A Generous Cooperative Routing Protocol for Vehicle-to-Vehicle Networks

Xiaohui Li and Junfeng Wang*
College of Computer Science, Sichuan University
Chengdu 610065, P.R. China
[e-mail: 2013323040006@stu.scu.edu.cn, wangjf@scu.edu.cn]
*Corresponding author: Junfeng Wang

Received February 11, 2016; revised June 4, 2016; accepted August 2, 2016; published November 30, 2016

Abstract

In vehicle-to-vehicle (V2V) networks, where selfishness degrades node activity, countermeasures for collaboration enforcement must be provided to enable application of a sage and efficient network environment. Because vehicular networks feature both high mobility and various topologies, selfish behavior judgment and establishment of a stable routing protocol become intensely challenging. In this paper, a two-phase-based generous cooperative routing protocol (called GEC) is presented for V2V networks to provide resistance to selfishness. To detect selfish behaving vehicles, a packet forwarding watchdog and an average connection rate based on the multipath weight method are used, where evidence is gathered from different watchdogs. Then, multihop relay decisions are made using a generous cooperative algorithm based on game theory. Finally, through buffering of the multiple end-to-end paths and judicious choice of optimal cooperative routes, route maintenance phase is capable of dealing with congestion and rapidly exchanging traffic. Specifically, it is proved that the GEC is theoretically subgame perfect. Simulation results show that for V2V networks with inherently selfish nodes, the proposed method isolates uncooperative vehicles and is capable of accommodating both the mobility and congestion circumstances by facilitating information dissemination and reducing end-to-end delay.

Keywords: Vehicle-to-vehicle networks, cooperative routing, selfish behavior, game theory, route performance

This work was supported by the National Key Research and Development Program (2016YFB0800600), the National Natural Science Foundation of China (91338107, 91438120, 91438120), the Ph.D. Program Foundation of Ministry of Education of China (20130181110095) and the Provincial Key Science and Technology Research and Development Program of Sichuan, China (2016ZR0087).

http://dx.doi.org/10.3837/tiis.2016.11.007 **ISSN : 1976-7277**

1. Introduction

With the increasing numbers of vehicles on the road and the rapid development of intelligent transportation systems, inter-vehicle communications in V2V networks are gaining increasing attention from both academic researchers and automotive industries [1][2]. The V2V networks represents a new kind of ad hoc networks that is characterized by a highly mobile topology, security requirements and the fact that most applications are heavily reliant on broadcast transmission. V2V networks also have the problem of selfish nodes which can hinder the implementation of any protocol dedicated to it [3]. Dealing with these nodes in V2V networks is more challenging than in mobile ad hoc networks because of the increased detection ambiguity caused by the high mobility of the vehicles. It is therefore highly desirable to provide cooperative inter-vehicle communication methods for V2V networks.

According to [4], misbehaving vehicles in vehicular networks can be summarized using the following categories: *free riders* (who defer from sharing information), *selfish liars* (who deliberately falsify disseminated information) and *bona fide mules* (who overwrite the fresh information carried by the mobile storage nodes and convert them into forged information). Among these categories, vehicles that display selfish behavior, *i.e.*, *free riders*, impede information dissemination and reduce the connectedness of V2V networks for a variety of reasons, including low resources (*e.g.*, network or bandwidth resources), fear of receiving malicious data from unknown users, privacy issues, or even a simple lack of interest in helping nodes from other communities [5]. As a result, a selfish node considers only its own characteristics and states without taking the effects of its selfishness on other nodes or on the overall network into account; such a node may even try to hinder itself by not participating in data forwarding for its own purposes.

To address this problem in vehicular networks, a cooperative routing protocol [6] based on a quality-of-service (QoS) scheme is proposed to motivate the nodes to behave cooperatively during cluster formation gathering and to detect misbehaving nodes after the clusters are formed. However, the protocol assumes that all participants will follow the rules of QoS provisioning of the underlying applications (*i.e.*, the bandwidth and the residual distance). In wireless networks, many studies have been conducted on cooperative routing for selfish node detection. These approaches can be briefly summarized as reputation-based schemes and credit-based schemes. Because of the limitations of credit-based schemes [7–9], including lack of scalability, centralization, and the need for tamperproof hardware, which means that they are not suitable for use in V2V networks, we concentrate on reputation-based schemes. In a reputation-based scheme [10–14], the nodes communicate with each other to provide feedback in terms of a reputation value. In this way, every node gathers a high reputation value to build trust and confidence about good behavior and cooperation within the network. These approaches have several efficiency limitations that must be solved, and these limitations are described in detail below.

In [10], the tit-for-tat (TFT) method associates incentive mechanisms with the reputation concept, so that cooperation with more reputable nodes offers incentives nodes to enhance their own reputations. Therefore, cooperation can provide the benefit of access to a larger range of services. However, this strategy has the problem that it neglects the cases of high mobility and collisions that may hinder monitoring processes in V2V networks. Marti et al. [11] included the watchdog and pathrater concepts in the dynamic source routing (DSR) [15] protocol. The proposed method is based on preventing detected misbehaving nodes from

forwarding packets. However, when using this scheme, misbehaving nodes are remunerated regardless of their behavior. CORE [12] is a collaborative reputation mechanism that also uses the watchdog concept and defines three types of mechanisms that are based on task specific behavior, observations, and positive reports by other nodes. To judge a specific node, a weight is assigned to each type of reputation to build an aggregated reputation. The weakness of CORE is that the weight associated with each functional reputation must be set accurately. CONFIDANT [13] sends an alarm to the network nodes upon detection of a misbehaving node. This protocol aims to isolate misbehaving nodes from the network. However, the credibility of the received alarms is not guaranteed. OCEAN [14] presents an investigation of the factors required for cooperation based on a game theory model in ad hoc networks in the absence of any incentive mechanisms. However, OCEAN is vulnerable to attack via tampering with the avoid-list and also does not take misjudgments into account. In summary, these reputation-based mechanisms have disadvantages that limit their efficiency for implementation in V2V networks in terms of ambiguous collisions, false alarms, and non-cooperative monitoring.

To solve the above problems, we propose a two-phase-based generous routing protocol (called GEC) that detects misbehaving vehicles using monitored information. First, to improve the probability of misbehavior detection and reduce the number of false alarms, a cooperative watchdog model is used and the evidence is correlated cooperatively; in this way, GEC overcomes the problem of the ambiguity in detection that results from packet collisions, the high mobility of the vehicles, and untrustworthy watchdogs. Incentives are then given in the form of reputations in the route discovery phase, where network services are offered based on each vehicle's accumulative reputation. In addition, optional multi-paths are buffered for stability in routing selection, which means that GEC is capable of handling congestion and rapidly exchanging traffic. Finally, our tests are based on realistic traffic data for various metrics. Our end-to-end cooperative routing protocol effectively optimizes network performance by reducing delay by 25% and improving goodput by 31.2%.

The remainder of this paper is organized as follows. Section 2 describes related work in cooperative vehicular communications, before Section 3 details the design of the proposed cooperative routing protocol that can solve the problems described in Section 1. Section 4 discusses the parameters used in our simulation scenario, and the main results of the simulations. Section 5 presents our conclusions.

