Outlier Detection in Ad Hoc Networks Using Dempster-Shafer Theory

Wenjia Li and Anupam Joshi
Department of Computer Science and Electrical Engineering,
University of Maryland, Baltimore County (UMBC),
1000 Hilltop Circle, Baltimore MD 21250
{wenjia1, joshi}@cs.umbc.edu

Abstract

Mobile Ad-hoc NETworks (MANETs) are known to be vulnerable to a variety of attacks due to lack of central authority or fixed network infrastructure. Many security schemes have been proposed to identify misbehaving nodes. Most of these security schemes rely on either a predefined threshold, or a set of well-defined training data to build up the detection mechanism before effectively identifying the malicious peers. However, it is generally difficult to set appropriate thresholds, and collecting training datasets representative of an attack ahead of time is also problematic. We observe that the malicious peers generally demonstrate behavioral patterns different from all the other normal peers, and argue that outlier detection techniques can be used to detect malicious peers in ad hoc networks. A problem with this approach is combining evidence from potentially untrustworthy peers to detect the outliers. In this paper, an outlier detection algorithm is proposed that applies the Dempster-Shafer theory to combine observation results from multiple nodes because it can appropriately reflect uncertainty as well as unreliability of the observations. ***The simulation results show that the proposed scheme is highly resilient to attackers and it can converge stably to*** *a common outlier view amongst distributed nodes with a limited communication overhead****.***

1. Introduction

A Mobile Ad-hoc NETwork (MANET), as its name suggests, has no fixed infrastructure, and is generally composed of a dynamic set of cooperative peers, which are willing to share their wireless transmission power with other peers so that indirect communication can be possible between nodes that are not in the radio range of each other .

The nature of MANETs, such as node mobility, unreliable transmission medium and restricted battery power, makes them extremely vulnerable to a variety of attacks [1] [2]. Wireless links, for instance, are generally prone to both passive eavesdropping and active intrusion. Moreover, there are various sophisticated attacks that are difficult to identify, such as greyhole attacks [3], blackhole attacks [4], wormhole attacks [5], and Sybil attacks [6]. Another security concern in ad hoc networks is caused by the cooperative nature of the nodes. Attacks from external adversaries may disturb communications, but the external intruder generally cannot directly participate in the cooperative activities among the nodes, such as routing, because they do not possess the proper secure credentials, such as shared keys. However, compromised nodes, which are taken over by an adversary, are capable of presenting the proper secure credentials, and consequently can interfere with almost all of the network operations, such as route discovery, key management and distribution, and packet forwarding. Misbehavior surveillance and detection is a crucial method that has been used in MANETs to protect them from both external adversaries and internal malicious nodes.

The misbehaviors observed by neighboring peers typically include dropping, modification, misrouting of packets at the network layer, as well as false Request/Clears in the MAC layer etc. Nevertheless, many of these events may occur due to environmental and mobility related reasons, not just malicious intent. For instance, a packet may get dropped when a node's buffer gets full because of its inability to forward packets on a noisy channel. Most of the current misbehavior detection schemes rely on either a predefined threshold, or a set of well-defined training data to infer thresholds. This threshold is used in the detection mechanism to separate malicious behaviors from what is normal given the conditions. However, it is generally difficult to set appropriate thresholds,

because the network is quite dynamic and unpredictable, and environmental conditions such as ambient RF noise can vary. Moreover, collecting training datasets representative of an attack ahead of time is also problematic. Since the adversaries can constantly modify their attack patterns, which may contain location (attack target), type, and degree (amount of packets affected) of the attack, it is also difficult to gather the training dataset that can appropriately predict the attack pattern of the adversaries. On the contrary, we do not need to rely on any previous knowledge to find a node that is an outlier with respect to a given observable. In order to disrupt communications or launch some other attack, the malicious node will have to behave in a way distinct from the "good" nodes. So we can detect node misbehaviors by means of outlier detection.

Outlier detection is generally an important step prior to almost all kinds of data analysis routines. Outliers are normally defined as data points that have significant difference from the rest of the data according to a certain measure [7, 8]. Outlier detection is used to either eliminate or amplify outliers: the first is to reduce the noise in the data; the second is to expose the outliers for further analysis.

In this paper, we develop and evaluate a gossip-based outlier detection algorithm for mobile ad hoc networks. In our approach, all the peers in MANETs observe and record the abnormal behaviors of their neighbors in a manner similar to existing methods such as [9, 10, 11, 12]. In contrast to most existing approaches, each peer will then calculate its local version of outliers based on its own observations. In the next stage, the peers exchange their local views with their immediate neighbors, and they update the local views if the received views are more accurate than their own views. The updated local views will be further broadcast to the immediate neighbors. The process continues, and it will not halt until there is no more view update amongst the peers.

The most important step in the proposed outlier detection algorithm is the local view update step. In this step, we need to derive the updated local view of outliers from multiple local views provided by our (one hop) peers. However, the peers may be giving us inaccurate data, either from malicious intent or out of ignorance. The contribution of this paper is to see if correct outlier decisions can be made in presence of potentially unreliable views. We use both the weighted voting method and the Dempster-Shafer Theory (DST) [38] of evidence to combine the local views from multiple neighbors. The Dempster-Shafer theory of evidence is particularly well suited for this type of problem because it can capture uncertainty. Furthermore, Dempster's rule for combination can be used to fuse together multiple pieces of views from both reliable and unreliable observers.

