Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 341579, 11 pages
http://dx.doi.org/10.1155/2015/341579

Research Article

A Hierarchical Reputation Evidence Decision System in VANETs

Yang Yang,[1] Zhipeng Gao,[1] Xuesong Qiu,[1] Qian Liu,[1] Yuwen Hao,[2] and Jingchen Zheng[2]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Information Management Center, General Hospital of Chinese People's Armed Police Forces, Beijing 10039, China*

Correspondence should be addressed to Yang Yang; yyang@bupt.edu.cn

Received 25 September 2014; Accepted 5 February 2015

Academic Editor: Ching-Hsien Hsu

In VANETs, users are rational, independent, and selfish. Stimulation-based reputation management system is critical for them to avoid selfishness and promote network performance in large-scale VANETs. The current reputation mechanisms produce some problems, for example, overfull energy consumptions, confused collusion, and misreport. In order to detect selfish and collusive behaviors accurately and quickly, we propose a dynamic three-layer reputation evidence decision and management mechanism, which combine with Dempster-Shafer evidence integration mechanism to distinguish selfish nodes. In particular, the system helps in collusion avoidance through reporting falsified reputation evidences of colluders. In addition, we borrow ideas from Weber-Fechner's law and design an adaptive reputation evidence gathering cycle for prolonging the lifetime of detector and overwhelming frequent polling for reputation evidences. The simulation results demonstrate that REDS has higher detection speed for selfish nodes and collusive observers and less network traffic of gathering reputations.

1. Introduction

Vehicular Ad hoc networks (VANETs) are one important type of the mobile ad hoc networks (MANETs) developed as the basis of Intelligent Transportation Systems (ITS) to provide safer, better, and more efficient road conditions. In VANETs, the main network nodes are the smart vehicles and the road-side infrastructure units (RSUs) that are able to communicate with each other through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Such communications provide a variety of applications ranging from exchanging life-saving information, such as environmental and driving hazards, to traffic congestion, touristic messages, and advertisements.

The V2V and V2I communications are at short distance and high speed in VANETs. Because the communication range of vehicles is limited, packets in VANETs need other vehicles to forward cooperatively. However, in real application scenarios, the vehicles are driven by humans and the human behavioral tendencies are reflected in the behaviors of the nodes. In the event of high-energy consumption and low bandwidth availability, some of the vehicle nodes in the network might refuse to forward other's message packets. Such nodes are called as selfish nodes and they always intend to maximize their own profit, causing undesirable delays in the message delivery and increase the network latency, which in turn affects the entire performance of the network. If a large number of selfish vehicles exist in VANETs, the performance and function of VANETs will be greatly influenced. For these reasons, it is essential to detect selfishness and encourage them in order to promote performance of VANETs. Reputation mechanism is pivotal for node cooperation in packet forwarding in the large-scale vehicular ad hoc networks [1–3]. Through reputation evaluation, selfish behaviors can be discovered and punished in a certain degree.

In order to monitor and isolate misbehaving of packet forwarding in VANETs, a number of reputation technologies have developed to manage reputations and limit the negative impact of selfish nature. For an observed vehicle node n_0, an evaluator (it is usually n_0's one-hop neighbor

which is in n_0's communication range) may use watchdog method, and so forth, to observe n_0's forwarding behaviors and evaluate its reputation, or else the indirect neighbors (it is usually not within n_0's communication range) can adopt recommended and mediate reputations from n_0's one-hop trustworthy neighbors [4, 5]. The direct observation behaviors are usually influenced by channel noise or other adverse conditions. If the observer is interfered, the sole reputation may be inaccurate. In the latter, the final evaluator merges local reputations from reliable recommenders and forms the global reputation in the end. However reputation recommendation may cause some problems, such as overfull cost due to periodical reputation information exchanging, reputation storage, and management because of movement of vehicles and so on. It is difficult to distinguish exactly collusion from misreport and further to punish conspirators.

The network traffic consumption by continually reporting reputation data is another major concern problem. In VANETs, vehicle nodes belong to different individual. The users are not willing to consume extra traffic or resources for periodically reporting local reputations in order to avoid influencing personal applications. In addition, reputations from different recommenders on the different period may cause local reputations confusion (see Section 3.2). Calculating reputation based on a reasonable reporting cycle is essential for saving network traffic and resources and making reputation more accurate.

Reputation management needs to better suit VANETs and increase the reputation accuracy, and thus we propose a hierarchical reputation evidence decision system (REDS) based on the Dempster-Shafer evidence theory. The main merit of Dempster-Shafer evidence method [6, 7] is that it combines different evidences, especially some uncertainties from different evidence sources. Uncertain evidences are used to denote reputations caused by vehicle mobility and channel noise and further promote reputation accuracy and mitigate the adverse impact of network characteristics.

REDS is a dynamic three-layer reputation evidence combination structure. In the lowest coordination observer decision layer, some of n_0's 1-hop neighbors which stay within n_0's communication range during a complete reputation gathering cycle are regarded as coordination observers. They detect and report n_0's forwarding behaviors as reputation evidences. The intermediate coordination manager combines the evidences reported from the coordination observers in its own detection domain using evidence theory and sends merged data to n_0's host manager in the highest layer. In the end, the host manager calculates several reputation evidences and correspondingly forms n_0's credit. If n_0's credit is negative, it is selfish.

One of the advantages of the paper is that colluders can also be detected through reputation evidence feedback. Colluders usually report higher reputations for each other and decrease other's reputations. To avoid collusion, the host manager feeds the ultimate reputation back to different layers' evaluators. The upper evaluator can manage the trustworthy degree of evaluators of lower layer. The trustworthy degrees of evidence directly influence evaluators' credits. If an evaluator's reputation evidences are always deviated from others, it tends to be a colluder, and then its credit is also negative in the end.

In addition, the responsibility for transmitting reputation evidences leads evaluators to consume more network traffic. So we design an adaptive evidence gathering cycle in views of vehicles' mobility and personal applications.

In detail, our contributions are as follows.

(1) The dynamic three-layer reputation evidence combination system: in the current reputation management technologies [8–10], the reputation manager ignores the reputation which deviates dramatically from the average value in order to avoid falsified reputation. It leads the accurate reputation to be abandoned when large-scale collusions exist. Our proposed scheme combines Dempster-Shafer evidence with hierarchical evidence clustering on the dissimilar evaluation layer to calculate more accurate reputations and detect selfish vehicle nodes in less time than other typical reputation algorithms. Especially it calculates the trust degree of evaluators and therefore effectively isolates collusive coordination observers that conspiratorially report higher reputation for selfish nodes.