2. Related Work

In recent years, considerable efforts have been made to merge or integrate the cooperative vehicular communication models from the automotive and research communities. Most of the work that has already been proposed to deal with cooperation in vehicular communications follows a mobile ad hoc network approach, which can be classified into two main categories: the credit-based and reputation-based approach, as well as the vehicular delay-tolerant networks (VDTNs) approach. The following section gives an overview of these approaches and discusses several state-of-art cooperative methodologies designed for vehicular networks.

In general, the basic idea of credit-based schemes [7]–[9] is that the nodes pay virtual money to be served and are paid to serve. For example, nodes must pay to obtain or use network resources, and they are paid to provide or share these resources with other network nodes. However, the lack of scalability, the centralization, and the requirements for tamperproof hardware are limitations that can be encountered when using these schemes. In reputation-based schemes [10–14], the nodes communicate with each other to provide

feedback in terms of a reputation value with regard to a specific node's cooperative behavior. In this way, each node collects a high reputation value to build trust and confidence about its good behavior and cooperation in the network. Subsequently, misbehavior announcements are broadcast across the network, which leads to misbehaving nodes being discarded from use in all future routes. reputation-based schemes are more suitable for cooperation in V2V networks.

Wahab *et al*. [6] proposed a cooperative routing protocol based on QoS-optimized link state routing (QoS-OLSR) combined with election of optimal cluster-heads. The protocol added both velocity and residual distance parameters as the QoS function, rather than the residual energy used in a mobile ad hoc network (MANET). However, this scheme flawed in that nodes are expected to reveal their available bandwidth and vehicle mobility information to participate in the election. The likelihood of revealing such information is dependent on the individual node's behavior. We therefore use packet forwarding information as an efficient judgment measure of the node behavior and record the historical behavior of the nodes. In [16], a repeated game theory TFT was used to model the packet forwarding strategies of vehicular nodes based on the QoS-OLSR protocol. After the election of multi-point distribution relays (MPRs), the cluster-head aggregates the collected observations using dempster–shafer (DS) theory and then spreads the results. However, recently proposed approaches based on DS theory [6] [16] suffer from instability when the observations come from dependent sources [17] and the DS method does not benefit from previous detection experience, which is considered to be a waste of huge amounts of data. Hence, in our proposal, historical behavior is an important metric that is used for cooperative routing, which needs to be recorded.

Despite VDTNs being a recently developed vehicular architecture, cooperation within this architecture has already been addressed. Most researchers have tried to understand the effects of the different cooperative strategies on the overall performance of this type of architecture. Soares et al. [18] studied the cooperation problem in VDTNs when considering urban scenarios where the node density is sparse and network partitioning may occur. VDTNs use both control and data plane separation. The former can improve the overall network performance, while the latter offers considerable gains. A reputation system [19] for VDTNs was proposed. This system includes four different ways to reward or punish the nodes, and the individual node performance was evaluated by considering three different protocols (First Contact, Spray and Wait, and GeoSpray). It can be concluded that improvement of the reputation system contributes to an increased bundle delivery probability. In VDTNs, to seek network performance improvements by increasing the bundle delivery probability and helping the nodes to save resources, optimal monitoring and management strategies may be developed to help the nodes to save network resources.

3. Generous Cooperative Routing Protocol (GEC)

The GEC routing protocol includes several components that are used to construct cooperative paths and select and distribute network traffic over the discovered paths. In the following, we describe these components in detail by first presenting the overall architecture of the proposed routing protocol. Then, we describe how the learning relay metrics and the cooperative relay selection algorithm work in detail. Finally, the route maintenance phase is presented.

3.1 GEC Architecture

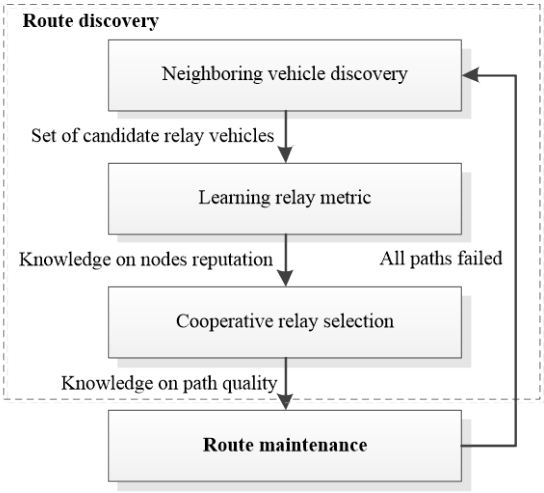


Fig. 1. Components of the CEC protocol.

Fig. 1 shows the architectural components of our distributed cooperative routing protocol, which can be viewed as a multistage decision-making problem. As shown in the figure, the GEC architecture has two main phases: *route discovery* and *route maintenance*. The *route discovery* phase consists of three components, which are *neighboring vehicle discovery*, *learning relay metric* and *cooperative relay selection*. Based on these two phases, the proposed protocol can identify suitable sets of relay vehicles for the source and for all intermediate vehicles on the paths, create an end-to-end path from the source to the destination while avoiding selfish-critical nodes, and select an optimal path. In the following subsections, we describe the design issues in detail.

3.2. Neighboring Vehicle Discovery and Learning Relay Metric

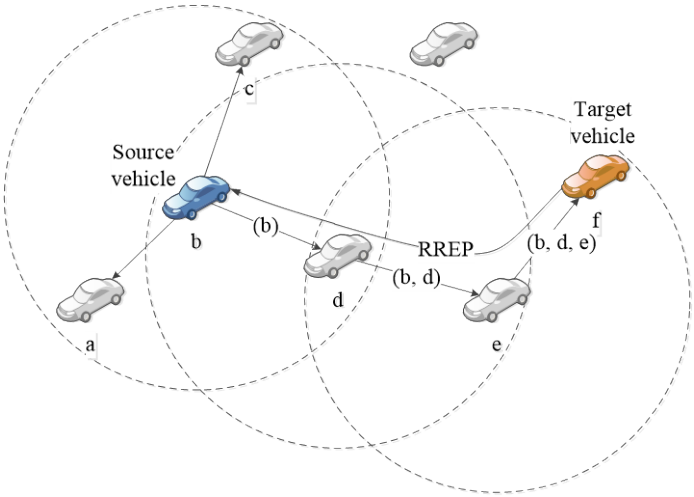


Fig. 2. Routing paths of the GEC protocol in V2V networks.

We first present a brief description of the DSR [15]-based ad hoc routing algorithm GEC, which creates an end-to-end path from the source to the destination vehicle. The design principle of GEC is based on the DSR routing protocol but with some modifications. As in

DSR, when a route is required, a GEC source vehicle broadcasts a route request (RREQ) packet, as shown in **Fig. 2**. However, unlike the situation in DSR, a GEC source node tags the RREQ packet with a packet forwarding ratio. We replace the DSR RREQ '*reserved*' fields with the packet forwarding ratio. A neighbor node then only rebroadcasts this RREQ packet upon reception if a certain reputation value is reached; this situation will be described in detail in Section 3.3. Additionally, the neighbor node deducts the sojourn period of the RREQ packet from the start timestamp before it rebroadcasts the packet. Therefore, similar to DSR, multiple copies of the RREQ packet will eventually reach the destination sink. The sink then returns all route reply (RREP) packet towards the source node of the RREQ packet with the packet forwarding ratio value. Each node maintains a route table that is composed of multiple routing entries and stores the routes to the various nodes in the V2V network. Thus, a reliable end-to-end path consisting of cooperative nodes is established.