Some important features of our algorithm are: (1) its deployment does not rely on any priori knowledge, such as pre-classified training dataset or pre-defined security threshold; (2) it is compatible with different outlier detection heuristics; (3) it is resilient to attempts by misbehaving nodes to defeat it; (4) it is resilient to the motion and failure of nodes in MANETs; (5) it is efficient in terms of communication overhead; and (6) all the nodes will stably converge to a common view of outliers as long as these nodes do not change their behaviors significantly during the convergence time of the algorithm.

In the rest of this paper, we give a survey of related work in Sec. 2. In Sec. 3, we present our outlier detection algorithm. We evaluate the effectiveness of our scheme through simulation in Sec. 4, and conclude in Sec. 5.

2. Related work

2.1. Outlier detection

Outlier detection is a long studied topic in the data mining research, and a variety of outlier detection approaches have been proposed for different application domains, such as large-scale databases [13, 14, 15], high-dimensional datasets [16, 17, 18], and wireless sensor networks [19, 20, 21, 22].

Notably, Branch et al. [22] propose an in-network outlier detection scheme to detect the outliers in wireless sensor networks. In this scheme, all the sensor nodes will first calculate the local outlier(s). Then some messages, which contain the local outlier(s) as well as some other supportive information, will be exchanged amongst all the nodes. The message exchanging process will not halt until all the nodes have the same global view of outlier(s). Our proposed outlier detection algorithm is somewhat similar to the method proposed by Branch et al. However, there are three 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 into consideration. Second, there is no malicious behaviors in the discussion of the method by Branch et al., 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. Third, the method by Branch et al. has not taken uncertainty of the local views into consideration: they assume that the information exchange and

dissemination process amongst the nodes is reliable, and no uncertainty will be introduced to the local views during this message exchange process. On the contrary, we have considered uncertainty of the local views that may be introduced during the message exchange process.

2.2. Misbehavior detection in mobile ad hoc networks

Work on misbehavior detection (may also be called as intrusion detection) has produced very rich literature in traditional, P2P and ad hoc networks. In the latter, most contributions assume that there is no fixed network and security infrastructure that misbehavior detection mechanism can rely on.

Four types of misbehaviors in ad hoc networks are identified and discussed in [23], which are failed node behaviors, badly failed node behaviors, selfish attacks, and malicious attacks. These four types of node misbehaviors are classified with respect to the node's intent and action. Remarkably, selfish attacks are intentional passive misbehaviors, where nodes choose not to fully participate in the packet forwarding functionality to conserve their resources, such as battery power; malicious attacks are intentional active misbehaviors, where the malicious node aims to purposely interrupt network operations. The existence of selfishness and malicious behaviors has motivated research in the area of misbehavior detection for mobile ad hoc networks.

Intrusion Detection System (IDS) is an essential means to detect various node misbehaviors. Several approaches have been proposed to build IDS on each individual peer due to the lack of a fixed infrastructure [10, 24, 25, 26]. In these approaches, every node is equipped with an IDS sensor, and each IDS sensor is assumed to be always on, which is apparently not energy efficient given the limited battery power of nodes in ad hoc networks. In contrast, Huang et al. [27] propose a cooperative intrusion detection framework, in which clusters are formed in ad hoc networks and all the nodes in each cluster will take over the intrusion detection operations in turn. This cluster-based approach can definitely reduce the power consumption for each node.

Routing misbehavior is another kind of malicious activity that is common in ad hoc networks. If an attacker aims to degrade the network service of ad hoc network, then he can try to compromise some nodes in the ad hoc network, and use them to disturb the routing services so as to make part of or the entire network unreachable. Marti et al. [28] introduce two related techniques, namely *watchdog* and *pathrater*, to detect and isolate misbehaving nodes, which are nodes that do not forward packets. There are also some other proposed solutions that aim to cope with the routing misbehaviors [29, 30, 31].

There is little work in which the Dempster-Shafer Theory has been applied to MANETs. The two most relevant research efforts are discussed in [40] and [41]. In [40], DST is utilized to combine the direct observation results from each IDS sensor, whereas we are using DST to combine views of outliers. Raya et al. [41] proposed a trust management scheme for VANET, in which DST is used to combine multiple evidences for trust. Nevertheless, there are two major differences between our work and theirs. First, there are strong logical relationships among the local views in our work, because an updated local view is always derived from the previous local views. On the contrary, the report of evidence in [41] may be independent. Second, DST is used to combine the local views in real-time in our approach. On the other hand, in [41], DST is not applied to combine evidences in real-time.

In our previous work [32], we have done a preliminary study where outlier detection method is adopted to identify node misbehaviors. However, we merely utilize weighted voting method to fuse together multiple pieces of local views, which may not perfectly reflect uncertainty in the local views. Here we apply the Dempster-Shafer theory of evidence to combine multiple local views, and compare their performances under different criteria in simulation.

3. Gossip-based outlier detection algorithm

The aim of our gossip-based outlier detection algorithm is to identify the top k outliers in terms of some abnormal behaviors observed by neighbors, such as packet drops or modifications. Here k is a user-defined parameter, and it can be assigned any positive integer value. Gossiping in MANETs generally refers to the repetitive probabilistic exchange of messages between two peers in MANETs. By the utilization of restricted gossiping method in the outlier detection algorithm, the communication overhead of our algorithm can be noticeably bounded.