(2) Adaptive reputation gathering cycle: too frequent polling reputation evidences will cost plenty of network traffic. An adaptive reputation gathering cycle based on Weber-Fechner's law is proposed in order to maintain the stability of reputation management system and save network traffic. Considering vehicles' movement and personal applications, dynamic changing cycle can seek a balance between reputation valuation accuracy and network traffic.

The rest of papers are organized as follows. Section 2 provides some related works about the reputation cooperation. Section 3 introduces our hierarchical reputation evidence decision system (REDS), and Section 4 depicts the simulation results with respect to typical reputation evaluation algorithms and demonstrates our algorithm's efficiency and superiority. In Section 5, we conclude the paper.

2. Related Works

Some research works have been done on vehicle stimulation cooperation in VANETs at present. Related research of MANETs is relatively abundant. The stimulation cooperation mechanisms in the MANETs are generally classified into virtual currency-based and reputation-based stimulation system. In the virtual currency system, nodes pay virtual money (called nuglets) [11] for forwarding service. In detail, a source pays for each forwarding node on the routing path. And each node also needs to provide forwarding service to gain nuglets. The previously mentioned price mechanism is not flexible enough to maximize cooperators' profits. For example, a source hardly evaluates accurately the total of nuglets in the dynamic changing network environment. Some researchers introduce economic incentive technology in the self-organized ad hoc networks. To maximize the payoff of relay nodes, Ji et al. [12] propose a game pricing-based routing

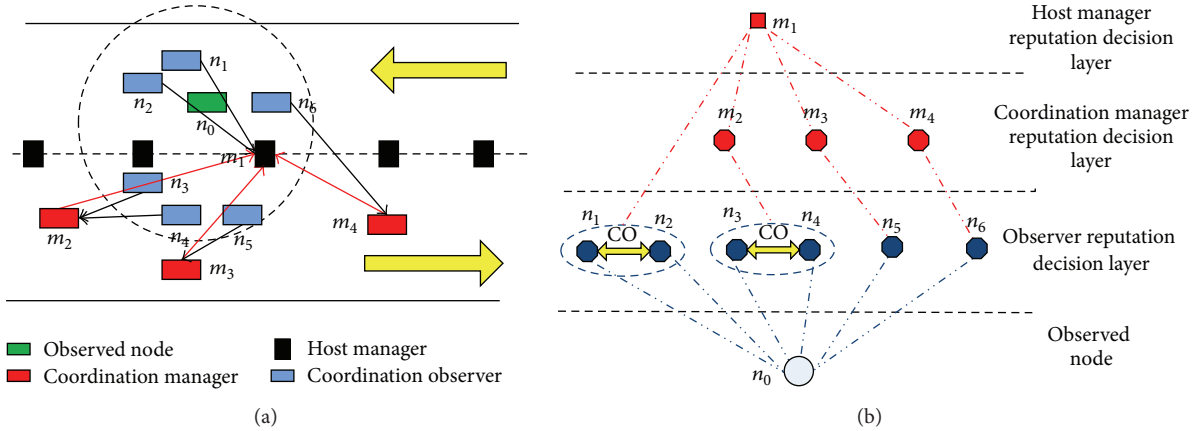

Figure 1: (a) Reputation-based network topology. (b) Three-layer reputation decision architecture.

mechanism. Packet-forwarding services are negotiated and auctioned in the orthogonal frequency-division multiplexing setting. Mukherjee and Kwon [13] design a robust multiobject bundled auction method for simultaneous partner selection. After successful auction, a contractor (winning bidder) achieves satisfied QoS and rewards the service provider (seller).

Reputation management systems are adopted to detect and isolate misbehaving nodes to release their negative influence on the network performance. In the light of adverse impact of misbehaving nodes, the paper [14] presents the watchdog and detects uncooperative relay nodes. Detecting node maintains a buffer to store the number of neighbors' received and sent packets. But it is probably influenced by channel collision. MobiGame [15] designs a user-centric reputation incentive system for delay tolerant network (DTN). Packet forwarding's cost and reward can get a Bayesian equilibrium through a game-theoretic scheme. Refaei et al. [1] represent a time-slotted evaluation approach to detect timely node's changing behaviors. In addition, the sequential probability ratio test (SRRT) is used for judging whether a neighbor is selfish. Anantvalee and Wu [9] define two thresholds, TH_{coop} and $TH_{selfish}$ ($TH_{coop} > TH_{selfish}$) to classify cooperative and selfish nodes, and then encourage suspicious nodes whose reputation in the range $[TH_{selfish}, TH_{coop}]$ to cooperate.

For the reputation store, DHT Trust overlay network [16] uses CHORD to distribute local reputation to reputation manager. Finger nodes issue reputation feedback in the certain interval and fast aggregate local reputation into the global reputation. Li and Shen [8] propose the account-aided reputation management system (ARM) to stimulate selfish mobile nodes. The ARM system calculates reputations and credits to distinguish selfishness from cooperation. In particular, the authors introduce the distributed hash table (DHT) to store circulating reputations and credits. As node mobility, DHT uses a lightweight maintenance protocol to reduce the number of reputation structure reestablishment.

Although the above-mentioned references have provided incentive mechanisms for selfish nodes in the wireless self-organized network, there are some deficiencies that need to be solved further, for example, reputation imprecision due to vehicles' mobility, reputation falsifying caused by collusion, and so forth. And thus, our REDS effectively promotes reputation accuracy by using hierarchical reputation evidence decision and distinguishes suspect collusion vehicle nodes and decreases their trustworthy degree through reputation feedback.

3. Reputation Evidence Decision System

In the paper, we propose a dynamic three-layer reputation evidence decision system. As illustrated in Figures 1(a) and 1(b), the hierarchical structure contains all of mobile vehicles that are classified into 4 roles: host manager (HM), coordination manager (CM), coordination observer (CO), and observed nodes. An observed node n_0 has only one host manager (HM). HM is a special type of infrastructure called Road Side Unit (RSU), which is deployed along the road for connectivity. They are deployed in specific areas and completely cover the whole VANETs. HM is regarded as a trustful node, in charge of merging reputation evidences about the observed node and forming the final reputation. CMs are also considered to be trustful in our paper. They are the coordination managers in the second layer. CMs take the charges of reputation decision, merge the reputation evidence from Cos, and send the local synthetically reputation evidence to HMs.

In addition, n_0 has a lot of 1-hop neighbors (vehicles which stay within the range of n_0's communication radius during a complete reputation gathering cycle), part of which are COs. For example, suppose that n_0 is an observed node and a vehicle node n_1 runs passing n_0. If the time from n_1 enters n_0's communication range to the time n_1 runs out of n_0's communication range is not less than the reputation gathering cycle, then n_1 can be regarded as a coordination observer. In the intermediate layer, several n_0's one-hop neighbors, which stay within the range of n_0's communication radius no less than two complete reputation gathering cycles,

are regarded as coordination managers. That is, if a coordination observer n_1 stays within the range of n_0's communication radius no less than two complete reputation gathering cycles, then it upgrades itself to a coordination manager. It is obvious that the coordination manager of one observed node has relatively similar motion state with the observed node.