As shown in **Fig. 2,** in the route discovery phase, source vehicle *b* is regarded as being at the central node of a set of neighbors, which is defined as $i \in S$, and all of the neighbor hops (*i.e.*, vehicle *a*, vehicle *c*, and vehicle *d*) of the broadcast RREQ message are regarded as neighbors $\{n_1, n_2, \ldots, n_N\} \in N$. The central vehicles and their neighbor vehicles established the set $\Re = S \cup N$. The dashed circles represent the transmission ranges of the central vehicles. We use a map in each vehicle to record the behavior of its neighbors; the required procedure for this process is explained as follows in **Algorithm 1**.

Algorithm 1. Neighbor Record algorithm

Initialization:
Let τ_t be the current time of the vehicle at time slot *t*.
Let *n* be a neighbor vehicle of the central node *i*.
Let $\Re$ be the set of the central node and neighbor nodes.
Let *IsNeighbor(x,y)* checks that if vehicle *x* is vehicle *y*'s neighbor.
Let *mapNeighbor* be the map of current neighbor vehicles.
Let τ_{last} be the end of time at time slot *t-1*.
Let λ be the threshold of the time window
Procedure NeighborRecord
  for each vehicle $n \in \Re$ **do**
    if $n \neq i$ and *IsNeighbor(n,i)*==true **then**
      Update *mapNeighbor*;
    end if
  end for
  Record the IP address of the sending vehicles into *mapNeighbor*;
  for each packet receive time τ_t **do**
    if $\tau_t - \tau_{last} \geq \lambda$ **then**
      Update $\tau_{last} = \tau_t$;
      Update *mapNeighbor*;
    end if
  end for
end procedure

Next, we present the learning relay metric algorithms for the packet forwarding ratio and the vehicle connection rate.

- **Packet Forwarding Ratio:** The packet forwarding ratio is the proportion of all packets that have actually been forwarded correctly. Correct forwarding means that the forwarding node not only transmits the packet correctly to its next hop node but also that it forwards the packet voluntarily. For instance, selfish nodes do not participate correctly in the routing function because they drop all the data packets that come to them for forwarding, by simply not advertising the available routes, or by not forwarding the RREQ packets. Consequently, these selfish nodes will not appear on the packet forwarding path. Senders then monitor this illegal behavior and the packet forwarding ratio of the neighbor will decrease accordingly.

  To perform this procedure, the nodes work in a promiscuous mode using a unique identifier that cannot be forged to detect misbehavior among their one-hop neighbors. By checking the IP header of each overheard packet, the monitor can then identify the packet uniquely. All the vehicles must keep a watchdog $\nu_{i\to n}^{t}$ for the number of messages that the source vehicle i let the neighbor vehicle n forward in the time slot t. Then, another watchdog, $\kappa_{i\to n}^{t}$, is used to define the number of packets that all of the last hop vehicles actually forward in time slot t. A selfish vehicle will not relay packets for the other nodes and the next hop will not update $\kappa_{i\to n}^{t}$ in this situation. Let $Q_{i\to n}^{t}$ be the packet forwarding ratio in the t-th time window. $Q_{i\to n}^{t}$ is updated according to **Algorithm 2**. When a packet is overheard in a promiscuous mode, the monitor will check whether or not the overheard packet matches the packet forwarding information.

Algorithm 2. Packet forwarding ratio algorithm

Initialization:
Let *i* be the source vehicle.
Let *n* be a neighbor vehicle of the source node *i*.
Let *mapNeighbor* be the neighbor map of current vehicle.
Procedure PacketForwardingRatio
Sender:
if *i* sends a packet to *n* **then**
  $\nu_{i\to n}^{t} = \nu_{i\to n}^{t} + 1;$
  $Q_{i\to n}^{t} = \kappa_{i\to n}^{t} / \nu_{i\to n}^{t}, \quad \tau_t - \tau_{last} \geq \hat{\lambda};$
end if
Receiver:
if *n* receive a packet from *i* and overheard it matches the fowarding packet information **then**
  for each $n \in mapNeighbor$ **do**
    $\kappa_{i\to n}^{t} = \kappa_{i\to n}^{t} + 1;$
    $Q_{i\to n}^{t} = \kappa_{i\to n}^{t} / \nu_{i\to n}^{t}, \quad \tau_t - \tau_{last} \geq \hat{\lambda};$
  end for
end if
end procedure

- **Vehicle connectivity rate:** Because the vehicle connectivity is the most important constraint in a V2V network, each vehicle must make decisions based on that connectivity. Each vehicle's connection rate is weighted using the perceived connectivity ratio that node *i* measured from node *n*. Using the multipath weighting method, the connection rate can avoid attacks that spread false values with the aim of improving a selfish node's reputation. Based on the average multipath weighting method, all neighbors can identify selfish vehicles with low connectivity, and thus the false values only have a small impact on the average. The vehicle connectivity rate C_n^t is explained as follows in **Algorithm 3**.

Algorithm 3. Vehicle connectivity rate algorithm

1: **Initialization:**
2: Let *i* be the source vehicle.
3: Let *n* be the next-hop vehicles.
4: **Procedure** VehicleConnectivityRate
5: Vehicle *i* gathers $Q_{i\to n}^t$ of $k \in mapNeighbor$.
6: **for each** $k \in \Re$ and $k \neq n$ **do**
7: $\tau_n^t = \tau_n^t + Q_{i\to k}^t \cdot Q_{k\to n}^t$
8: $\pi_n^t = Q_{i\to k}^t$
9: **end for**
10: $C_n^t = \tau_n^t / \pi_n^t$
11: **end procedure**

Apparently, we can define the disconnection rate as that at the time window *t* in Equation (1).

$$N_i^t = 1 - C_i^t \tag{1}$$

Because the nodes in V2V networks move around dynamically and there are many interference conditions (such as collisions) as a result, the packet loss rate cannot actually be detected. The parameter e stands for an erroneous judgment of the GEC.

3.3. Cooperative Rlay Slection

For each hop on the cooperative routing path, determination of the set of candidate relay vehicles between two adjacent routers and selection of the optimal relay vehicle from the set is challenging. Our mechanism provides a distributed reputation evaluation scheme that is implemented autonomously at every vehicle with the objective of identification and isolation of selfish neighbors. As part of the neighbor monitoring scheme, each node maintains a reputation table, in which a reputation index is stored for each of that node's immediate neighbors. For example, to prevent selfish behavior and thus provide motivation for nodes to build up their reputations, each node determines whether to forward or drop a packet based on the reputation of the packet's historical behavior.