3.1. Preliminaries

We denote *node* as a system entity in mobile ad hoc networks that is capable of observing the behaviors of other entities within its radio transmission range, and exchanging these observations with other entities in its radio transmission range. A *neighbor* of a node A is defined as a node that resides within A's radio transmission range. The type of abnormal behaviors that each node constantly observes can be fully user-

defined as long as all the nodes observe the same set of abnormal behaviors.

While a node observes and records the abnormal behaviors that its neighbors demonstrate, it also keeps track of the total amount of incoming packets it has observed for each neighbor. When a node needs to summarize its observation and thereby form its local view of outliers, it will calculate the rate of abnormal behaviors over the all behaviors it has observed for the node. For example, if all the nodes choose to observe the behaviors of packet drop, modification and misroute, then the packet drop rate (PDR), packet modification rate (PMOR) and packet misroute rate (PMIR) can be defined as follows, respectively.

$$PDR = \frac{Number\ of\ Packets\ Dropped}{Total\ Number\ of\ Incoming\ Packets}$$

$$PMOR = \frac{Number\ of\ Packets\ Modified}{Total\ Numer\ of\ Incoming\ Packets}$$

$$PMIR = \frac{Number\ of\ Packets\ Misrouted}{Total\ Number\ of\ Incoming\ Packets}$$

We define the *trustworthiness* of a node N_k as a real value θ_k that can properly reflect the probability with which the node will perform the exact actions that it is supposed to take. θ_k can be assigned any real value in the range [0, 1], and the higher the value of θ_k, the node N_k is more reliable and has a higher probability to take the correct actions. For instance, some nodes are deployed with stronger encryption mechanisms, closely monitored by certain security surveillance system, and better protected against various attacks. Given that they are generally less likely to perform faulty actions, they are regarded as more trustworthy.

The trustworthiness θ_k of a node N_k is defined as a function of all misbehaviors that other nodes have observed for the node N_k. Namely, the trustworthiness θ_k is calculated as follow.

$$\theta_k = 1 - \sum_i P_i * M_{ki}$$

Here P_i denotes the *punishment factor* for the i-th misbehavior, which indicates the severity degree of its outcome. M_{ki} represents the rate of this misbehavior over the total observed behaviors. For example, if packet drop, packet modification, and packet misroute are the three exact misbehaviors we are observing, then θ_k can be derived as follow.

$$\theta_k = 1 - P_{drop} * PDR - P_{modification} * PMOR - P_{misroute} * PMIR$$

In our outlier detection framework, trustworthiness of nodes is initialized and updated by a trust management scheme described in Sec. 3.3.

3.2. Framework description

The outlier detection algorithm has the following four steps, viz. local view formation, local view exchange, view combination, and global view formation. In this algorithm, we have utilized two methods to properly combine multiple local views, which are weighted voting and the Dempster-Shafer theory of evidence, respectively. Fig. 1 illustrates our framework.

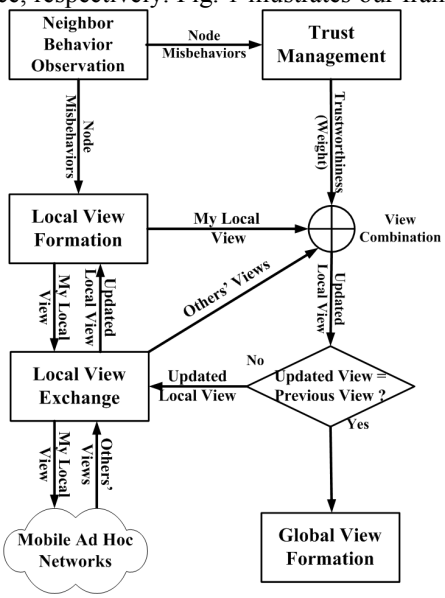

Figure 1. **Gossip-based outlier detection framework**

Prior to the local view formation, each node observes and record the behaviors of their neighbors, and also keeps track of the total number of incoming packets each of their neighbors has. Based on their observations, the initial local view of outliers is generated. We may note that there are a variety of definitions for outliers. Here we adopt two well-known distance-based definitions: (1) distance to the nearest neighbor (*NN*), and (2) average distance to k nearest neighbors (k-*NN*) [13].

Once all the nodes form their initial local views, they broadcast their initial local views to all of their immediate neighbors, i.e., all the nodes that are within their direct radio transmission ranges. When a node receives a local view from one of its neighbors, it then checks whether the incoming view differs from its own local view. If so, it combines the two views, and rebroadcasts the combined view to its immediate neighbors. If not, it simply retains its local view and keeps silent.

Unlike the traditional gossiping method, the more the number of nodes that accept the same view of outliers, the less the number of new view updates that are sent out. Here, we assume that all the nodes will not add any new observation results of their own to their own local views once they start exchanging local

views with their neighbors. Ultimately, the algorithm converges to a global view that all the nodes hold.

The global view is regarded as a *snapshot* that can properly illustrate the comprehensive observation results of all nodes at the time spot when the nodes start exchanging their views. Due to node mobility and changing network topology, the status of nodes and the network changes over time. Therefore, the global views will get outdated. To address this problem, we can periodically repeat the outlier detection process in order to keep the global views up-to-date. The repeat interval can be determined by both the availability of resources (bandwidth, battery power, etc.) and the levels of node mobility as well as topology change.