However, if a CO or a CM runs out of the observed node's communication range, it downgrades itself to a normal node, and will be no longer engaged in reputation gathering in the current round until it enters the observed node's communication range again. In a reputation gathering cycle, each CO chooses the nearest CM in n_0's communication range, detects the statistical value with respect to n_0's forwarding behaviors as reputation evidences through vehicle wireless communication devices, and reports the reputation evidences to the CM. However, a CM manages several COs and merges the evidences from COs in its management domain and sends to n_0's host manager (the highest layer). For a running observed node n_0, if there is no CO in its communication range, then CMs detect the observed node's forwarding behaviors directly. If there is no CM in its communication range, then COs report the reputation evidences directly to HM. Each observed node has its own host manager that is responsible for calculating the account of forwarding services.

An example depicted in Figures 1(a) and 1(b) is given as follows. It is assumed that the observed vehicle node n_0 has 6 coordination observers which sequence from n_1 to n_6. m_2 is the nearest CM to n_3 and n_4, so n_3 and n_4 are managed in the same domain of m_2. Here m_2, m_3, and m_4 are viewed as m_1's CO. The host manager of n_0 is m_1, which is the nearest road-side unit to n_0. In the lowest layer of reputation evidence decision (coordination observer reputation decision layer), each coordination observer records the received and forwarded packets of n_0, judges the uncertain conditions (may be caused by wireless channel unconventionality) by inquiring the coordination observer, and forms the basic belief assignment (BBA) values (see Section 3.1 for more explanations) corresponding to the uncertain conditions. According to n_0's behaviors and coordination observer's monitoring value, the coordination observer reports BBA values to its upper manager. They transmit the BBA value to m_2.

In the middle-layer of reputation decision, m_2 merges the reputation evidences from n_3 and n_4 and sends the local synthetic reputation evidence to m_1. As n_0's host manager is m_1, so m_1 combines different reputation evidences reported by several coordination managers (m_2, m_3, m_4) and reputations reported from n_1 and n_2. In the end, m_1 calculates the comprehensive value in the highest reputation decision layer.

Some reputation evaluation algorithms ignored the high deviated reputation which may be caused by misreport or falsification. On the one hand, the benign reputation may be abandoned in the large-scale collusion environment and then the benign node is classified into selfish. On the other hand, they have no restraint measures to punish colluders with falsified reports in many times. These falsified or misreporting nodes are unsuitable to continue to serve as COs. So we introduce the Dempster-Shafer Evidence theory [17, 18] into reputation decision in order to calculate reputation more accurately. Through reputation feedback on COs and CMs, we can fast detect selfish node and colluders.

Moreover, we design a dynamic reporting cycle for coordination managers and coordination observers according to their mobility and resources. This is because they will consume extra network traffic for collecting and merging evidences. However mobile vehicles belong to different individual users, and then the extra network traffic consumption will influence user's personal applications. It means the reputation managers should save extra network traffic consumption.

3.1. Reputation Evidence Combination. In the Dempster-Shafer evidence reasoning mechanism, evidences are denoted as some possible events. Using reasoning combination rule to aggregate multiple belief evidences under uncertainty. To calculate an accurate comprehensive reputation, we consider effect of each reputation evidence rather than simply ignore the value that deviates largely from the average. We design the dynamic three-layer reputation evidence aggregation structure for reputation evaluator at different layers. And furthermore, the system executes reputation feedback to amend each evaluator's trust weight and distinguish colluders.

In the coordination observer reputation decision layer, a set of hypotheses about observed forwarding behaviors is denoted as a frame of discernment $\Theta = \{O, \overline{O}\}$. O indicates that the observer has monitored the forwarding events of the monitored node and $\overline{O}$ indicates that the observer does not monitor any forwarding behaviors. In the following, the power set 2^Θ includes all the subsets of Θ. Here $2^\Theta = \{\{\Phi\}, \{O\}, \{\overline{O}\}, \{O, \overline{O}\}\}$, each symbol of which respectively represents the hypotheses about *impossible, forwarded, nonforwarded,* and *uncertainty*.

Observer i calculates n_0's BPA function per unit time as

$$m : 2^\Theta \longrightarrow [0, 1]$$
$$m_{i,j}^{(n)}(\Phi) = 0. \tag{1}$$

The BPA for proposition O is

$$m_{i,n_0}^{(n)}(\{O\}) = \frac{NF_{i,n_0}^{(n)}}{NR_{i,n_0}^{(n)}}, \tag{2}$$

where $NF_{i,n_0}^{(n)}$ and $NR_{i,n_0}^{(n)}$ denote observer i recording the number of packets that n_0 forwarded and received at nth unit time in the normal wireless network environment when observer i runs in n_0's communication range. However wireless network conditions possibly cause adverse interferences (e.g., buildings' obstruction, moving, channel noise, collision, congestion, etc.), which lead to monitoring incompletely. Define uncertainty being aroused by wireless channel unconventionality. When an observer i detects nonforwarding events corresponding to node n_0 caused by adverse wireless environment in a period, then it contacts any coordination observer k which runs in the same CM's domain. Observer i acquires n_0's forwarding data recorded by k in i's jammed

period and treats it as uncertainty value. Observer i determines whether the observed value is equal to k's observed value. If it is, it means that detection behaviors are not affected by network environment. It is assumed that $AF_{i,n_0}^{(n)}$ and $AR_{i,n_0}^{(n)}$ indicate that observer i records the number of packets that n_0 forwarded and received in the abnormal period. Consider $AF_{i,n_0}^{(n)} = AF_{k,n_0}^{(n)}$ and $AR_{i,n_0}^{(n)} = AR_{k,n_0}^{(n)}$. Consider

$$m_{i,n_0}^{(n)}(\{O\}) = \frac{NF_{i,n_0}^{(n)} + AF_{i,n_0}^{(n)}}{NR_{i,n_0}^{(n)} + AR_{i,n_0}^{(n)}}. \quad (3)$$

If less, it means that i's monitoring data has been affected by network environment. And then i should utilize k's observed value in its abnormal period. But in order to avoid collusion, i will incompletely believe k's observed value and multiplies that value by a discount factor α, or else, i abandons k's observed value because k may fall into wireless channel unconventionality. For example, i gets k's detected total number of forwarded and received packets in the anomalous period and then BPA for proposition $\{O, \overline{O}\}$ is

$$m_{i,n_0}^{(n)}\left(\left\{O,\overline{O}\right\}\right) = \frac{\alpha \cdot AF_{k,n_0}^{(n)}}{NR_{i,n_0}^{(n)} + AR_{k,n_0}^{(n)}}. \quad (4)$$