We now introduce the principle concept from game theory [20] that is adopted in this paper. For simplicity, a strategic cooperative game $\Upsilon(P,S,U)$ includes the following fundamental components:

- A player set P: $\rho \in P$,where $P=\{\rho_1,\rho_2,\ldots,\rho_N\}$ is the set of participants and N is the number of players in the game.
- The players' strategy space is $S=\{\delta_1,\delta_2,\ldots,\delta_N\}$.
- The payoff function is U: for each player $\rho \in P$, the payoff function $U_\rho : S \to R$. $U=(\mu_1,\mu_2,\ldots,\mu_N)$ denotes the vector of the results in the game for each player.

Definition 1. In a standard game expression, where there are N players. The players' strategy space is S. Payoff function is U. Thereafter, a strategic cooperative game uses $\Upsilon(P,S,U)=\{\delta_1,\delta_2,\ldots,\delta_N;\mu_1,\mu_2,\ldots,\mu_N\}$ to define the game.

Definition 2. In a standard game $\Upsilon(P,S,U)$, there is an alternative strategy where player ρ can be assumed such that $\{\delta_\rho^*,\delta_\rho^{'}\} \in \delta_\rho$. For each possible combination of strategies with other participants, the benefits of ρ choosing δ_ρ^* outweighs the benefits of selecting any other $\delta_\rho^{'}$ options. Then, by comparing strategy $\delta_\rho^{'}$ with strategy δ_ρ^*, the strategy profile δ_ρ^* constitutes a Nash equilibrium. For each player ρ, if $\mu_\rho(\delta_1,\delta_2,\ldots,\delta_\rho^*,\delta_{\rho+1},\ldots,\delta_n) > \mu_\rho(\delta_1,\delta_2,\ldots,\delta_\rho^{'},\delta_\rho,\ldots,\delta_n)$, then accordingly, δ_ρ^* is a strictly dominated strategy.

Definition 3. If the combination of strategies δ_ρ is a subgame of $\Upsilon(P,S,U)$ and is a perfect Nash equilibrium of the original game $\Upsilon(P,S,U)$, it is a finite extension of the Υ-type game subgame perfect equilibrium.

We now model the network described in the experiment above as a strategic game, called a cooperative-defect game. In this game, each source node is a player, and the set of paths from each source to each destination is a set of strategies. Consider two players, player 1 and player 2, who need to decide whether to forward packets or not. The matrix that represents this game is shown in **Table 1**.

Table 1. Payoff matrix of cooperative-defect game.

		Player 2	
		Cooperate	**Defect**
Player 1	**Cooperate**	$\gamma-\sigma$, $\gamma-\sigma$	$-\gamma-\sigma$, γ
	Defect	γ, $-\gamma-\sigma$	$-\gamma$, $-\gamma$

Let x be any entry in **Table 1**, where we use $y=\dfrac{x+\gamma}{2\gamma-\sigma}$ to normalize the table. As a result, the utility of cooperation is normalized to 1 and the benefit of bilateral defection is normalized to 0. Therefore, the average payoff at time slot t is computed as shown in Equation (2)

$$\mu_i^t = c_i^t \cdot c_{\hat{i}}^t + \frac{2\gamma}{2\gamma-\sigma} p_i^t c_{\hat{i}}^t - \frac{\sigma}{2\gamma-\sigma} c_i^t p_{\hat{i}}^t \tag{2}$$

where c_i^t represent the connectivity rates and p_i^t represent the disconnection rates. We define

$$c_i^t = 1 - p_i^t \tag{3}$$

Rearranging terms, we derive:

$$\mu_i^t = 1 + \frac{\sigma}{2\gamma-\sigma} p_i^t - \frac{2\gamma}{2\gamma-\sigma} p_{\hat{i}}^t \tag{4}$$

The discounted average payoff of player *i* when starting at time slot *t* is then given by

$$U_i^t = \sum_{t=n}^{\infty} \varepsilon^t \mu_i^t \tag{5}$$

where $\varepsilon \in (0,1)$ is the discount factor. Because node *i* cannot know ρ_i^t for sure, it also does not know its payoff either. However, we can use the actual payoff in the analysis because it tells us whether a given node can gain by deviating from a given strategy.

In this game, each player is allowed to use a strategy to decide whether or not to drop or forward the packets based on their history. We use $R_{i,Gec}^t$ to denote the dropping probability that player *i* should use at the time slot *t* based on strategy δ.

It is not always possible to determine whether a packet was relayed or not because of the detection ambiguity that results from packet collisions, the high mobility of the vehicles, and the untrustworthy nature of the watchdogs. One way to deal with this is to use a generosity factor ℓ that allows cooperation to be restored. This type of strategy is known as a generous tit-for-tat strategy [21]. The factor ℓ is difficult to define, and we use the last time slot value to define this value as follows:

$$R_i^t = \begin{cases} 0 & \text{, for } t = 0. \\ \max\left\{ \mathrm{N}_{\hat{i}}^{t-1} - \ell_{\hat{i}}^{t-1}, 0 \right\} & \text{, for } t \geq 1. \end{cases} \tag{6}$$

where we define i as the source vehicle and $\hat{i}$ as its neighbor vehicle. Also,

$$\ell_i^t = \alpha \left(\mathrm{N}_i^t - R_{i,Gec}^t \right) \tag{7}$$

Additionally, we then normalize the function

$$R_{i,Gec}^t = \min\left\{ R_{i,Gec}^t, 1 \right\} \tag{8}$$

Thus, if $N_i^t > R_{i,Gec}^t$, it means than node *i* has been detected to be dropping more packets than it should in the GEC method. The parameter ℓ_i^t measures this deviation. The parameter α is a scaling value of the generosity based on various phenomena, such as collisions, rather than on selfishness. In addition, a feasible threshold for the time window can also be determined based on the vehicles' speed.

Theorem 1. A GEC game is subgame perfect if and only if

$$\ell_i^t \le e_i^t \text{ and } \varepsilon > \frac{1}{2\gamma\left(1-e_i^t\right)} \tag{9}$$

It should be noted here that GEC is a single-stage history strategy because it only needs to take what happened in the previous stage into account. If both nodes use the GEC protocol, then cooperation is achieved on the equilibrium path if and only if $\ell_i^t = e_i^t$ [22].

Finally, we summarize how the reputation model can be integrated into the overall routing process and how it is updated. Before starting a new route discovery process, the routing table or cache should first be examined to check whether it already has a workable route to the destination or not. If, unfortunately, no appropriate route information is available, then a route discovery phase is triggered by flooding of the RREQ message. In route discovery, as shown in Fig.2, if the reputation value of vehicle *d* when evaluated by vehicle *e* is higher than a specific threshold Γ, vehicle *e* would then forward the RREQ as vehicle *d* requests. Otherwise, vehicle *e* would reject this RREQ because of vehicle *d*'s low reputation. Upon receiving the RREP, vehicle *e* would encapsulate its reputation values of other peers on the same route and return the updated RREP to its upstream peer. When the source vehicle receives the RREP, it choose the optimized QoS route (*i.e.*, the hop count) to forward packets.

3.4. Route Maintenance

In the route maintenance phase, path reconstruction should be performed to reduce the performance degradation. A path rebuilding process can be initiated in a situation where an active path has failed. Because the topologies of the V2V networks change dynamically, *i.e.*, mobile nodes may join or quit the network for a variety of reasons, initiation of a route rediscovery process frequently imposes high overheads. Therefore, the GEC will not trigger its route discovery phase until a certain number of active paths have failed.