3.3. Trust management

A variety of trust and reputation management approaches have been studied during the past decades for instance [33, 34, 35]. All of these trust management approaches can fit our system. For the experiments presented in this paper, we adopt a simple but well-defined trust management scheme, in which each nodes trustworthiness θ_k is initially set to a default value. A peers θ_k is modified whenever we obtain any novel information regarding its trustworthiness in terms of both direct observation results from the node itself and indirect observation results from other nodes. Direct observation results and indirect observation results are generally called *first-hand information* and *second-hand information*, respectively [36].

First let us see how the direct observation results are obtained and utilized in our trust management scheme. As we have mentioned, trustworthiness θ_k is initially set as 1. Whenever a node observes any misbehavior of its neighbor k, it reduces θ_k according to the *punishment factors*. These can vary for different misbehaviors. For instance, packet drop and packet misroute are both misbehaviors. Nevertheless, packet drop may be caused by either intentional malicious behavior or power failure. On the contrary, if we observe packet misroute by a node, it is more likely that this is a deliberate act. This observation is also true for packet modification. Therefore, we set a higher punishment factor for both packet modification and packet misroute than for packet drop.

With the limited radio transmission range as well as the mobility of the nodes, it is highly unlikely that a node can have the opportunity to observe the behaviors of all other nodes in MANETs. Hence, it is essential to integrate second-hand information obtained from other nodes to our trust management scheme.

However, since malicious nodes can intentionally disseminate falsified second-hand information to their neighbors so as to disturb the trust management scheme as well as protect themselves from being disclosed, second-hand information from other nodes may not be trustworthy at all times. Therefore, it is necessary to adopt a proper method to combine multiple pieces of second-hand information from both trustworthy and untrustworthy neighbors. In our trust management scheme, we may utilize either weighted voting or the Dempster-Shafer theory to appropriately integrate multiple pieces of second-hand information into the first-hand information that each node directly observes. Fig. 2 shows the trust management scheme.

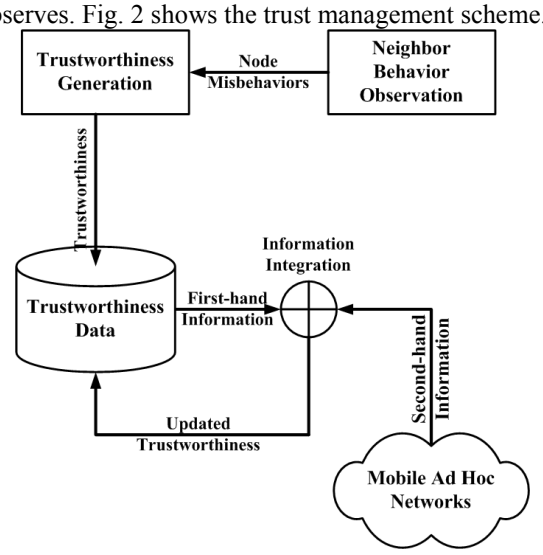

Figure 2. **Trust Management Scheme**

3.4. View combination

View combination is the most important step in our outlier detection algorithm. Because some of the incoming views are not reliable, it is essential to find a view combination technique to properly fuse together multiple pieces of views.

Among the various data fusion techniques, Bayesian approaches [37] and Dempster-Shafer theory of evidence [38] are two of the most frequently used techniques. There are two fundamental differences between DST and Bayesian theorem. First, unlike the definition in the Bayesian theorem, the lack of knowledge about an incident is not regarded as a negative evidence for that incident in DST. Second, given that there are two incidents in DST that are inconsistent with each other, the uncertainty about one of them may be viewed as the positive evidence for the other incident. In one word, in DST a node can hold either supportive or uncertain opinion toward an event, whereas a node must pick either positive or negative attitude toward an event in Bayesian theorem. For instance, suppose a node A observes that its neighbor B has dropped 10% of its incoming packets. Then according to our definition of PDR in Sec. 3.1, the

PDR for node *B* is 0.1. If we apply the Bayesian approach in this example, then we may draw a conclusion that node *B* has 90% probability to properly forward the incoming packets, with which we can easily conclude that node *B* is a good node. However, if node *A* observes 10% packet drops because of node movement and attack pattern change (e.g. switch from packet drop to packet modification), then we should not have drawn the conclusion that node *B* is good. On the contrary, in DST, we still compute PDR as 10%, but we will mark the rest 90% as "uncertain" instead of a refutal of evidence, which makes the decision process more realistic.

Hence, the Dempster-Shafer theory is more suitable when there is uncertainty or even no priori knowledge for the event. In our outlier detection framework, each node initially builds its local view for its neighbors based on its own observations. Due to the node mobility and limited radio range, each node may only observe a part of its neighbors' behaviors. As a result, the initial local views may be biased, and they definitely contain uncertainty. In this light, we believe that DST fits well in our framework.

