different from that in large-scale central databases: in mobile ad hoc networks, data are generated and stored in scattered nodes and transmitted via wireless channels, which are unreliable and bandwidth and power-constrained. Outlier detection methods for the large-scale databases cannot be directly applied to mobile ad hoc networks because they will cause a large communication overhead.

Several outlier detection algorithms have been recently proposed for wireless sensor networks (WSNs) [9, 16–18]. Branch et al. [16] propose an in-network outlier detection scheme to detect the outliers based on data exchange among neighbors. Our outlier detection mechanism is somewhat similar to the method proposed by Branch et al. However, there are two significant differences between the two methods. First, the method by Branch et al. does not consider the mobility of the nodes, whereas our proposed method takes the mobility issue in consideration. Second, malicious behaviors are not considered in their approach, i.e., the nodes will not deliberately fabricate fake local views or alter incoming local views in their method. On the contrary, we have considered the malicious behaviors of the nodes, and applied the knowledge of trust and reputation as the countermeasure to the malicious behaviors.

2.2 Misbehavior detection and trust management in MANETs

In mobile ad hoc networks, all network operations such as routing and forwarding rely on cooperation of the nodes because there is no centralized infrastructure. Hence, if some nodes choose not to participate in the network operations, then these network services may be incomplete or even unavailable. These non-cooperative nodes are generally called *selfish nodes* [19]. Besides selfishness, ad hoc network misbehavior may also be conducted by *malicious nodes*, which aim to harm the whole ad hoc networks. A malicious node can perform different attacks to either compromise individual node(s) or degrade the performance of the overall network [20]. The existence of selfishness and malicious behaviors has motivated research in the area of misbehavior detection as well as trust management for mobile ad hoc networks.

Intrusion Detection Systems (IDS) are an important means to detect node misbehavior. Several mechanisms have been proposed to build IDS on individual nodes due to the lack of a centralized infrastructure [21–24]. In these mechanisms, every node is equipped with an IDS, and each IDS is assumed to be always on, which is not energy-efficient given the limited battery power of nodes in ad hoc networks. On the other hand, Huang et al. [25] propose a cooperative intrusion detection framework in which clusters are formed in ad hoc networks and all the nodes in one single cluster will cooperate in intrusion detection operation. There are also some other proposed solutions that aim to cope with various routing misbehaviors [19, 26, 27].

The majority of these misbehavior detection mechanisms either rely on pre-labelled misbehaviors to train the classifier, or a threshold needs to be set prior to the detection procedure in order to distinct misbehaving nodes from normal nodes. However, it is generally difficult to predict the behavioral model of the misbehaving nodes. Hence, outlier detection is a feasible solution to detect node misbehaviors in lack of priori knowledge for node misbehaviors as well as in presence of novel attack models.

There are numerous trust and reputation management schemes that have been proposed in the past decades [30–33]. Most of the current trust management schemes model the trustworthiness of an object in one *dimension*. For instance, in our previous work [13–15], a single scalar trust is used to evaluate the trustworthiness of each node in MANETs. As we have discussed in the previous section, a single scalar trust cannot precisely reflect the trustworthiness of a node in different contexts. Therefore, we extend our previous work in that the multi-dimensional trust management scheme is used to assess the trustworthiness of nodes from several separate perspectives.

3 Gossip-based outlier detection algorithm

In this section, we describe our gossip-based distributed outlier detection algorithm. The goal of the algorithm is to find the top k outliers in terms of some observed behaviors (such as packet drops or misroutes) from all the nodes in mobile ad hoc networks (Here k is a user-defined parameter). The algorithm leads to a consistent global view of the top k outliers in all the nodes as long as these nodes do not change their behavior significantly during the convergence time of the algorithm. By using constrained gossiping, the algorithm avoids flooding the network.

3.1 Algorithm description

The proposed outlier detection algorithm contains the following four steps, namely local view formation, local view exchange, local view update, and global view formation. We have adopted two local view update methods in our algorithm: one is the simple averaging method, in which all the local views are merged by simply averaging them; the other method is the trust-based weighted method, in which the local views are merged incorporating the trust in other nodes.

The first step of our algorithm is the formation of local views. The nodes monitor and record the possible malicious behaviors

of other nodes within their radio range. Each node generates its local view of outliers based on their own observations.

Once all the nodes form their local views, they will broadcast the local views to all of their immediate neighbors, i.e., all the nodes that are one hop away from them. Upon reception of a local view from another node, the recipient will update its local view based on the received view. The first local view update method we employ is the simple averaging method, which is shown in the Subroutine 1 below. Here n_i denotes the i-th node in the mobile ad hoc networks. V_i denotes the initial view of n_i. V_i' denotes the updated view of n_i.

Subroutine 1 Update of Local View for node *i* Using the *Simple Averaging* Method

Input of n_i: V_i

Output of n_i: V_i'

Upon reception of V_j from node n_j:

if $V_j \neq V_i$

—merge the V_i and V_j according to the following rules:

——if node m is in BOTH V_i AND V_j, then calculate the average of the corresponding columns for node m in BOTH V_i and V_j, and store the average of node m to an intermediate list $TEMP_i$ as an entry.

——if node m is in EITHER V_i OR V_j, but NOT BOTH, then add a virtual entry of node m to the view that previously does not contain m, and set all the columns of this virtual entry as 0. Then calculate the average between the true entry of m and virtual entry of m for each column, and then store the average values of node m to an intermediate list $TEMP_i$ as an entry.

—calculate the top k outliers from $TEMP_i$, and assign these k top outliers to V_i'.