Each vehicle maintains and updates a routing table when it receives an RREP or a route error packet. When a link failure is detected (via link layer feedback, for example), that route is then erased from the route table. Then, if there is another path available at the source node, route maintenance takes place. The GEC will choose the next maximum reputation value with optimized QoS parameters (*i.e.*, the hop count), and $\left(1-R_{i,Gec}^t\right)$ should be larger than the Γ threshold, which is set at 0.8. At the instant in time when the available paths are broken, the source node immediately initiates new route discovery without any examination.

The unused routes in the routing table expire using a timer-based technique. When the time at the source node $\hat{\lambda}$ expires, the source node then validates the end-to-end reliability by sending a route check message [23] to each of the existing paths. The objective of providing a tolerance parameter is to balance the number of control messages that result from route

discoveries during data transmission with the degree of end-to-end reliability that satisfies the reliability requirement.

4. Simulation Results

In this section, we explain the simulation scenario and the parameters that we used to build our simulations in detail.

4.1. Mobility Traces and Simulation Parameters

Fig. 3. Simulation topology.

By approximating the vehicle flows of real-world samples, the optimal model parameter values can be determined so that the networks that are generated using the traffic models have the closest graph attributes to the real road network samples. The road topology that we imported is the realistic case of road samples from Beijing, China, as shown in **Fig. 3**. Multiple monitoring systems that can record every vehicle's information are located on every corner of the main street. **Fig. 4** shows the traffic data format for the vehicles that we used here. The maximum speed allowed on this highway is 90 km/h, while the average speed is 30 km/h.

Location_Out	Direction_Out	Location_in	Direction_In	Travel_Time	Speed	Matched_count

Fig. 4. Traffic data format.

We evaluate the GEC performance using simulation experiments that were conducted on the network simulator *ns-2* [24]. We compared the results with those of cooperative routing, *i.e.*, with QoS-OLSR [6], OCEAN [14], the DSR with TFT [10] strategy, and QoS-OLSR with a Dempster-Shafer and TFT (DS-TFT) strategy [16]. By considering the realistic case of the road samples shown above, the parameter settings of the simulated networks are as shown in **Table 2**, where 171 vehicles are simulated in a square area of 3000×3000 m^2. Each vehicle has a physical radio range of 250 m and a raw bandwidth of 2 Mb/s, while the IEEE 802.11 standard is used at the media access control (MAC) layer. There are from 16 up to 40

connections from the different source vehicles to the different target vehicles, and each source transmits at a constant bit rate (CBR) of 2 packets/s with the packet size of 512 bytes. The total simulation time is 200 s. The number of selfish nodes that is used to simulate the protocols is 5% of the total number of nodes by default. At this level, the impact of the selfish nodes on the network will be catastrophic. The time window λ is set to 30s based on average speed.

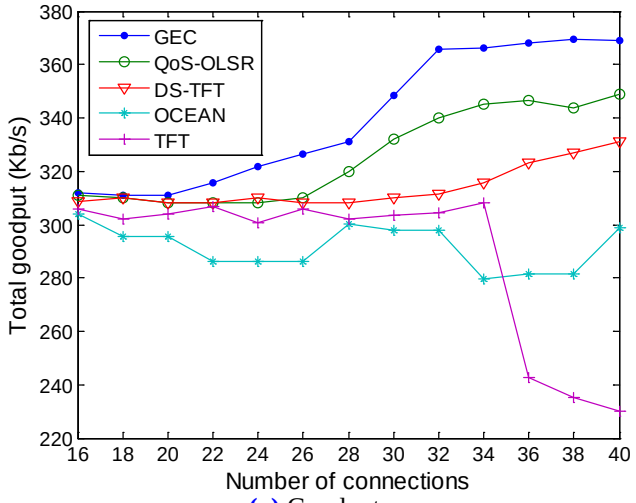
Table 2. Simulation settings.

Parameter	Value
Geolocation	Beijing, China
Platoon Size	171 vehicles
Network architecture	Homogeneous and flat
Radio Model	two-ray ground reflection model
MAC Layer	IEEE 802.11standard
Physical Radio Range	250 m
Section of Road	3000*3000 m^2
Ttraffic Stream	CBR(2 packets/s)
Packet Size	512 bytes
Initial node energy	300 J

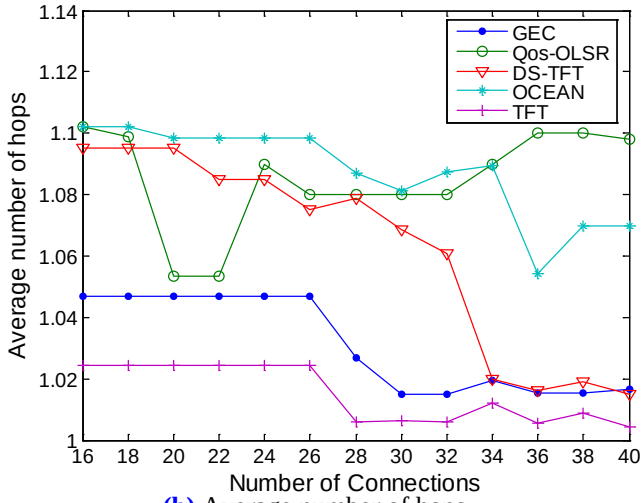
4.2. Simulation Results

In this section, we evaluate the performances of the protocols under study for various numbers of connections from the source vehicles to the target vehicles. For all sets of connections, we looked at the performance metrics, which are very important in V2V networks, and at how they are affected by the different network connections. We define the following metrics for performance comparison:

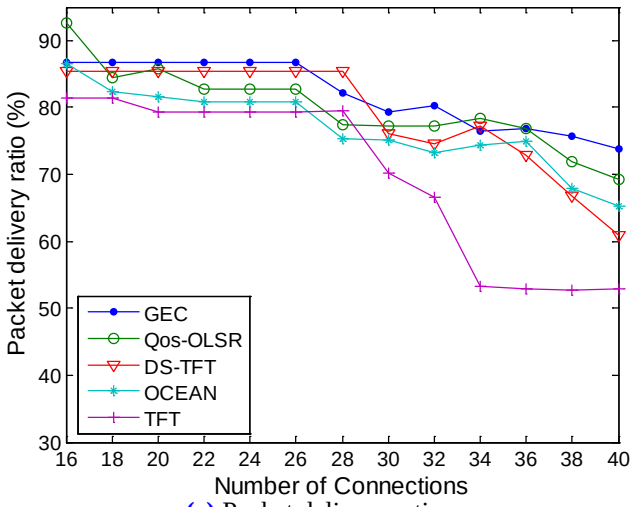
- *Goodput* is the number of application level information bits delivered by the network to a destination per unit time.
- *Average number of hops* is the average end-to-end hop count required to transfer data from the source to the destination during data forwarding.
- *Packet delivery ratio* is the ratio of the total number of packets received by the sink before the delay deadline to the total number of packets generated by all source nodes in the network.
- *Packet average delay* of a single packet is determined by taking the end-to-end delivery delays experienced by the individual data packets and then averaging them over the total number of packets received by the sink.
- *Average energy consumption* is the average difference between the initial energy level and the final energy level that is left in each node.
- *Average routing overhead* can be measured as the ratio of the number of control messages that are transmitted to the number of data packets that are delivered to the sink before the network lifetime expires.