Here the discount factor α is defined as follows:

$$\alpha = \frac{NF_{i,n_0}^{(n-1)}}{NR_{i,n_0}^{(n-1)}}, \quad (5)$$

$$m_{i,n_0}^{(n)}\left(\left\{\overline{O}\right\}\right) = 1 - m_{i,n_0}^{(n)}(\{O\}) - m_{i,n_0}^{(n)}\left(\left\{O,\overline{O}\right\}\right).$$

A coordination observer may be colluder that reports a higher reputation for selfish nodes. Once a CO is detected to be a colluder, it will be not suitable for reputation gathering and decision even though it still runs within the observed node's communication range. These collusive COs will be punished and isolated and will not be permitted to upgrade themselves to be CMs. To detect colluders, m_1 calculates the trust weights of coordination observers and coordination managers which run within the range of n_0's communication radius during the current reputation gathering cycle and further feedbacks them, respectively, to coordination managers (Figure 2). Define the ultimate belief reputation calculated by the host manager as follows:

$$e^* = \left\langle m^*(\{O\}), m^*\left(\left\{\overline{O}\right\}\right), m^*\left(\left\{O,\overline{O}\right\}\right), m^*(\Phi)\right\rangle. \quad (6)$$

Define the direction cosine between any evidence by any evaluators e_i and e^* as follows:

$$\rho_i = \cos(e_i, e^*) = \frac{e_i^T e^*}{\|e_i\| \cdot \|e^*\|} = \frac{e_i^T e^*}{\left[\left(e_i^T e_i\right)\left(e_i^{*T} e^*\right)\right]^{1/2}}. \quad (7)$$

The trust weights of evaluators (coordination observers and managers) are defined as follows:

$$\lambda_i = \frac{\rho_i}{\sum_n \rho_n}. \quad (8)$$

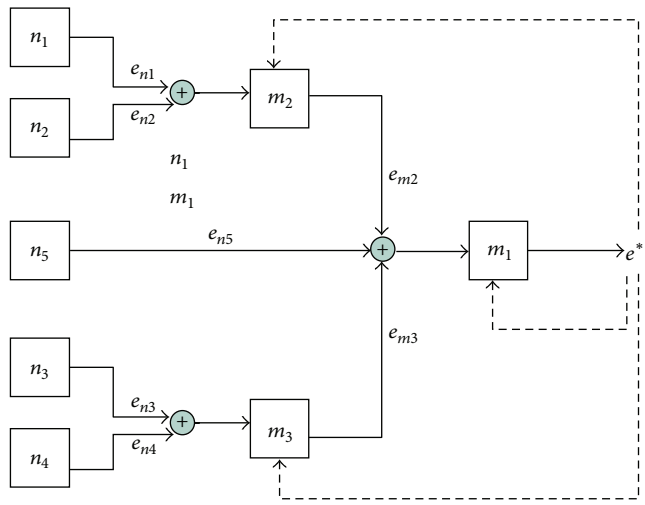

Figure 2: Feedback for trustworthy weights.

Through evaluation of trust weights of COs and CMs, a reputation evidence reported by a CO or a CM will be influenced:

$$m_i(A) \longleftarrow m_i(A) \cdot \lambda_i \quad \left(\forall A \in 2^{\Theta}\right). \quad (9)$$

Define the reputation evidence combination rule based on reputation evidence feedback as follows:

$$m_{ij}(A) = m_i \oplus m_j(A) = \frac{1}{1-k} \sum_{X \cap Y = A} m_i(X)\, m_j(Y). \quad (10)$$

Here k is a normalized constant. Consider $k = \sum_{X \cap Y = \emptyset} m_i(X) m_j(Y)$. A new reputation evidence is deduced through combining two BPAs m_i and m_j. But the typical D-S algorithm may cause evidence conflict; for example,

$$\begin{aligned} &m_1 : m_1(A) = 0.99, \quad m_1(B) = 0.01, \quad m_1(C) = 0 \\ &m_2 : m_2(A) = 0, \quad m_2(B) = 0.01, \quad m_2(C) = 0.99 \\ &k = 0.9999, \quad m(A) = m(C) = 0, \quad m(B) = 1. \end{aligned} \quad (11)$$

The combined evidence represents the probability of event B being 1 although BPAs of m_1 and m_2 about B are 0.01.

To further avoid evidence conflict, we improve D-S evidence rule integrated with hierarchical clustering theory besides adding trust weight for each reputation evaluator. In the coordination manager reputation decision layer, a coordination manager receives the coordination observers' belief reputation value in its own domain. If some evidence equals 0, then we use the hierarchical clustering mechanism to combine evidences to avoid evidence conflict, or else it uses weighted D-S evidence combination rule.

It is assumed that the coordination manager acquires N evidences. Classify N clusters according to N evidences, and each cluster only includes a reputation evidence. When combining reputation evidences by using hierarchical clustering

Begin $D_{ij} = 0, D_{kl} = 0$,
Do While n evidences are combined as a compositive conclusion
For $i = 1$ to N
For $j = 1$ ***to*** N
Calculate D_{ij} according to formula (12);
End For
End For
D_{kl} = minimize (D_{ij});
If $\exists m_i(A_j) = 0$ $(i \in \{k, l\}, j = \{\{O\}, \{\overline{O}\}, \{O, \overline{O}\}\})$
Then
Combine reputation evidences based on hierarchical clustering (formula (10)–(12));
Else
Use reputation D-S evidence combination (formula (7)–(9)) for integration. And m_{ij} is regarded as a new gravity center if using hierarchical evidence clustering next combination.
End If
End While
End

Pseudocode 1

theory, the distance between evidences is defined as distance between their gravity centers. We assume the gravity centers of clusters ω_p (each cluster only includes an evidence) and ω_q are x_p and x_q, and the number of the samples in them is n_p and n_q, respectively. Then we combine ω_p and ω_q as a new reputation evidence cluster ω_i; thus ω_i would own $n_p + n_q$ samples. Cluster ω_i's gravity center would be as follows:

$$x_i = \frac{1}{n_p + n_q}\left(n_p x_p + n_q x_q\right). \tag{12}$$

Assume that another evidence cluster ω_j's gravity center is x_j, the Euclidean distance D_{ij} between clusters ω_i and ω_j is defined as follows. D_{kl} is the minimal value of D_{ij} $(\forall i, j)$:

$$D_{ij} = \sqrt{\left(x_i - x_j\right)^T \left(x_i - x_j\right)}. \tag{13}$$

The pseudocode of reputation evidence calculation is shown in Pseudocode 1.