In DST, probability is replaced by an uncertainty interval bounded by belief and plausibility. Belief is the lower bound of this interval and represents supporting evidence. Plausibility is the upper bound of the interval and represents non-refuting evidence. For instance, if a node *N* observes that one of its neighbors, say node *M*, has dropped packets with probability *p*, then node *N* has *p* degree of belief in the packet dropping behavior of node *M* and 0 degree of belief in its absence. The belief value with respect to an event α_i and observed by node *N* can be computed as the following.

$$bel_N(\alpha_i) = \sum_{e:\alpha_e \subset \alpha_i} m_N(\alpha_e)$$

Here α_e are all the basic events that compose the event α_i, and $m_N(\alpha_e)$ represents the view of the event α_e by node *N*. In our case, since node *N* only get a single report of node *M* from itself, i.e., $\alpha_i \subset \alpha_i$. Thus, we can conclude that $bel_N(\alpha_i) = m_N(\alpha_i)$. The equation $pls(\alpha_i) = 1 - bel(\overline{\alpha_i})$ holds for belief and plausibility. Therefore, in our example, we can get the following: $bel_N(M) = m_N(M) = p$ and $pls_N(M) = 1 - bel_N(\overline{M}) = 1$.

Given that belief indicates the lower bound of the uncertainty interval and represents supportive evidence, we define the combined packet dropping level of node *A* as the following.

$$pd_A = bel(A) = m(A) = \bigoplus_{k=1}^{K} m_k(A)$$

Here $m_k(A)$ denotes the view of node *k* on another node *A*. We can combine reports from different nodes by applying the Dempster's rule, which is defined as following.

$$m_B(A) \oplus m_C(A) = \frac{\sum_{q,r:\alpha_q \cap \alpha_r = A} m_B(\alpha_q) m_C(\alpha_r)}{1 - \sum_{q,r:\alpha_q \cap \alpha_r = \Phi} m_B(\alpha_q) m_C(\alpha_r)}$$

As a comparator, we consider the Weighted Voting method (WV) in our framework. As the name suggests, weighted voting method adds up the multiple pieces of views with each view weighted by the corresponding trustworthiness to yield the updated view of outliers. The weighted voting method can be expressed as:

$$V = \sum_{i=1}^{N} w_i * V_i$$

4. Performance evaluation

In this section, we examine the performance of our outlier detection framework. We compare the two view combination techniques: DST and WV (which we have recently proposed [32]) against the baseline scheme, which utilizes the siMple aVerging method (MV).

4.1. Simulation setup

We use GloMoSim 2.03 [39] as the simulation platform, and table 1 lists the parameters used in the simulation scenarios.

Table 1. **Simulation parameters**

Parameter	*Value*
Simulation area	150m × 150m, 300m × 300m, 450m × 450m, 600m × 600m
Number of nodes	15, 25, 50, 100, 150, 200
Transmission range	45m, 60m, 90m, 120m
Mobility pattern	Random waypoint
Node motion speed	5m/s, 10m/s, 20m/s
Number of malicious nodes	5, 10, 15, 20
Simulation time	900 s

We use three parameters to evaluate the correctness and efficiency of our algorithm: Correctness Rate (*CR*), Communication Overhead (*CO*), and Convergence Time (*CT*). They are defined as follows.

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

$$CO = \frac{Number\ \ of\ \ Packets\ \ for\ \ Outlier\ \ Detection}{Total\ \ Number\ \ of\ \ Packets\ \ in\ \ network}$$

$$CT = Time\ \ taken\ \ to\ \ form\ \ a\ \ consistent\ \ global\ \ view$$

Each simulation scenario has 30 runs with distinct random seed, which ensures a unique initial node placement for each run.

4.2. Adversary Model

In our simulation, nodes either abide by various MANET protocols, such as AODV routing protocol, or their behaviors deviate from the protocol definition either intentionally (i.e. attackers) or unintentionally (i.e. faulty nodes). Both attackers and faulty nodes can do harm to the network functionalities, and consequently we regard them both as adversaries. In general, 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 so as to improperly occupy the communication channel all the time, which is also regarded as the RTS flood attack. The adversaries may mix all these misbehaviors 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 faulty data and spreading these fake data to other benign nodes. In this way, the benign nodes may be induced to generate faulty reports in which benign nodes can be misclassified as misbehaving nodes.

4.3. Simulation results

The goal of the simulations is to observe the performance of our algorithm under different parameter configurations. We have compared the performance of our algorithm under the following five conditions: different number of nodes, different simulation areas, different radio ranges, different percentage of malicious nodes, and different node motion speeds. The simulation results are showed in the following Fig. 3 through Fig. 7.

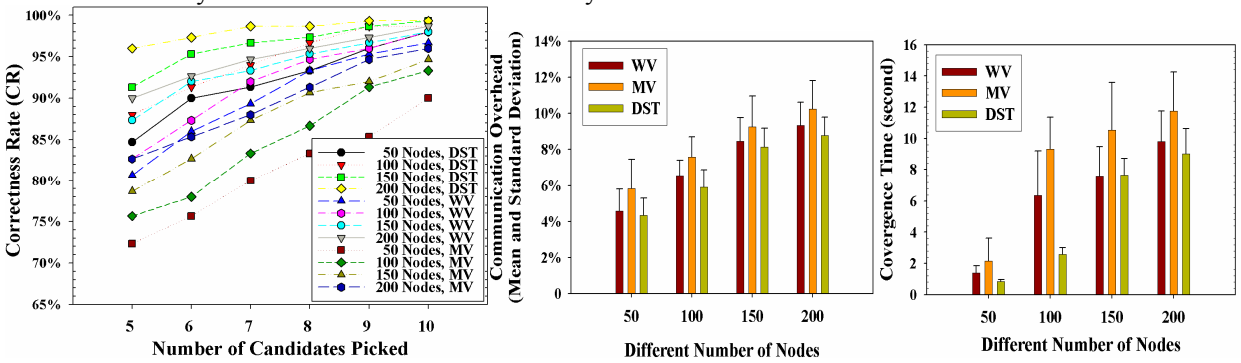

Figure 3. **CR, CO, CT with different number of nodes (number of malicious nodes: 5, area: 600m ×600m, radio range: 120m, motion speed: 5m/s)**