—broadcast V_i' to all of its immediate neighbors (number of hop = 1).

else keep V_i unchanged, and not send any message out

The averaging is necessary due to the existence of malicious nodes that may produce false views to mislead other nodes. Suppose a malicious node randomly generates some entries reporting large misbehaviors for a good node, and sends this false view to others. If the recipients simply take the false view it will miss the true outliers. Averaging the information from all neighbors helps avoid this. Another heuristic is that if a recipient receives information about any node that has never been seen before, it will use only half of the reported value in computing the average. In other words, it will treat this new information conservatively. On the other hand, the true outliers will not be influenced by either of the heuristics because several nodes will report their observed outlier values. Of course, this scheme will be vulnerable in a locality where most of the nodes are malicious, but in such circumstances most misbehavior detection algorithms fail anyway.

Another possibility is to use the trust-based weighted method during the local view update process. Unlike the simple averaging method, the trust-based weighted method relies on the reputation of a node to determine how to merge the view it sends out with the local view of the receiver. The trust-based weighted method is listed in the Subroutine 2 below. Again, n_i denotes the i-th node in the mobile ad hoc networks. V_i denotes the initial view of n_i. V_i' denotes the updated view of n_i. w_{ik} denotes the weight of local view sent from node k to node i. We should also note that the trust value can be derived by two different trust management schemes. The first trust management scheme, namely the simple trust management scheme, views the trustworthiness of each node as a scalar [13]. In contrast, the other trust management scheme attempts to determine the trustworthiness of the node from multiple perspectives. We will further discuss these two trust management schemes in Section 3.2.

Subroutine 2 Update of Local View for node *i* using the *Trust-based Weighted* Voting Method

Input of n_i: V_i

Output of n_i: V_i'

Upon reception of V_k from node n_k:

if $V_j \neq V_k$

—merge the V_i and V_k according to the following rule:

——if node m is in BOTH V_i AND V_k, then calculate the weighted average WA_i of the corresponding columns for node m in BOTH V_i and V_k according to the following formula:

$$WA_i = \left(w_{ii} * m_i + w_{ik} * m_k\right) / \left(w_{ii} + w_{ik}\right)$$

and then store the weighted average WA_i of node m to an intermediate list $TEMP_i$ as an entry.

——if node m is in EITHER V_i OR V_k, but NOT BOTH, then we simply set m_i or m_k to be zero, and the calculation of WA_i follows the formulae below:

$$WA_i = \begin{cases} w_{ii} * m_i, & \text{when } m_k = 0 \\ w_{ik} * m_k, & \text{when } m_i = 0 \end{cases}$$

and then store the weighted average WA_i of node m to an intermediate list $TEMP_i$ as an entry.

—calculate the top k outliers from $TEMP_i$, and assign these k top outliers to V_i'.

—broadcast V_i' to all of its immediate neighbors (number of hop = 1).

else keep V_i unchanged, and not send any message out

Note that unlike traditional gossiping, the more the nodes that accept the same view of outliers, the less the number of new messages that are sent out. Ultimately, when all the nodes hold the same view of outliers, the algorithm will halt, and the view that all the nodes hold is regarded as the global view of outliers.

The pseudo-code of the algorithm is given in Algorithm 1 and uses the same notation as described earlier. In addition, *GV* denotes the ultimate global view.

Algorithm 1 Gossip-based Outlier Detection

Input of n_i: V_i

Output of n_i: *GV*

For each node n_i:

broadcast V_i to all of its immediate neighbors

Upon reception of V_j from node n_j:

invoke **Subroutine1** ***OR*** **Subroutine 2**

When no more message exchange occurs:

$\forall k$, GV = V_k

3.2 Multi-dimensional trust establishment and management

In this section, we describe the multi-dimensional trust management scheme, in which *multi-dimensional trust* is used to evaluate the trustworthiness of the peer from multiple perspectives. The term *multi-dimensional trust* has been used to assess the trustworthiness of the agents in the multi-agent systems [34, 35]. In these trust management schemes, the term *dimension* refers to factors such as quality, timeliness and cost, in selecting a cooperative partner. On the other hand, in our case, the term *dimension* is used to express the different perspectives by which a peer is assessed.

In our proposed multi-dimensional trust scheme, the trustworthiness of a peer is evaluated from three perspectives, namely *Collaboration Trust* (COLT), *Behavioral Trust* (BET), and *Reference Trust* (RET), respectively. The multi-dimensional trust scheme is shown in Fig. 2.

From Fig. 2 we see that COLT is determined by the *peer collaboration degree*, which is defined by how collaborative a peer is when it is asked to participate in some network activities such as route discovery or packet forwarding. BET is derived by the degree of abnormal behavior that a peer has shown, including packet modification, packet misroute or RTS flooding attack. RET is generally computed based on the correctness of the opinion that a peer gives for other nodes. For example, if a peer has been witnessed giving fake reports for its neighbouring

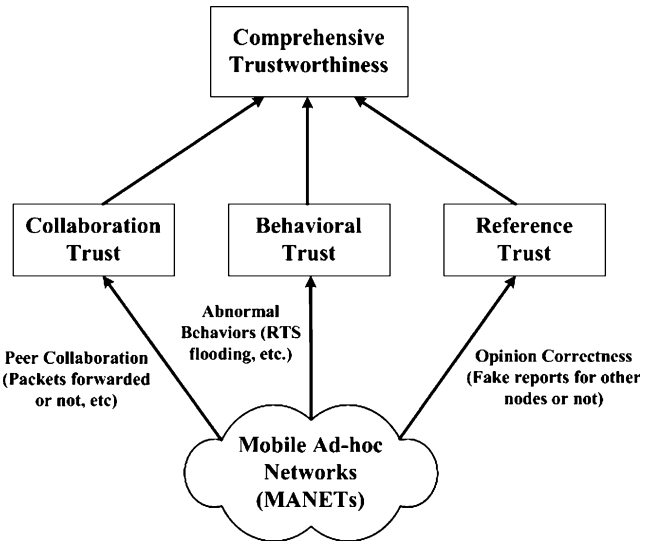

Fig. 2 Multi-dimensional trust scheme

nodes, the reference trust of this peer should be assigned a low value. In this way, other peers can properly interpret the opinions given by this peer by using its reference trust in the future.