In the host manager reputation decision layer, the host manager needs to know the relative degree about each coordination manager when evidence combination. Namely, it calculates the direction cosine between e^* and e_{mi} reported by each coordination manager and further gets the trust weights of these coordination managers. And then it uses the same combination mechanism and integrates reputation evidences per unit time reported by different coordination managers.

High-reputed and low-reputed nodes should be rewarded or punished. Define a selfish character factor S_o (is set as initial number) for each observed node. The host manager calculates the global reputation R_g and relation between S_o and R_g is

$$S_o \longleftarrow S_o + (-1)^r \cdot \left|0.5 - R_g\right|. \tag{14}$$

Here $r = 0$ when $R_g \geq 0.5$; otherwise $r = 1$. When $S_o < 0$, then the observed node is completely selfish and must be isolated.

Colluders report high reputation for selfish nodes and low reputation for cooperators, so it also expects its ally to report high reputation in return. It can be regarded as selfish in certain extent. Given a trust weight of the coordination observer i, its selfish character factor will decline along with the decreasing trust weight. The punishment measure for a colluder j is as follows (n is the current iteration):

$$S_0 \longleftarrow S_0 - \left(\epsilon - \lambda_j^{(n)}\right). \tag{15}$$

Here S_o is only used for judging the selfishness and ϵ is an initial value. Credit that stimulates nodes to provide or share more forwarding services is defined as follows. If S_o of a node is less than zero, its credit correspondingly is cleared:

$$C_i(n) = C_i(n-1) - p_i + R_{g_i} + c_i. \tag{16}$$

Here $C_i(n)$ is the observed node i's credit in the process of nth reputation evaluation period of the host manager. p_i is the paid credit of i for sharing forwarding service. c_i is cost factor for consuming extra network traffic for reputation evidence report and combination. It is because the coordination observers consume extra network traffic for reputation report and combination that the corresponding credits should be assigned to them.

3.2. Adaptive Reputation Gathering Cycle. However polling may cause more network traffic consumption if vehicle nodes gather reputation more frequently. In VANETs, selfish vehicles [19, 20] pursue the least cost to reduce network traffic consumption. But the responsibility for gathering and merging reputation leads evaluator to consume more. It should seek a balance between gathering frequency and consumption of network traffic. Therefore, each neighbor vehicle node which runs within the range of n_0's communication radius needs to determine an adaptive gathering cycle according to its current applications in the network, and then according to the new reputation gathering cycle and

their velocities, vehicle nodes determine whether they can be regarded as COs or CMs. The number of current applications reflects traffic load and remaining resources of evaluators. The more applications an evaluator has, the longer the gathering cycle is set.

In the paper, we improve Weber-Fechner's law [21] in order to update gathering cycle that is subject to a variety of environment stimulating factors, such as the number of current applications and mobility of vehicles. Weber-Fechner's law uses a linear function of logarithm for describing the relationship between individual's response and incitement due to external environments. In the following formula, S denotes the intensity of sensation, R the magnitude of stimulation, and K a constant. Consider

$$S = K \ln R. \tag{17}$$

Assume that Will is the response degree of changing gathering cycle, which corresponds to VANETs environment reaction such as the number of current applications A and the velocity of vehicles V. These two influence factors can be considered as environment stimulating factors, and Will reflects the corresponding responses of changing gathering cycle. $\text{Will}^{(n)}$ should be recalculated before the nth cycle. As an evaluator is also a common mobile vehicle node, its traffic loads and computational resources are subjected to a variety of applications. The larger the number of applications executed on a vehicle node, the more traffic will be occupied. The gathering cycle will be prolonged companying with remaining traffic decreasing. As vehicles move quickly which makes the topology of VANETs change frequently. When a vehicle node moves at a high speed, its surroundings change frequently. To gain more accurate reputation, the gathering cycle should be moderately reduced. ω_{ij} $(j = 1, 2)$ is the weight of each environment stimulating factors:

$$\text{Will}^{(n)} = \omega_1 \ln A^{(n)} + \omega_2 \ln \frac{1}{V^{(n)}},$$

$$\Delta\text{Will}^{(n)} = \text{Will}^{(n)} - \text{Will}^{(n-1)} = \omega_1 \ln \frac{A^{(n)}}{A^{(n-1)}} + \omega_2 \ln \frac{V^{(n-1)}}{V^{(n)}}, \tag{18}$$

where $\text{Will}^{(n)}$ reflects the corresponding responses of changing gathering cycle, $A^{(n)}$ is the number of current applications, and $V^{(n)}$ is the velocity of vehicles. $\Delta\text{Will}^{(n)}$ is the just noticeable difference (JND) for changing the cycle according to Weber-Fechner's law. If $\exists|\Delta\text{Will}^{(n)}| > \text{Will}^{(n-1)} \times k_{\text{web}}$ (k_{web} is Weber fraction, generally $k_{\text{web}} = 1/30$), then evaluator's gathering cycle increases or decreases 1 minute. Otherwise, the cycle is inalterable:

$$T = \begin{cases} T + 1, & \Delta\text{Will}^{(n)} > \text{Will}^{(n-1)} \cdot k_{\text{web}}, \\ T - 1, & \Delta\text{Will}^{(n)} < -\text{Will}^{(n-1)} \cdot k_{\text{web}}, \\ T, & \text{Else}. \end{cases} \tag{19}$$

For example, $A^{(n-1)} = 10$, $A^{(n)} = 12$, $B^{(n-1)} = 0.3$, and $B^{(n)} = 0.25$. Consider $\text{Will}^{(n)} = 1.9356$, $\text{Will}^{(n-1)} = 1.7533$, $\text{Will}^{(n)} = 0.1823 > \text{Will}^{(n-1)} \times k_{\text{web}} = 0.0584$; then gathering cycle adds 1 minute.

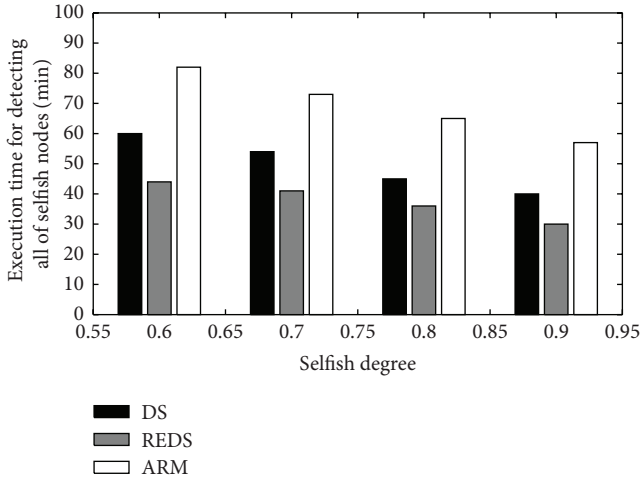

Figure 3: The time for detecting all selfish observed nodes.