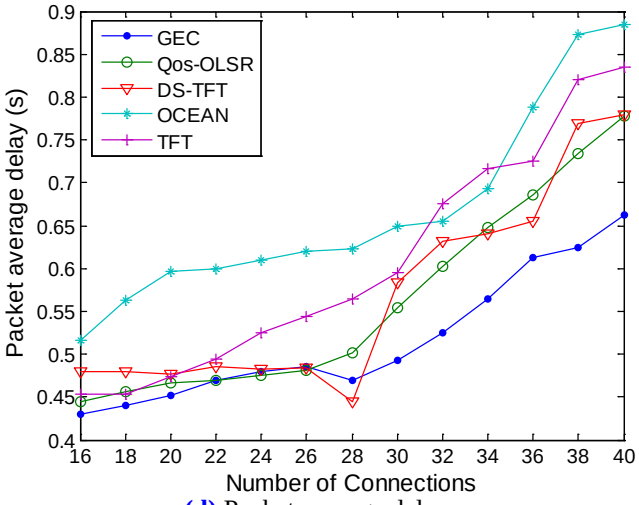
(a) Goodput.



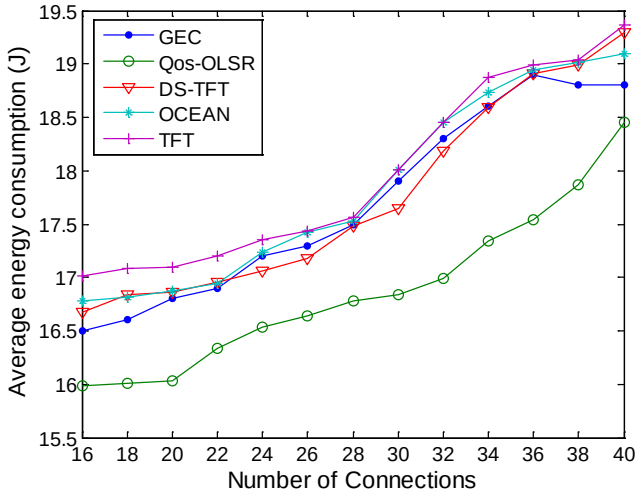
(b) Average number of hops.



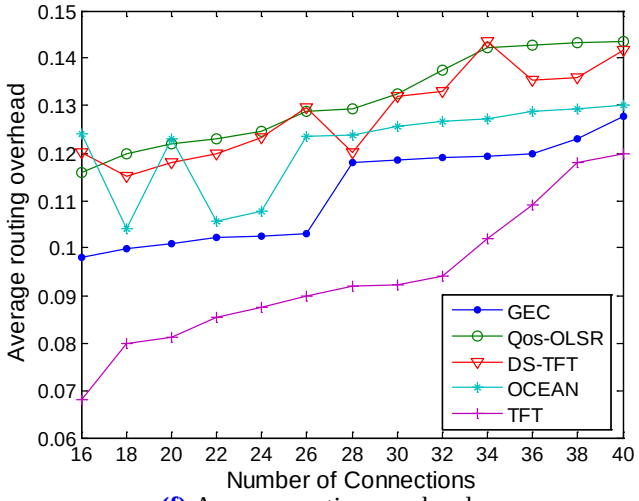
(c) Packet delivery ratio.



(d) Packet average delay.



(e) Average energy consumption.



(f) Average routing overhead.

Fig. 5. Performance comparisons of the different protocols for various numbers of connections.

The simulation results, as shown in **Fig. 5**, indicate that the performance rates of the different protocols vary greatly among the protocols as the number of connections increases in terms of all performance metrics, as was expected theoretically. For example, in **Fig. 5(a)**, for lower numbers of connections (≤ 20), the total goodput results for the proposed GEC and QoS-OLSR protocols are similar. However, the gap between the two protocols increases sharply as the number of connections increases. The figure shows an improvement in the performance of GEC over QoS-OLSR that is as high as 31.2%. The reason for this disparity is that when there are more connections available, the probability that a vehicle makes an erroneous judgment of the forwarding ratio increases, thus leading to increased numbers of misunderstandings where cooperative nodes are deemed to be acting selfishly. This increases the level of retaliation situations in the OCEAN protocol, the TFT strategy and the TFT strategy with DS theory. While QoS-OLSR performs better based on multi-metrics, it still has misjudgment limitations.

We also conclude from **Fig. 5(b)** that GEC can reduce the average number of hops required. The average number of hops for GEC decreases by up to 6.7% when compared to that for OCEAN. In addition, the average number of hops in GEC also decreases when the number of connections increases. This is because the path stability in GEC is improved by finding the highest reputation value with the shortest path. Additionally, we use a multipath mechanism, by which a relay node can buffer all available paths. This reduces the costs of new route discovery in the case of broken routes and eventually reduces the overall hop count. However, the DS-TFT and TFT strategies cannot benefit from the history information and suffer from misjudgment of cooperative nodes.

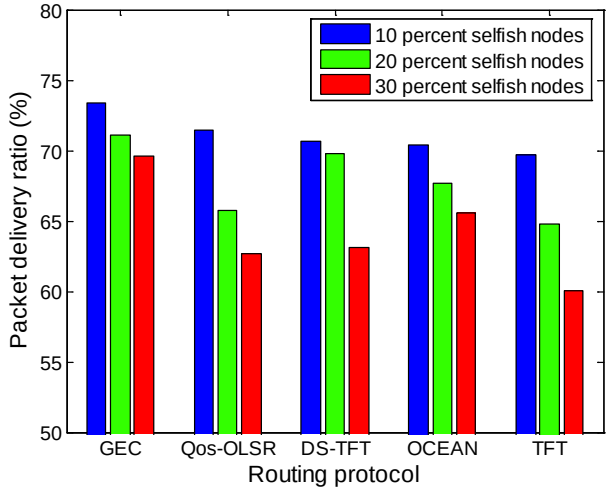
Fig. 5(c) shows that, in meeting a reliability requirement of 0.8, the proposed GEC protocol can tolerate a data traffic load of up to 32 connections, as compared to 28 connections for DS-TDT, 26 connections for QoS-OLSR and OCEAN, and only 18 connections for TFT. An in-depth look into the simulation trace file shows that when the number of the connections is more than 28, some paths become disabled because of vehicle motion. Overall, GEC outperforms the other protocols because of its judicious selection of transmission modes and relay nodes that optimize the network reliability.

Fig. 5(d) shows that GEC can also reduce the end-to-end delay as the number of connections increases. The figure shows that the GEC protocol achieves the shortest latency when the number of connections is higher than 28. The reason for this is that the lengths of some of the paths gradually become longer than they were when the original connections were established. Additionally, the message redundancy of epidemic-based protocols would be serious in this case. Based on generous game theory, GEC routing convergence occurs in a shorter period; in other words, the proposed routing method is simply more efficient.