More specifically, all the trust values are initialized to be 1. Whenever the node observes any evidence (such as packet dropping, packet modification, or fake opinion for its neighbour) of its neighbors, the node reduces the corresponding trust value of its neighbour according to the *punishment factors*. We set different punishment factors for different observed behaviors when we adjust the trust value. For example, packet dropping and packet modification are both misbehaviors. However, packet dropping may be caused either by intentional malicious behavior or by power failure. On the other hand, when we find that a node is modifying the incoming packets, we can safely conclude that it is malicious. Hence, we set a higher punishment factor for packet modification than packet dropping.

During the local view update process, when a peer *j* gets local view from its neighbour *k*, the peer will use the RET value of its neighbour as the weight w_{ik}, and its own weight w_{ii} will always be 1. The COLT, BET and RET values will also be updated accordingly during the local view update step using the weighted voting method. In this way, we apply the knowledge of trust and reputation to the local view update process, and we can ensure that the fake local views spread by the malicious peers will not influence the formation of the global view.

On the other hand, the COLT value of a peer can be used to determine if a node should be included in the network activities, such as packet forwarding. The BET value is generally an indicator by which we can tell how a peer behaves, and it can be used to tell if a peer is malicious or not.

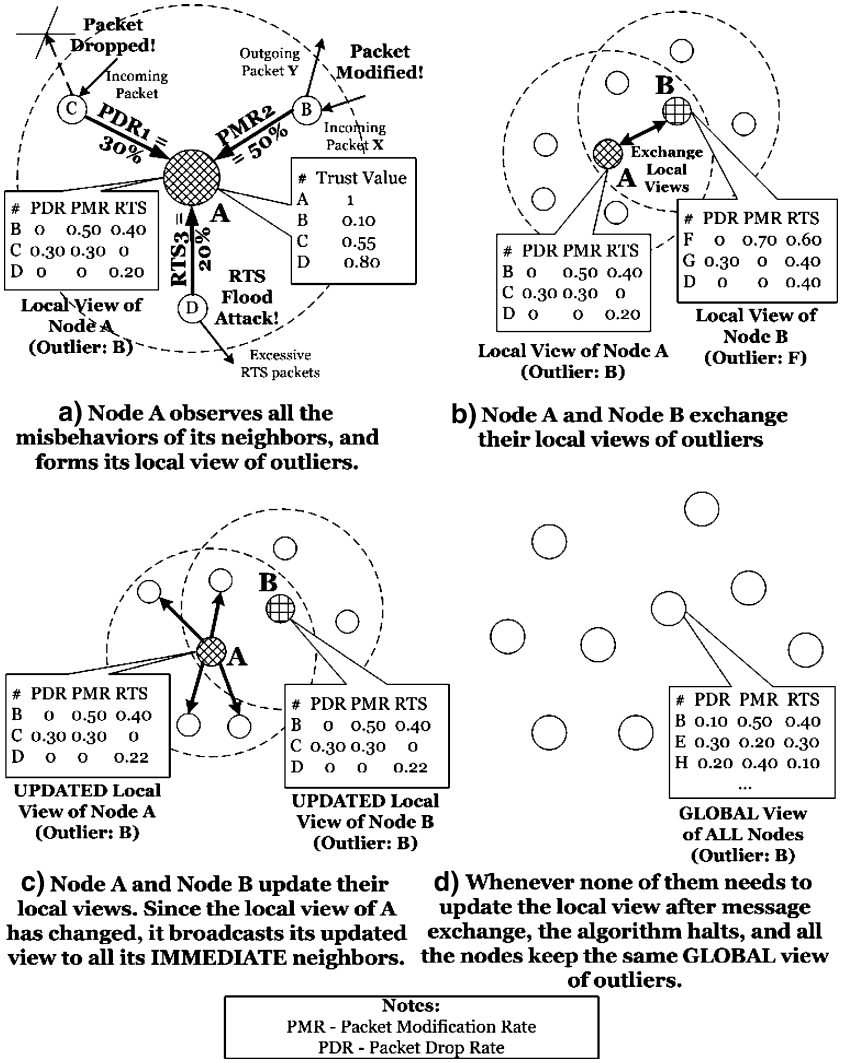

Fig. 3 An example scenario of the trust-based outlier detection algorithm

3.3 An example scenario

To help better understand the proposed algorithm, an example is presented in Fig. 3. In Fig. 3a, node A observes all the misbehaviors of its neighbors, and then forms its local view based on its own observation. Node A will also construct its initial trust table based on its observation to its neighbors. Note that the local view is composed of a list of neighbors and the rate of their abnormal behaviors that a node has observed. In this example, a node records Packet Drop Rate (PDR), Packet Modification Rate (PMR), and RTS flooding rate (RTS) for its neighbors. All other nodes are simultaneously collecting their neighbours' misbehavior information, and generate their local views as well as trust tables. The outlier candidates in the local views are sorted according to the distances between their nearest neighbors and themselves, and the top three outliers are picked in this example. We note that as long as all the nodes are observing the same set of behaviors, our approach can handle anything defined as a misbehavior.

The next step is the initial exchange of the local views, which is demonstrated in Fig. 3b. In this step, all the nodes send their local views to all of their *immediate* neighbors, which are defined as the nodes that are located one hop away from them. From Fig. 3b we find that the local views of node A and node B are not consistent.