4. Simulation Analysis

We conduct simulations to demonstrate the performance of REDS in our paper. The simulated VANETs include 60/120 vehicle nodes randomly deployed in the 1 km driveway area. The velocity of each vehicle is dynamically changed and is set randomly from 50 km/h to 100 km/h. As shown in Figure 1, the RSUs are deployed along the road and regarded to be HMs. Different roles of vehicle nodes such as COs or CMs are determined by vehicle nodes themselves according to the dynamic changing gathering cycles and vehicle's velocities. We assume all of managers are trustworthy, while coordination observers maybe collude with observed nodes in order to report high reputation evidences for them. Colluders are divided into random collusion and group collusion. Group collusion means all of coordination observers in a group advisedly overwhelm other groups. The initial credits of all of observed nodes (ONs) and COs are equal to 2. Physical bandwidth is set to 2 Mbit/s. We compare our algorithm with ARM [8] and purely D-S Proof Fusion mechanism.

The ARM system calculates reputations and credits to distinguish selfishness from cooperation. In particular, the authors introduce the distributed hash table (DHT) to store circulating reputations and credits. As node mobility, DHT uses a lightweight maintenance protocol to reduce the number of reputation structure reestablishment.

Firstly we verify the validity of selfish node detection when the network size changes. In Figure 3, there are 10 selfish observed nodes (ONs) and 10 cooperative ONs. In the communication ranges of these observed nodes, we set 15 collusive nodes and 25 cooperative nodes (include ONs' host managers). We assume that all collusive nodes and cooperative nodes have qualification to be COs, whereas all managers (including HMs and CMs) need to be cooperative and trustworthy. That is to say, a collusive CO cannot upgrade

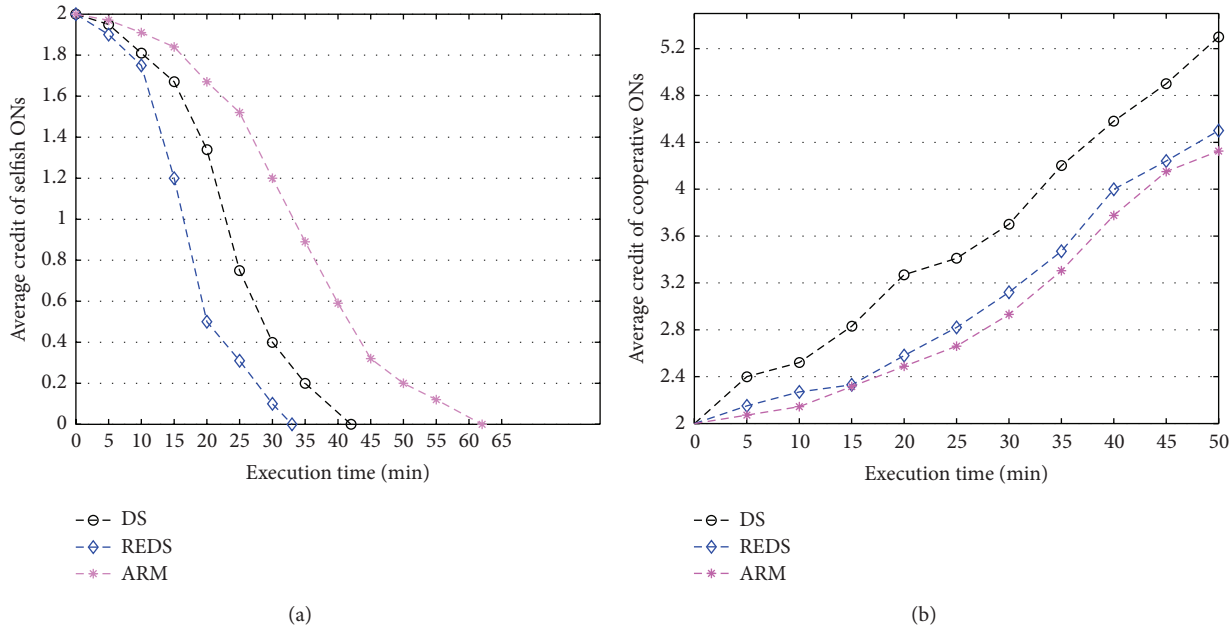

Figure 4: (a) Average credit of selfish ONs. (b) Average credit of cooperative ONs.

itself to a CM even though it runs within the corresponding observed node's communication range all the time.

Figure 3 shows the time for detecting all of selfish observed nodes. Here selfish degrees (ON's drop packet rate) are defined in [0.6, 0.9]. REDS has the minimum detection time for all of selfish nodes. The greater the selfish degree is, the less the detection time is taken. While in the figure we see a comparative result in D-S when no one ticket veto situation happens, and ARM shows the worst performance in this situation. As ARM only simply ignores anomalous reputations rather than punishing colluders, it is difficult to distinguish cooperators from collusive COs in the large-scale group collusion. And then ARM consumes the longest time for detecting selfishness.

In Figure 4, we define, respectively, the selfish degree of the coordination observer nodes in [0.8, 1] (Figure 4(a)) and [0, 0.2] (Figure 4(b)) and evaluate their credits according to the simulation setting as Figure 3. Each ON's initial credit is set to 2. For selfish ONs in REDS, the average credit decreases slowly in the start detection stage. That is because the reputation feedback has not distinguished all of collusive COs. As time goes on, collusive COs are detected one by one (shown in Figures 6–9). Based on this, the selfish ON's credits (shown in Figure 4(a)) decrease quickly because less collusive COs report fake high reputation evidences. REDS and DS act more steeply which means that they detect selfishness more quickly than ARM. ARM puts selfish nodes into blacklist but has no ability to detect collusive COs. The selfish nodes' reputation evaluations are always influenced by falsified high reputation reported by collusive COs.

Figure 4(b) represents average credit of cooperative ONs. Selfish observed nodes are detected as quickly as possible in the REDS, and thus the system has a higher throughput than others. As more trustworthy nodes are selected to be forwarding node, the credits of cooperative ONs have stably increased compared with DS and ARM.