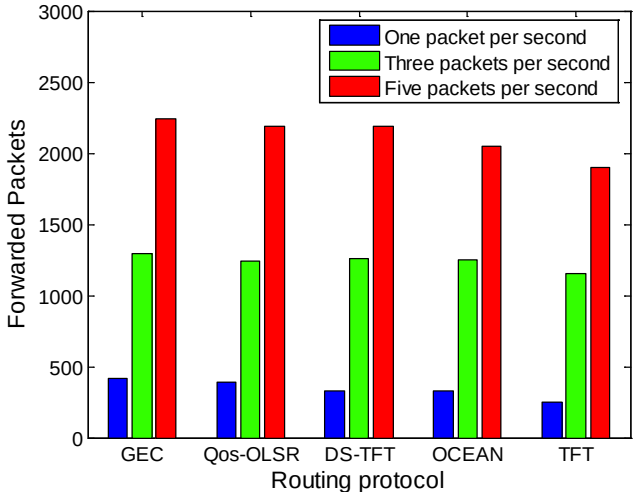
As shown in **Fig. 5(e)**, the average energy consumption of the GEC protocol is not the best when compared with the other protocols, with a 3.0% maximum improvement over the TFT strategy and 7.2% higher consumption than the QoS-OLSR system with increasing numbers of connections. When the number of connections is higher than 34, GEC performs a maximum of 2.58% better than the DS-TFT strategy. The rationale behind this result is that the energy parameter has minimal importance in V2V networks because of the long battery lifetimes of the vehicles. In the extension of the routing protocol, an energy constraint is not considered to be an important metric.

Fig. 5(f) shows the average routing overhead of the GEC protocol, which is neither the highest nor the lowest among the protocols. QoS-OLSR and DS-TFT are the highest of all the protocols. This is caused by periodic HELLO packet broadcasting and execution of shortest path algorithms to find the relays that can provide minimum power end-to-end paths. Unlike

QoS-OLSR, GEC relies only on single hop neighborhood information for relay selection and buffers all the end-to-end paths during route maintenance, which can reduce the overheads.



(a) Packet delivery ratio for various percentages of selfish nodes.



(b) Forwarded packets for various packet sending rates.

Fig. 6. Performance comparison of the different protocols.

Fig. 6(a) explores the effects of the packet delivery ratio for various fractions of selfish nodes. As expected, the packet delivery ratio of the protocols under study decreased when the number of selfish nodes increased. This is because when the number of selfish nodes increases, the total number of packets that are dropped increases proportionately. In this case, the packet delivery ratios of all the protocols are seen to be more than 70% for 10% selfish nodes. When the selfish node fraction increases to 30%, GEC maintains a 69% packet delivery ratio, while the ratio of the protocol using the TFT strategy deceases to 60%. **Fig. 6(b)** shows the relationship between the source rate and the number of forwarded packets. Because GEC estimates the erroneous judgment of the packet forwarding ratio and also compensates for misunderstandings between cooperative nodes, we see that the advantages of using GEC over the other strategies become more apparent when the network becomes heavily congested.

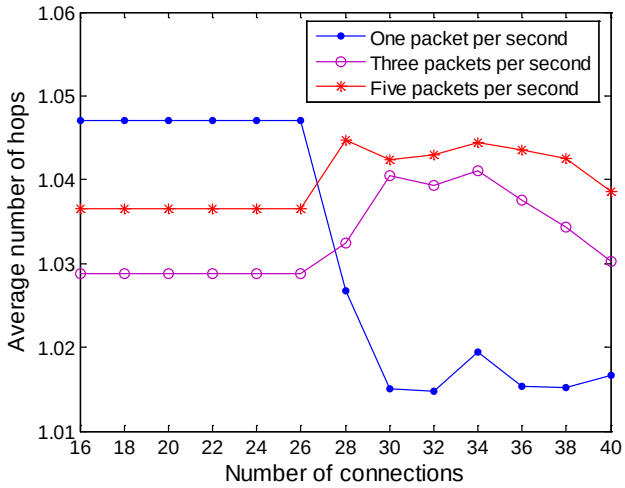


Fig. 7. Average number of hops for GEC protocol at different packet sending rate.

Fig. 7 plots the average number of hops for the GEC routing protocol for various packet sending rates. When packet sending rate is 1 packet/s, the average number of hops is observed to decrease as the number of connections grows. With increasing numbers of connections and higher packet sending rates, the average hop count increases. This is because the average delay and the buffer loads at the queues grow with the increasing data traffic load.

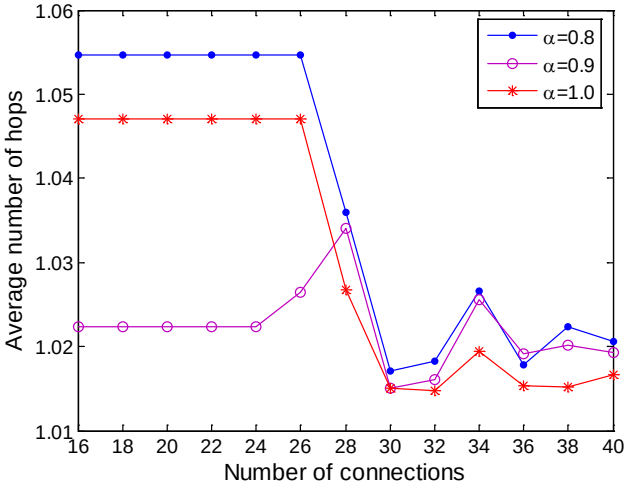


Fig. 8. Performance comparisons of GEC protocol with different value of α for varying numbers of connections.

We have also performed performance evaluations of the proposed GEC algorithm for various values of α, which is a scaling factor for the generosity of erroneous judgments. We see in **Fig. 8** that the introduction of α to our algorithm affected the performance of GEC in terms of the average number of hops for various numbers of connections. We also noticed that the optimal value of $\alpha = 0.9$ is better than larger (*e.g.*, $\alpha = 1.0$) or smaller values (*e.g.*, $\alpha = 0.8$). We see that by selection of an appropriate value for ℓ_i^t, it is possible to reduce the false alarm rate of the packet dropping probability. One possible approach for selection of α is to select a constant number, such as $\alpha = 1$, which we used in our simulations.

5. Conclusions

This work has addressed the problem of inherently misbehaving nodes in V2V networks through a proposed protocol called GEC. This two-phase-based protocol can motivate cooperation using a reputation-based approach. First, to overcome the challenge of detection of misbehaving vehicles, an average multipath weight method is used, where the evidence from the different watchdogs is gathered. Then, the decisions on relay node selection and judicious path choices are based on the reputation values offered during route discovery, allowing the other vehicles to efficiently isolate vehicles that show selfish behavior. In addition, in the route maintenance phase, the optimized QoS parameters (*i.e.*, the hop count) and the reputation value are used to reduce the route rediscovery time and the associated overhead. The simulation results show that the proposed protocol can increase the goodput by up to 31.2% and can reduce the average packet delay by 25% while maintaining both network stability and performance. In future work, we plan to introduce service differentiation into our proposed solution.