Figure 3c exhibits the view update and optional rebroadcast step. Both node A and node B update their

Table 1 Simulation parameters

Parameter	Value
Simulation area	600 m×600 m
Number of nodes	50, 100, 150, 200
Transmission range	60 m, 90 m, 120 m
Mobility pattern	Random waypoint
Node motion speed	5 m/s, 10 m/s, 20 m/s
Number of malicious nodes	5, 10, 20
Simulation time	900 s

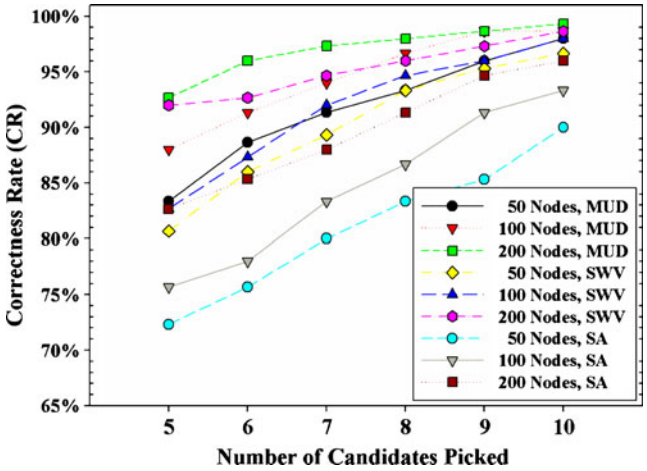

Fig. 4 CR with different number of nodes. (Number of malicious nodes: 5, area: 600 m×600 m, radio range: 120 m, motion speed: 5 m/s)

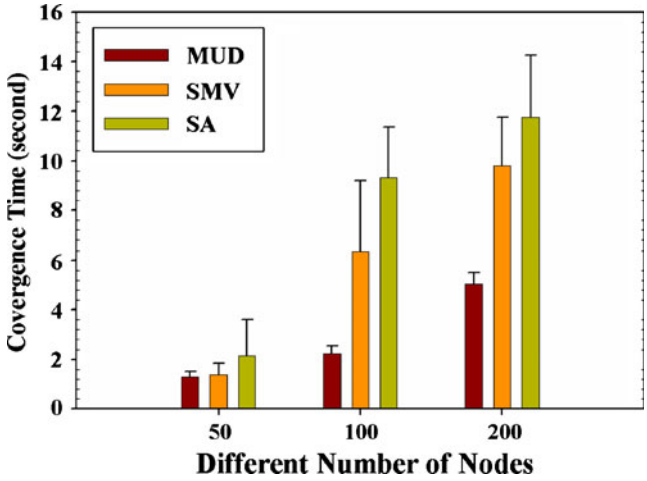

Fig. 6 CT with different number of nodes. (Same configuration as Figs. 4 and 5)

local views according to the view they have received. We note that node A applies the knowledge of trust to the local view update process. In this way, node A ensures that its updated local view contains the least fake information from node B, who is likely to be a malicious node since its trust value is quite low. Then, they rebroadcast their updated views to all the immediate neighbors. We should also be aware that node B may send out any arbitrary view to its immediate neighbors regardless of the true updated view it gets, because node B seems to be malicious.

The view update and optional rebroadcast process will continue until all the nodes hold the same view of the top three outliers, and this final state is shown in Fig. 3d. We find from Fig. 3d that the composition of the outliers has been significantly altered for both node A and B when compared to their initial views.

4 Evaluation

In this section, we examine the performance of our Multi-trust-based outlier Detection framework (MUD). We compare the two view combination techniques: MUD and Simple-trust-based Weighted Voting (SWV) against the baseline scheme, which utilizes the Simple Averaging method (SA). The SWV is proposed in our previous work [13].

4.1 Experimentation setup

We use Glomosim 2.03 [28] as our simulation platform, and the simulation setup is shown in Table 1. We use three parameters to assess the correctness and efficiency of our algorithms: Correctness Rate (*CR*), Communication Over-

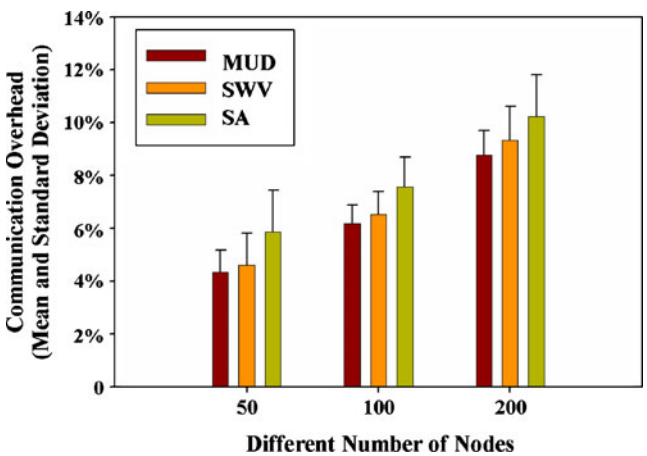

Fig. 5 CO with different number of nodes. (Number of malicious nodes: 5, area: 600 m×600 m, radio range: 120 m, motion speed: 5 m/s)