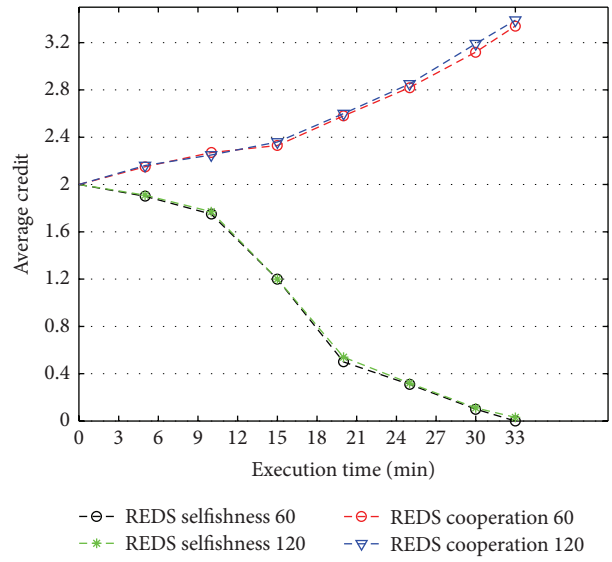

Figure 5: Average credits change over different network size.

In Figure 5, we evaluate whether the average credit of REDS would be influenced by the network size or not. The simulation executes 33 minutes, respectively, with different network size (60 or 120). The number of selfish nodes is 1/6 of the total. The total of cooperative ONs follows the number. We can see that the average credit of selfish observed nodes has decreased to zero. The credit's slope with 60 nodes is similar with 120 nodes, which means REDS executes stably regardless of the network size. When the rate of number of

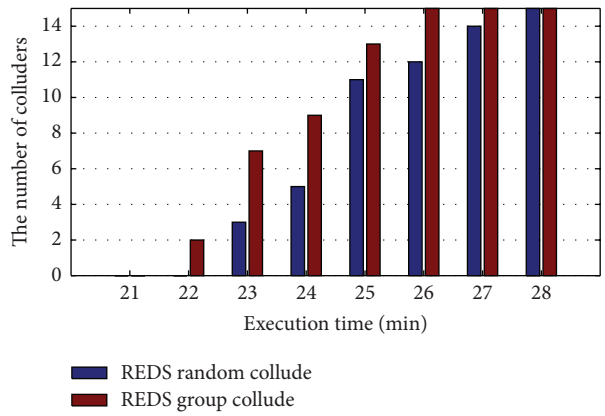

Figure 6: The detection time for collusive COs.

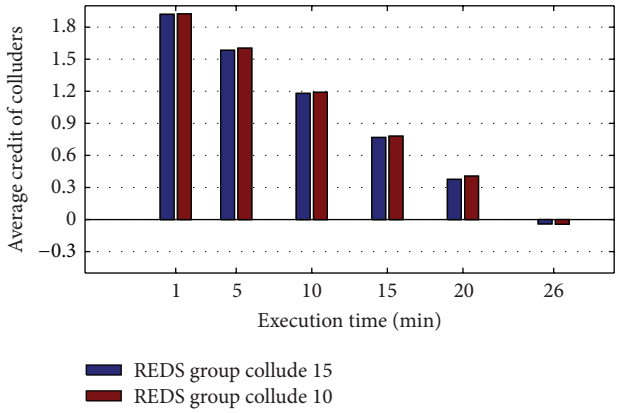

Figure 7: Average credit of group colluders.

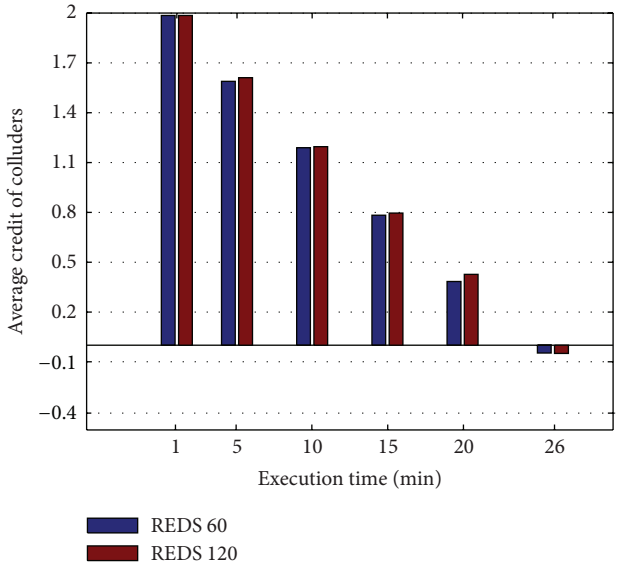

Figure 8: Average credit of colluders over different network size.

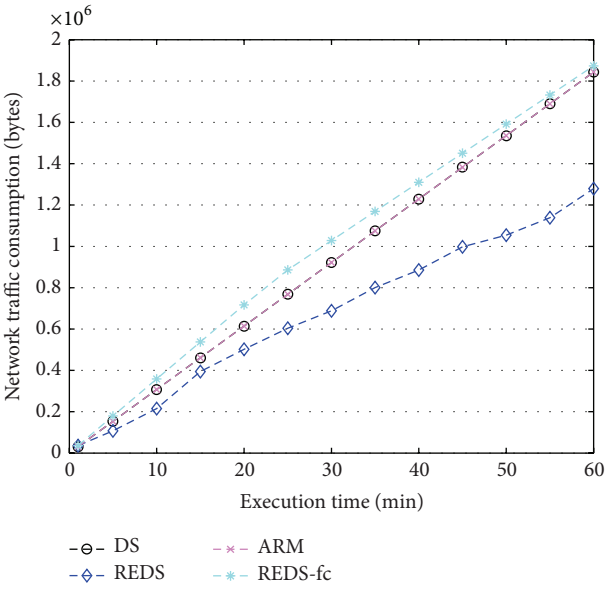

Figure 9: Network traffic consumption.

COs to number of ONs is changeless, the declining ratio of credit is also stationary.

We will verify the effectiveness of collusive COs' detection in REDS when the network size is 60. Define 15 collusive nodes are randomly distributed and they have equal change to be COs. In each coordination managers' domain, 2~3 random colluders are included at most in the random collusion mode. Group collusion mode represents the network including several random colluders and 1~2 collusion groups. All of coordination observers in each collusion group agree with selfish observed nodes to report falsified high reputation evidences.

Figure 6 demonstrates the detection time for collusive coordination observers. We can see that group collusion detection is slightly faster than random collusion in REDS. Through reputation feedbacks on COs and CMs, REDS amends their trust weights. We have assumed that all of coordination managers are trustworthy. As to group collusion, it is easier for CM to evaluate the behaviors of all of coordination observers in its own management domain.

Figures 7 and 8 verify the stability of REDS in the conditions of random collusion and group collusion. The network size is 60, and the total of colluders (including group and random colluders) is 15. Figure 7 represents the average credit of colluders when the number of group colluders is 10 (5 random collusion, in addition) or 15 (no random collusion in the network). We can see that average credit decreasing speed of 15 group colluders is 0.088 per minute, slightly faster than 10 group colluders (0.069 per minute). According to the results of Figure 6, group colluders are detected in less time due to the trustworthy weights of group colluders and their high-level coordination managers are always lower than others.