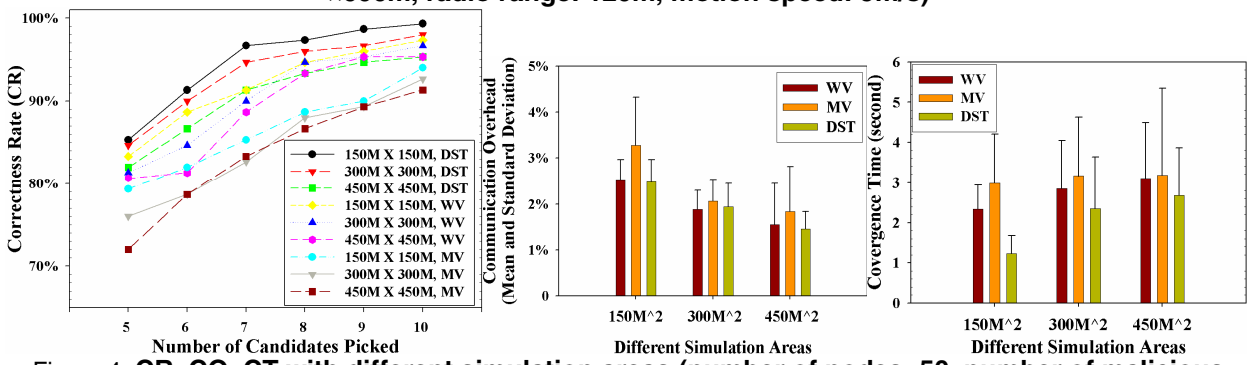

Figure 4. **CR, CO, CT with different simulation areas (number of nodes: 50, number of malicious nodes: 5, radio range: 60m, Motion Speed: 5m/s)**

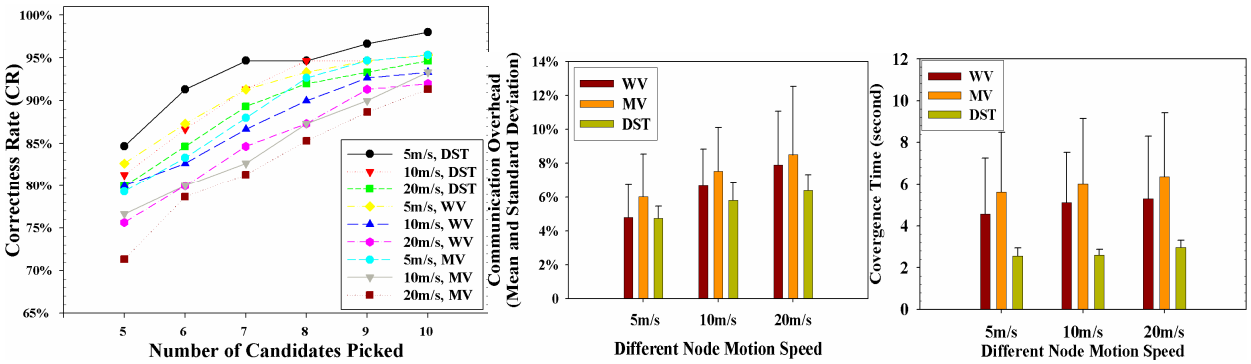

Figure 5. **CR, CO, CT with different node motion speeds (number of nodes: 100, number of malicious nodes: 5, radio range: 120m, area: 600m×600m)**

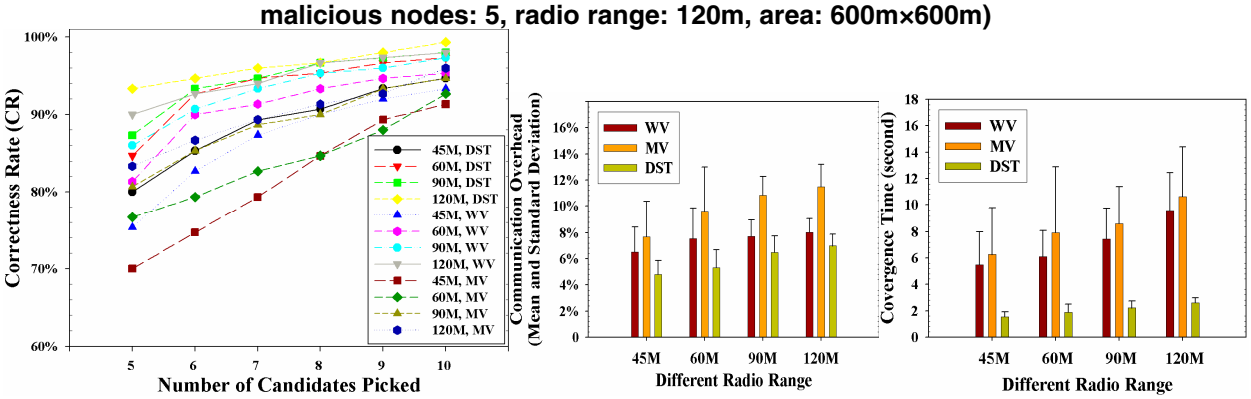

Figure 6. **CR, CO, CT with different transmission ranges (number of nodes: 100, number of malicious nodes: 5, area: 600m×600m, node motion speed: 5m/s)**