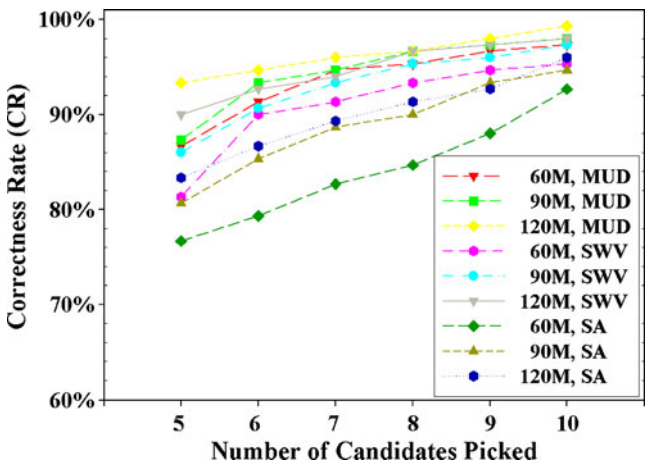

Fig. 7 CR with different radio ranges. (Same configuration as Figs. 8 and 9)

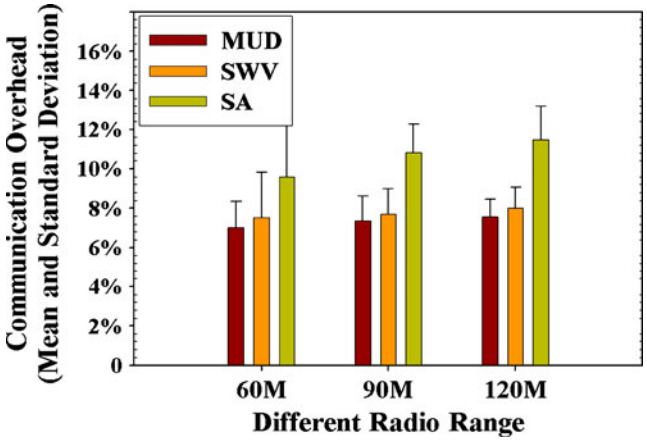

Fig. 8 CO with different radio ranges. (Number of nodes: 100, number of malicious nodes: 5, area: 600 m×600 m, motion speed: 5 m/s)

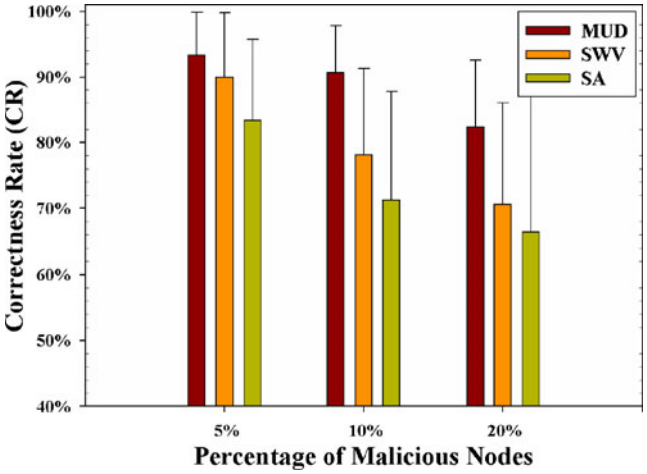

Fig. 10 CR with different percent of bad nodes. (Number of nodes: 100, radio range: 120 m, area: 600 m×600 m, motion speed: 5 m/s)

head (*CO*), and Convergence Time (*CT*). They are defined as follows:

$$CR = \frac{Number\ of\ True\ Outliers\ Found}{Total\ Number\ of\ Outliers}$$

$$TNPOD = Total\ Number \text{ of Packets for Outlier Detection}$$

$$CO = \frac{TNPOD}{Total \text{ Number of Packets in the network}}$$

$$CT = Time \text{ taken to form a } consistent \text{ global view of outliers}$$

Here we want to keep track of *CO* since we want to see the ratio of network traffic that outlier detection consumes over the whole network traffic. However, we also have interest in exploring the possible relationship between *TNPOD* and the number of nodes in the network. Note that we may keep track of the top k outliers during the outlier detection process, with k being a user-defined parameter. Therefore, "Number of Candidates Picked", which is the x-axis for Figs. 4, 7 and 13, refers to the value of k.

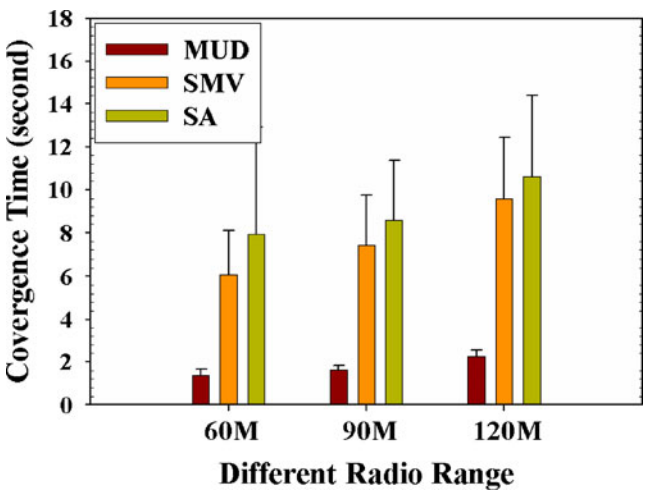

Fig. 9 CT with different radio ranges. (Number of nodes: 100, number of malicious nodes: 5, area: 600 m×600 m, motion speed: 5 m/s)

4.2 Adversary model

In our simulation, we assume that nodes either conform with various MANET protocols, such as routing protocols, or their behaviors deviate from the general protocol definition either intentionally (i.e. attackers) or unintentionally (i.e. faulty nodes). Both attackers and faulty nodes can hurt the network functionalities, and consequently we regard them both as adversaries.