References

[1] S. Zeadally, R. Hunt, Y. S. Chen, A. Irwin and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217-241, December, 2012. Article (CrossRef Link).

[2] Kuk-Hyun Cho, Min-Woo Ryu, "A Survey of Greedy Routing Protocols for Vehicular Ad Hoc Networks," *Smart Computing Review*, Vol. 2, No. 2, pp.125-137, April 2012. Article (CrossRef Link).

[3] A. Jesudoss, S. K. Raja and A. Sulaiman, "Stimulating truth-telling and cooperation among nodes in VANETs through payment and punishment scheme," *Ad Hoc Networks*, vol. 24, pp. 250-263, January, 2015. Article (CrossRef Link).

[4] Evangelia Kokolaki, Georgios Kollias and Maria Papadaki, "Opportunistically-assisted parking search: a story of free riders, selfish liars and bona fide mules," in *Proc. of IEEE 10th Annual. Conf. on Wireless On-demand Network Systems and Services*, pp.17-24, March 18-20, 2013. Article (CrossRef Link).

[5] R. I. Ciobanu, C. Dobre, M. Dascălu, Ş. Trăuşan-Matu and V. Cristea, "Sense: A collaborative selfish node detection and incentive mechanism for opportunistic networks," *Journal of Network and Computer Applications*, vol. 41, pp. 240-249, May, 2014. Article (CrossRef Link).

[6] O. A. Wahab, H. Otrok and A. Mourad, "A cooperative watchdog model based on Dempster–Shafer for detecting misbehaving vehicles," *Computer Communications*, vol. 41, pp. 43-54, March, 2014. Article (CrossRef Link).

[7] S. Zhong, J. Chen, and Y.R. Yang, "Sprite: A Simple Cheat-Proof, Credit-Based System for Mobile Ad Hoc Networks," in *Proc. of 22nd Annual Joint Conf. of the IEEE Computer and Communications, INFOCOM 2003*, vol.3, pp. 1987–1997, March 30-April 3, 2003. Article (CrossRef Link).

[8] L. Anderegg and S. Eidenbenz, "Ad Hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents," in *Proc. of the 9th Annual Int. Conf. on Mobile Computing and Networking*, pp.245-259, September 14-19, 2003. Article (CrossRef Link)

[9] H. Janzadeh, K. Fayazbakhsh, M. Dehghan, and M.S. Fallah, "A Secure Credit-Based Cooperation Stimulating Mechanism for MANETs Using Hash Chains," *Future Generation Computer Systems*, vol. 25, no.8, pp. 926-934, September, 2009. Article (CrossRef Link)

[10] Q. Lian, Y. Peng, M. Yang, Z. Zhang, Y. Dai and X. Li, "Robust incentives via multilevel tit-for-tat," *Concurrency Computat.: Pract. Exper*, vol. 20, no. 2, pp. 167–178, May, 2007. Article (CrossRef Link)

[11] S. Marti, T.J. Giuli, K. Lai and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. of the 6th Annual Int. Conf. on Mobile Computing and Networking*, vol. 4, pp. 255–265, August 6 - 11, 2000. Article (CrossRef Link)
[12] P. Michiardi and R. Molva, "Core: A Collaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks," in *Proc. of the 6th IFIP Communication and Multimedia Security Conference*, pp. 107-121, September 26–27, 2002. Article (CrossRef Link)
[13] S. Buchegger and J.-Y.L. Boudec, "Performance analysis of the confidant protocol," in *Proc. of the 3rd ACM Int. Symposium on Mobile Ad Hoc Networking & Computing*, pp. 226–236, June 9-11, 2002. Article (CrossRef Link)
[14] S. Bansal and M. Baker, "Observation-based cooperation enforcement in ad hoc networks," *Technical Report, Computer Science Department, Stanford University*, 2003. Article (CrossRef Link)
[15] D.B. Johnson and D.A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, pp. 153–181, 1996. Article (CrossRef Link)
[16] O. A. Wahab, O. Hadi, and M. Azzam, "A Dempster–Shafer Based Tit-for-Tat Strategy to Regulate the Cooperation in VANET Using QoS-OLSR Protocol," *Wireless Personal Communications*, vol. 75, no. 3, pp. 1635-1667, April, 2014. Article (CrossRef Link)
[17] A. E. Khatib, A. Mourada, H. Otrokb, O. A. Wahabc, J. Bentahar, "A Cooperative Detection Model Based on Artificial Neural Network for VANET QoS-OLSR Protocol," in *Proc. of 15' IEEE Int. Conf. on Ubiquitous Wireless Broadband*, pp. 1-5, October 4-7, 2015. Article (CrossRef Link)
[18] V. N. G. J. Soares and J. J. P. C. Rodrigues, "7. Cooperation in DTN-Based Network Architectures," *Cooperative Networking, Wiley*, pp. 101–115, 2011. Article (CrossRef Link)
[19] J. Dias, J. Rodrigues, L. Shu, and S. Ullah, "Performance evaluation of a cooperative reputation system for vehicular delay-tolerant networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1-13, May, 2014. Article (CrossRef Link)
[20] J. Jude Kline, "Basic game theory," *Australian Economic Review*, vol. 33, no. 4, pp. 381–387, December, 2000. Article (CrossRef Link)
[21] L. A. Imhof, D. Fudenberg and M. A. Nowak, "Tit-for-tat or win-stay, lose-shift?" *Journal of Theoretical Biology*, vol. 247, no. 3, pp. 574-580, August, 2007. Article (CrossRef Link)
[22] F. Milan, J. J. Jaramillo and R. Srikant, "Achieving cooperation in multihop wireless networks of selfish nodes," in *Proc. of 06' Workshop on Game Theory for Communications and Networks*, October 14, 2006. Article (CrossRef Link)
[23] R. Leung, J. Liu, E. Poon, A. L. C. Chan and B. Li, "MP-DSR: a QoS-aware multi-path dynamic source routing protocol for wireless ad-hoc networks," in *Proc. of 26th Annual IEEE Conf. on Local Computer Networks*, pp. 132-141, November 14-16, 2001. Article (CrossRef Link)
[24] K. Fall and K. Varadhan, "The network simulator (ns-2)," 2007. Article (CrossRef Link)

Xiaohui Li received the B.S. degree in Computer Science from the College of Computer Science, Sichuan University, Chengdu, China in 2012. She is currently a Ph.D. student at the College of Computer Science, Sichuan University, Chengdu, China. Her research interests cover a wide variety of topics in vehicle-to-vehicle networks and spatial information networks, with emphasis on design of routing protocols and transport layer protocols.

Junfeng Wang received the M.S. degree in Computer Application Technology from Chongqing University of Posts and Telecommunications, Chongqing in 2001 and Ph.D. degree in Computer Science from University of Electronic Science and Technology of China, Chengdu in 2004. From July 2004 to August 2006, he held a postdoctoral position in Institute of Software, Chinese Academy of Sciences. From August 2006, Dr Wang is with the College of Computer Science, Sichuan University, Chengdu as a professor. His recent research interests include spatial information networks, network and information security, and cloud computing.