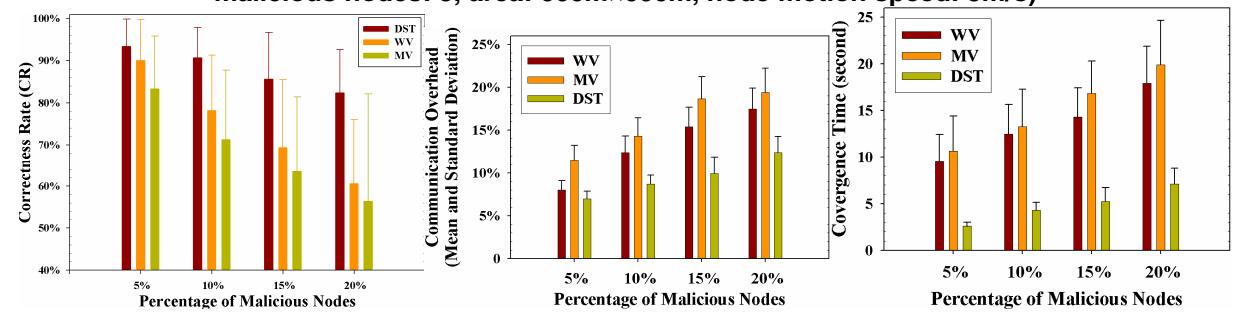

Figure 7. **CR, CO, CT with different percentage of malicious nodes (number of nodes: 100, radio range: 120m, area: 600m×600m, node motion speed: 5m/s)**

Fig. 3 exhibits the performance of our algorithm with different number of nodes in the network. From Fig. 3 we find that when the number of nodes is increased, the algorithm yields a higher correctness rate, but it also introduces more communication overhead. 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 exchanged amongst all the nodes to reach a consistent view when there are more nodes. We also note that the both DST and WV demonstrate better performance than MV. Moreover, DST has better performance over WV in terms of higher correctness rate, lower communication overhead, and shorter convergence time.

In Fig. 4, we can find the different performance of our algorithm with different simulation areas. It is clear that the correctness rate decreases as we increase the simulation area for all three methods. We also find that the communication overhead is reduced as the simulation area becomes larger. Since the nodes have a lower probability to communicate with other nodes if the simulation area becomes larger, the correctness rate will surely become lower. Moreover, there will also be less communication overhead, and it will generally take longer time for all the nodes to reach an agreement on the global view of outliers. Similarly as Fig. 3 implies, we can find that both DST and WV produce better performance that MV. Moreover, we also note that DST always wins over WV in terms of

correctness and convergence time when simulation area enlarges. However, there is no significant difference between DST and WV in terms of communication overhead.

The simulation results with different motion speeds are shown in Fig. 5. We may conclude from Fig. 5 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 traveling in a higher speed. However, in spite of the performance downgrade for all the methods, DST still achieves a far better performance than both WV and MV when the nodes move fast. With a higher mobility of the nodes, it is more difficult for the nodes to exchange their views. Hence, there is higher uncertainty in the network. Since DST is suitable to deal with the problems with uncertainty, the performance downgrade introduced by node mobility is minimized for DST.

Fig. 6 illustrates 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.

Fig. 7 shows the simulation results with different percentage of malicious nodes. It is obvious that DST can yield a much better performance than WV and MV with a higher percentage of malicious nodes. This is true because both WV and MV rely on enough trustworthy information to make a correct decision: MV simply follows the decision from the majority of nodes, and the weights in WV 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 WV and MV degrade noticeably. On the other hand, DST can properly handle the outlier detection problem even in a more hostile environment because it can well deal with unreliability.

5. Conclusion

In this paper, an outlier detection framework is proposed that aims to reveal the malicious nodes in a MANET environment. We apply both the Dempster-Shafer theory of evidence and the weighted voting method to combine observation results from multiple nodes. **The simulation results show that the proposed framework is highly resilient to attackers and it can converge stably to** a common outlier view amongst distributed nodes with a limited communication overhead**.**

One possible future work is to apply the Bayesian inference (BI) method to the view combination process, and compare the performance of DST with that of BI. Since BI has been widely used to fuse together various pieces of evidence, each of these two methods should outperform the other in some circumstances. Moreover, some of the misbehaviors, such as packet dropping, may result from both intentional denial of forwarding and mobility or channel effect. Hence, it will be valuable to try to discriminate malicious nodes from faulty nodes.

6. References

[1] L. Zhou and Z. Haas, "Securing Ad Hoc Networks," IEEE Network, vol. 13, no. 6, 1999, pp. 24–30.

[2] H. Deng, W. Li, and D. Agrawal, "Routing Security in Wireless Ad Hoc Networks," IEEE Comm. Magazine, vol. 40, no. 10, 2002, pp. 70–75.

[3] Y. Hu, A. Perrig and D. Johnson, "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, 2002.

[4] B. Sun, Y. Guan, J. Chen and U. W. Pooch, "Detecting black-hole attack in mobile ad hoc networks", in *Proceedings of 5th European Personal Mobile Communications Conference* , Glasgow, Scotland, UK, pp. 490 – 495, 2003.