In our simulation, we assume that adversaries can partially or completely drop, modify or misroute any packet that is sent to them. We also assume that they can deploy the Denial-of-Service (DoS) attack by continuously sending out Request-To-Send (RTS) packets to improperly occupy the communication channel all the time, which is also

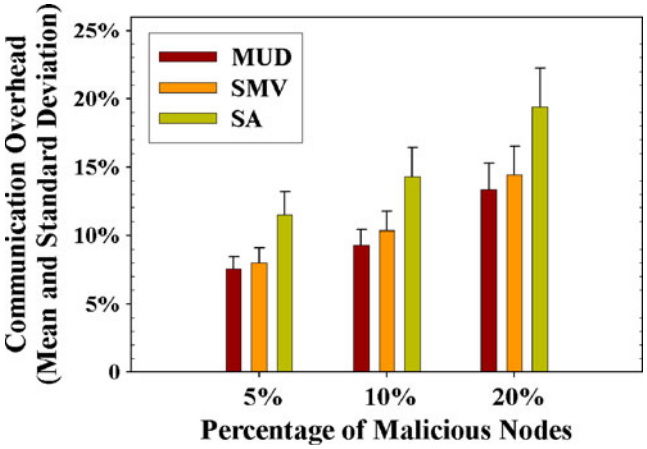

Fig. 11 CO with different percent of bad nodes. (Number of nodes: 100, radio range: 120 m, area: 600 m×600 m, motion speed: 5 m/s)

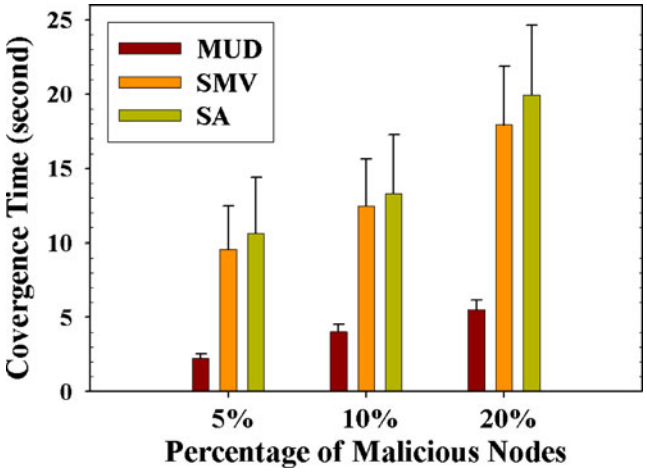

Fig. 12 CT with different percent of bad nodes. (Same configuration as Figs. 10 and 11)

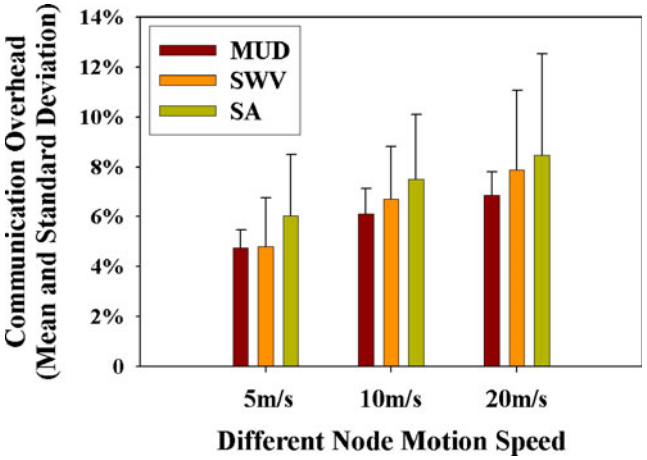

Fig. 14 CO with different node motion speeds. (Number of nodes: 100, number of malicious nodes: 5, radio range: 120 m, area: 600 m×600 m)

regarded as the RTS flood attack. The adversaries may mix all these misbehaviors at any arbitrary proportion so that it will be more difficult to identify their misbehaviors if observed only from one or two perspectives.

More importantly, the adversaries are capable of deliberately injecting fake observation reports for other benign nodes and spreading these fake reports to others. In this way, the benign nodes may be induced to draw incorrect conclusions that benign nodes are misclassified as misbehaving nodes. In our simulation, we also include some malicious nodes in the network that only spread fake observations, and these malicious nodes will not conduct any other misbehavior. Therefore, it is more difficult for us to identify these rumor spreaders since their misbehaviors are more difficult to be observed when compared to other evident misbehaviors, such as packet dropping or modification (Figs. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15).

4.3 Experimental results

In our experiments, we aim to observe the performance of our algorithm under different parameter configurations. We have compared the performance of our algorithm under the following four conditions: different number of nodes, different radio ranges, different percentage of malicious nodes, and different node motion speeds.

Figure 4 through Fig. 6 demonstrate the performance of the MUD algorithm with different number of nodes in the network. From these figures we find that when the number of nodes increases, the algorithm yields a higher correctness rate, but it also introduces more communication overhead and larger convergence time. This is consistent with our analysis because the information gathered to identify the outliers is generally more accurate if there are more observers. At the same time, more messages need to be

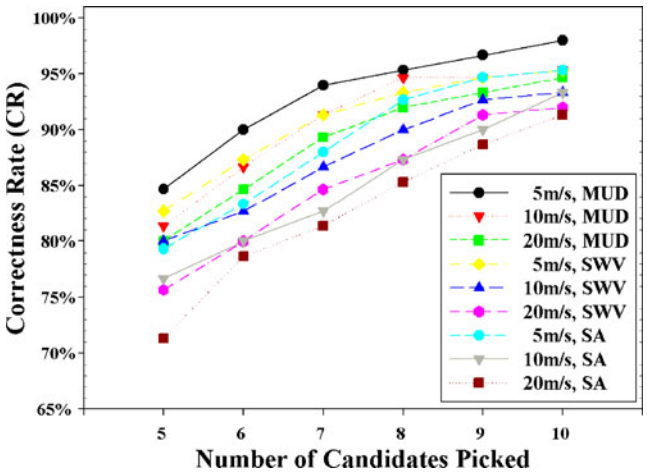

Fig. 13 CR with different node motion speeds. (Same configuration as Figs. 14 and 15)