Figure 8 presents observation results of average credit of colluders over different network size (60 or 120). If all

collusive vehicles become COs, then the number of collusive COs is 1/4 of the total. We adopt reputation feedback and credit update in REDS, and each observed node insures more than 5 COs to detect itself though the network size enlarges. REDS detects colluders in a stable speed and approximately regardless of network size in the distributed conditions.

Figure 9 represents the network traffic consumption in different reputation evaluation system. The simulation settings are followed as in Figure 3. The packets size of reputation evidence and weight feedback is 512 bytes. For simplicity, the following algorithms do not add more nodes for new ONs after removed selfish ONs and collusive COs to blacklist. It means that they only continue to detect old cooperative ONs.

Curves of DS and ARM have the same gradient and Y-value at the same time. ARM collects local reputation and calculates global reputation in the end. DS calculates the comprehensive reputation evidence in the light of similar rule. DS detects selfish observed nodes faster than ARM according to Figure 3, and totally it shows a lower consumption than ARM after 43 minutes. REDS-fc means the simulation results adopt reputation evidence feedback as REDS, but coordination managers gather reputation evidences at stable period. It produces more extra network traffic consumptions than others because trustworthy weight feedback packets are distributed to coordination managers in fixed cycle. REDS determines an adaptive gathering cycle according to evaluators' applications and velocities in the network. The changing gathering cycle can effectively reduce gathering consumptions though adopting feedback evidence mechanism. So REDS makes better performance than others.

5. Conclusion

In the paper, we propose a three-layer reputation evidence decision system (REDS) to detect misbehaving nodes in VANETs. REDS can distinguish fraudulent information from real reputation evidences and avoid credits of cooperative nodes being affected by falsified information. Collusive coordination observers usually conspiratorially report fraudulent reputation evidences in a random or group collusion way. If only ignores highly deviated information rather than punishes premeditated reporters, collusion will always exist. We feed back trust degree of each coordination observer to its coordination manager, thus helping reputation evidence combination and collusion detection. The credits of the coordination observers decreased or increased according to the results of their trust degree weights. Moreover, an adaptive reputation evidence gathering cycle is proposed to replace frequent polling mechanism and save the network traffic. The simulation results demonstrate REDS having high performance of detection for selfish and collusive behaviors.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partly supported by NSFC (61372108, 61401033), Ph.D. Programs Foundation of Ministry of Education of China (no. 20110005110011), Fundamental Research Funds for the Central Universities (no. 2014RC1102), and Beijing Higher Education Young Elite Teacher Project (YETP0474).

References

[1] M. T. Refaei, L. A. DaSilva, M. Eltoweissy, and T. Nadeem, "Adaptation of reputation management systems to dynamic network conditions in ad hoc networks," *IEEE Transactions on Computers*, vol. 59, no. 5, pp. 707–719, 2010.

[2] T. Chen, F. Wu, and S. Zhong, "FITS: a finite-time reputation system for cooperation in wireless ad hoc networks," *IEEE Transactions on Computers*, vol. 60, no. 7, pp. 1045–1056, 2011.

[3] S. Wang, C. Fan, C.-H. Hsu, Q. Sun, and F. Yang, "A vertical handoff method via self-selection decision tree for internet of vehicles," *IEEE Systems Journal*, vol. 99, pp. 1–10, 2014.

[4] S. Zhong, Y. R. Yang, and J. Chen, "Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1987–1997, San Francisco, Calif, USA, April 2003.

[5] J. Mundinger and J.-Y. Le Boudec, "Analysis of a reputation system for Mobile Ad-Hoc Networks with liars," *Performance Evaluation*, vol. 65, no. 3-4, pp. 212–226, 2008.

[6] D. Ghosh, D. A. Pados, R. Acharya, and J. Llinas, "On dempster-shafer and bayesian detectors," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 36, no. 5, pp. 688–693, 2006.

[7] C. Romer and A. Kandel, "Applicability analysis of fuzzy inference by means of generalized Dempster-Shafer theory," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 4, pp. 448–453, 1995.

[8] Z. Li and H. Shen, "A hierarchical account-aided Reputation Management system for large-scale MANETs," in *Proceedings of the International Conference on Computer Communications (INFOCOM '11)*, pp. 909–917, Shanghai, China, April 2011.

[9] T. Anantvalee and J. Wu, "Reputation-based system for encouraging the cooperation of nodes in mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 3383–3388, IEEE, Glasgow, UK, June 2007.

[10] F. Oliviero and S. P. Romano, "A reputation-based metric for secure routing in wireless mesh networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 1–5, New Orleans, La, USA, December 2008.

[11] J.-P. Hubaux, T. Gross, J.-Y. Le Boudec, and M. Vetterli, "Toward self-organized mobile ad hoc networks: the terminodes project," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 118–124, 2001.

[12] Z. Ji, W. Yu, and K. J. R. Liu, "A game theoretical framework for dynamic pricing-based routing in self-organized MANETs," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 7, pp. 1204–1217, 2008.

[13] A. Mukherjee and H. M. Kwon, "General auction-theoretic strategies for distributed partner selection in cooperative wireless networks," *IEEE Transactions on Communications*, vol. 58, no. 10, pp. 2903–2915, 2010.

[14] 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 (MOBICOM '00)*, pp. 255–265, Boston, Mass, USA, August 2000.

[15] L. Wei, Z. Cao, and H. Zhu, "MobiGame: a user-centric reputation based incentive protocol for delay/disruption tolerant networks," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference (GLOBECOM '11)*, pp. 1–5, Houston, Tex, USA, December 2011.

[16] L. Yaqiong, X. Weilian, L. Keqiu, C. Zhongxian, M. Geyong, and Q. Wenyu, "DHTurst: a robust and distributed reputation system for trusted peer-to-peer networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '10)*, pp. 1–6, Miami, Fla, USA, 2010.

[17] R. R. Yager, "Cumulative distribution functions from Dempster-Shafer belief structures," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 34, no. 5, pp. 2080–2087, 2004.

[18] C. K. Murphy, "Combining belief functions when evidence conflicts," *Decision Support Systems*, vol. 29, no. 1, pp. 1–9, 2000.

[19] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehavior in MANETs," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 536–550, 2007.

[20] Y. Xi and E. M. Yeh, "Pricing, competition, and routing for selfish and strategic nodes in multi-hop relay networks," in *Proceedings of the IEEE 27th Conference on Computer Communications (INFOCOM '08)*, pp. 1463–1471, Phoenix, Ariz, USA, April 2008.

[21] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo, "The logarithmic nature of QoE and the role of the Weber-Fechner law in QoE assessment," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–5, Cape Town, South Africa, May 2010.