[5] A. Perrig, Y.-C. Hu, and D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks", in Proceedings of IEEE Infocom 2003, Pittsburgh, PA, USA, pp. 1976-1986, 2003.

[6] J. R. Douceur, "The sybil attack", Proceedings of the 1st International Workshop on Peer-to-Peer System, Lecture Notes In Computer Science (vol. 2429), Springer-Verlag, pp. 251 – 260, 2002.

[7] F. Grubbs, "Procedures for Detecting Outlying Observations in Samples*",* Technometrics, Vol. 11, No. 1, pp. 1-21, Feb. 1969.

[8] V. Barnett and T. Lewis, "Outliers in Statistical Data", New York, NY, John Wiley and Sons, 1994.

[9] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "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, 2000.

[10] Y. Zhang and W. Lee, "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, 2000.

[11] Y. Huang and W. Lee, "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, 2003.

[12] J. Parker, A. Patwardhan and A. Joshi, "Cross-layer Analysis for Detecting Wireless Misbehavior", in *Proceedings of the IEEE Consumer Communications and Networking Conference*(*CCNC 2006*), Las Vegas, Nevada, USA, pp. 6-9, Jan. 2006.

[13] S. Ramaswamy, R. Rastogi, and K. Shim, "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, 2000.

[14] E. M. Knorr and R. T. Ng, "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, 1999.
[15] S. D. Bay and M. Schwabacher, "Mining Distance-based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule", in *Proceedings of the Ninth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (KDD03)*, Washington, D.C., USA, pp. 29-38, 2003.
[16] C. Aggarwal and S. Yu, "An Effective and Efficient Algorithm for High-dimensional Outlier Detection", *The VLDB Journal*, vol. 14, no. 2, pp. 211-221, 2005.
[17] C. Aggarwal and S. Yu, "Outlier Detection for High Dimensional Data", in *Proceedings of the 2001 ACM SIGMOD international Conference on Management of Data (SIGMOD01)*, Santa Barbara, California, USA, pp. 37-46, 2001.
[18] A. Lazarevic and V. Kumar, "Feature Bagging for Outlier Detection", in *Proceeding of the Eleventh ACM SIGKDD international Conference on Knowledge Discovery in Data Mining (KDD05)*, Chicago, Illinois, USA , pp. 157-166, 2005.
[19] B. Sheng, Q. Li, W. Mao and W. Jin, "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, 2007.
[20] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Distributed Deviation Detection in Sensor Networks", *ACM SIGMOD Record*, vol. 32, issue. 4, pp. 77-82, 2003.
[21] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "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, 2006.
[22] J. Branch, B. Szymanski, C. Giannella, R. Wolff, and H. Kargupta, "In-network Outlier Detection in Wireless Sensor Networks", in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS2006)*, Lisbon, Portugal, 2006.
[23] P. W. Yau and C. J. Mitchell, "Security Vulnerabilities in Ad hoc Networks", *Proceedings of the 7th International Symposium on Communication Theory and Applications*, pp. 99–104, 2003.
[24] H. Deng, Q. Zeng, and D. P. Agrawal, "SVM-based Intrusion Detection System for Wireless Ad Hoc Networks", in *Proceedings of the IEEE VTC03*, vol. 3, pp. 2147–2151, Orlando, FL, USA, 2003.
[25] O.Kachirski and R. Guha, "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, 2002.
[26] C. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "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, 2003.
[27] Y. Huang and W. Lee, "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, 2003.
[28] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "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, 2000.
[29] M. Kefayati, H. R. Rabiee, S. G. Miremadi, and A. Khonsari, "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, 2006.
[30] Y. Xue and K. Nahrstedt, "Providing Fault-Tolerant Ad hoc Routing Service in Adversarial Environments", *Wireless Personal Communication,* vol. 29, issue 3-4, pp. 367-388, 2004.
[31] L. Anderegg and S. Eidenbenz, "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, 2003.
[32] W. Li, J. Parker and A. Joshi, "Security through Collaboration in MANETs", in *Proceedings of the 4th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2008)*, Orlando, FL, USA, Nov 2008.
[33] S. Buchegger, J. Le Boudec, "Performance analysis of the CONFIDANT protocol", in *Proceedings of the 3rd ACM international Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC02)*, Lausanne, Switzerland, pp. 226 - 236, Jun 2002.
[34] A. Patwardhan, F. Perich, A. Joshi, T. Finin, and Y. Yesha, "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 (MASS2005)*, November 2005.
[35] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. R. Rao, "An analytical approach to the study of cooperation in wireless ad hoc networks", *IEEE Transactions on Wireless Communications*, 4(2):722–733, March 2005.
[36] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas, "Robust cooperative trust establishment for MANETs", in *Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN06)*, Alexandria, VA, USA, pp. 23-34, Oct 2006.
[37] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.
[38] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.
[39] Glomosim 2.03, http://pcl.cs.ucla.edu/projects/glomosim/.
[40] T. M. Chen and V. Venkataramanan, "Dempster-Shafer theory for intrusion detection in ad hoc networks", *IEEE Internet Computing*, 9(6):35 – 41, Nov. – Dec. 2005.
[41] M. Raya, P. Papadimitratos, V.D. Gligor, J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks", in Proceedings of 27th IEEE Conference on Computer Communication (INFOCOM2008), Phoenix, AZ, USA, pp. 1238-1246, 2008.