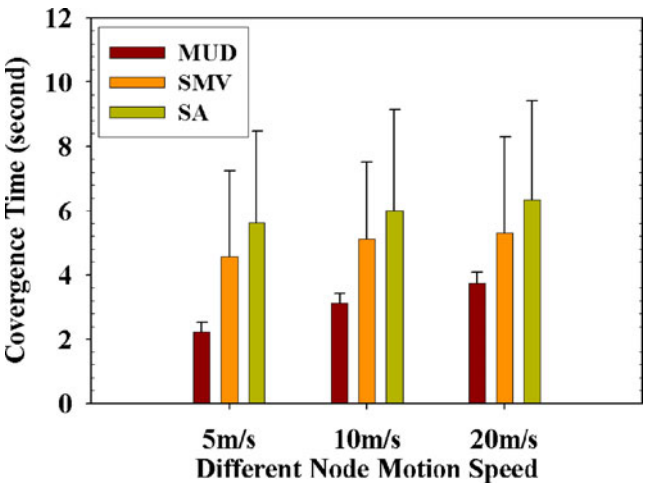

Fig. 15 CT with different node motion speeds. (Number of nodes: 100, number of malicious nodes: 5, radio range: 120 m, area: 600 m× 600 m)

exchanged amongst all the nodes to reach a consistent view when there are more nodes. Moreover, it may take a longer time to converge to a consistent global view if there are more nodes in MANETs. We also note that both MUD and SWV achieve better performances than SA. Moreover, MUD outperforms SWV in terms of higher correctness rate, lower communication overhead, and shorter convergence time.

Figures 7, 8 and 9 illustrate how the simulation results differ with different transmission ranges. We find that with a smaller radio range, all the three methods suffer from a performance degradation. When it is more difficult for the nodes to exchange the local views, the correctness rate of the final global view will surely be degraded. On the other hand, it is obvious that MUD still achieves a better performance than SWV and SA in that it yields a higher correctness rate with a lower communication overhead in a shorter period of time. This is true because MUD derives the trustworthiness of nodes from different perspectives, and the malicious nodes that devote to either mixed misbehaviors or certain misbehavior can easily be identified. On the other hand, since SWV determine the trustworthiness of the nodes from one dimension, it may be more difficult to recognize the malicious nodes that only perform a certain category of misbehaviors.

From Fig. 10 through Fig. 12, we find the simulation results with different percentage of malicious nodes. It is obvious that with a higher percentage of malicious nodes, MUD can yield a much better performance than SWV and SA. This is true because both SWV and SA rely on enough trustworthy information to make a correct decision: SA simply follows the decision from the majority of nodes, and the weights in SWV are also significantly determined by the second-hand information sent by other nodes. Hence, when there are a higher percentage of malicious nodes, the performances of both SWV and SA degrade noticeably. On the other hand, MUD can properly handle the outlier detection problem even in a more hostile environment because it derives the trustworthiness of each node from different view angle. For example, if some malicious nodes only spread fake observations, and they behave normally in all other cases, then it will be quite hard to recognize them as malicious nodes in SWV. On the contrary, since the MUD uses *reference trust* to keep track of the authenticity of the observations that each node reports, the *rumor spreader* will be much easier to detect. Consequently, the convergence time of MUD will surely be much shorter than that of SWV.

The experimental results under different node motion speeds are demonstrated in Fig. 13 through Fig. 15. We may conclude from these figures that while the nodes travel in a higher speed, the performance for all the methods become worse. This is true because it is harder for the nodes to exchange their views when they are travelling in a higher speed. However, in spite of the performance downgrade for all the methods, MUD still achieves a far better performance than both SWV and SA when the nodes move faster.

5 Conclusion

In this paper, we propose a collaborative and multi-dimensional-trust based outlier detection algorithm for securing mobile ad hoc networks. The gossip-based outlier detection algorithm can help us identify the outliers, which are generally the nodes that have exhibited some kind of abnormal behaviors. Given the fact that benign nodes rarely behave abnormally, it is highly likely that the outliers are malicious nodes. Moreover, a multi-dimensional trust management scheme is proposed to evaluate the trustworthiness of the nodes from multiple perspectives. Simulation results show that our algorithm is efficient and accurate with a small communication overhead.

References

1. Wu B, Chen J, Wu J, Cardei M (2006) A survey on attacks and countermeasures in mobile ad hoc networks. Wireless Network Security (Part II, Chapter 12), Springer
2. Zhang Y, Lee W (2002) Intrusion detection in wireless ad-hoc networks. In Proceedings of the 8th International Conference on Mobile Computing and Networking (MobiCom), Atlanta, GA, USA, pp 275–283
3. Hu Y, Perrig A, Johnson D (2002) Ariadne: a secure on-demand routing protocol for ad hoc networks. In Proceedings of the 8th International Conference on Mobile Computing and Networking (MobiCom), Atlanta, GA, USA, pp 12–23
4. Sun B, Guan Y, Chen J, Pooch UW (2003) Detecting black-hole attack in mobile ad hoc networks. In Proceedings of 5th European Personal Mobile Communications Conference, Glasgow, Scotland, UK, pp 490–495
5. Grubbs F (1969) Procedures for detecting outlying observations in samples. Technometrics 11(1):1–21
6. Lazarevic A, Ertoz L, Ozgur A, Srivastava J, Kumar V (2003) A comparative study of anomaly detection schemes in network intrusion detection. In Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, May
7. Zhang J, Zulkernine M (2006) Anomaly based network intrusion detection with unsupervised outlier detection. In Proceedings of IEEE International Conference on Communications (ICC 2006), Istanbul, Turkey, pp 2388–2393
8. Ferdousi Z, Maeda A (2006) Unsupervised outlier detection in time series data. In Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW06), Atlanta, GA, USA, Apr
9. Sheng B, Li Q, Mao W, Jin W (2007) Outlier detection in sensor networks. In Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc07), Montreal, Quebec, Canada, pp 219–228

10. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large datasets. In Proceedings of the 2000 ACM SIGMOD international Conference on Management of Data, Dallas, Texas, USA, pp 427–438
11. Knorr EM, Ng RT (1998) Algorithms for mining distance-based outliers in large datasets. In Proceedings of the 24th international Conference on Very Large Data Bases (VLDB98), New York, NY, USA, pp 392–403
12. Knorr EM, Ng RT (1999) Finding intensional knowledge of distance-based outliers. In Proceedings of the 25th international Conference on Very Large Data Bases (VLDB99), Edinburgh, Scotland, UK, pp 211–222
13. Li W, Parker J, Joshi A (2008) Security through collaboration in MANETs. In Proceedings of the 4th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2008), Springer LNICST vol. 10, Orlando, FL, USA, pp 696–714, November
14. Li W, Joshi A (2009) Outlier detection in ad hoc networks using dempster-shafer theory. In Proceedings of the 10th International Conference on Mobile Data Management (MDM 2009), Taipei, Taiwan, pp 112–121, May
15. Li W, Joshi A, Finin T (2009) Policy-based malicious peer detection in ad hoc networks. In Proceedings of 12th IEEE International Conference on Computational Science and Engineering (CSE 2009), Vancouver, Canada, Vol. 3, pp 76–82, August
16. Branch J, Szymanski B, Giannella C, Wolff R, Kargupta H (2006) In-network outlier detection in wireless sensor networks. In Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS2006),Lisbon, Portugal
17. Palpanas T, Papadopoulos D, Kalogeraki V, Gunopulos D (2003) Distributed deviation detection in sensor networks. ACM SIGMOD Record 32(4):77–82
18. Subramaniam S, Palpanas T, Papadopoulos D, Kalogeraki V, Gunopulos D (2006) Online outlier detection in sensor data using non-parametric models. In Proceedings of the 32nd international Conference on Very Large Data Bases (VLDB06), Seoul, Korea, pp 187–198
19. Marti S, Giuli TJ, Lai K, Baker M (2000) Mitigating routing misbehavior in mobile ad hoc networks. In Proceedings of the 6th Annual international Conference on Mobile Computing and Networking (MOBICOM00), Boston, MA, USA, pp 255–265
20. Liu Y, Comaniciu C, Man H (2006) A Bayesian game approach for intrusion detection in wireless ad hoc networks. In Proceeding of the 2006 Workshop on Game theory For Communications and Networks (GAMENET06), Pisa, Italy
21. Zhang Y, Lee W (2000) Intrusion detection in wireless ad-hoc networks. In Proceedings of the 6th Annual international Conference on Mobile Computing and Networking (MOBICOM00), Boston, MA, USA, pp 275–283
22. DengH, Zeng Q, Agrawal DP (2003) SVM-based intrusion detection system for wireless ad hoc networks. In Proceedings of the IEEE Vehicular Technology Conference (VTC03), vol. 3, pp 2147–2151, Orlando, FL, USA
23. Kachirski O, Guha R (2002) Intrusion detection using mobile agents in wireless ad hoc networks. In Proceedings of the IEEE Workshop on Knowledge Media Networking, pp 153–158, Kyoto, Japan
24. Tseng C, Balasubramanyam P, Ko C, Limprasittiporn R, Rowe J, Levitt K (2003) A specification-based intrusion detection system for AODV. In Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks(SASN03), Fairfax, VA, USA, pp 125–134
25. Huang Y, Lee W (2003) A cooperative intrusion detection system for ad hoc networks. In Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN03), Fairfax, VA, USA, pp 135–147
26. Kefayati M, Rabiee HR, Miremadi SG, Khonsari A (2006) Misbehavior resilient multi-path data transmission in mobile ad-hoc networks. In Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN06), Alexandria, VA, USA, pp 91–100
27. Anderegg L, Eidenbenz S (2003) Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In Proceedings of the 9th Annual international Conference on Mobile Computing and Networking (MOBICOM03), San Diego, CA, USA, pp 245–259
28. Glomosim 2.03, http://pcl.cs.ucla.edu/projects/glomosim/
29. Parker J, Patwardhan A, Joshi A (2006) Cross-layer analysis for detecting wireless misbehavior. In Proceedings of the IEEE Consumer Communications and Networking Conference(CCNC 2006), Las Vegas, Nevada, USA, Jan
30. He Q, Wu D, Khosla P (2004) SORI: a secure and objective reputation-based incentive scheme for ad hoc networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), vol. 2, pp 825–830
31. Patwardhan A, Perich F, Joshi A, Finin T, Yesha Y (2005) Active collaborations for trustworthy data management in ad hoc networks. In Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, November
32. Srinivasan V, Nuggehalli P, Chiasserini C-F, Rao RR (2005) An analytical approach to the study of cooperation in wireless ad hoc networks. IEEE Trans Wireless Commun 4(2):722–733
33. Buchegger S, Boudec J-YL (2003) A robust reputation system for mobile ad-hoc networks. In Proceedings of P2PEcon
34. Griffiths N (2005) Task delegation using experience-based multi-dimensional trust. In Proceedings of the Fourth international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 05), pp 489–496, The Netherlands, July
35. Reece S, Rogers A, Roberts S, Jennings NR (2007) Rumors and reputation: evaluating multi-dimensional trust within a decentralized reputation system. In Proceedings of the 6th international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 07), pp 1–8, Honolulu, Hawaii, USA May