Wireless Pers Commun (2014) 75:1635–1667
DOI 10.1007/s11277-013-1443-y

A Dempster–Shafer Based Tit-for-Tat Strategy to Regulate the Cooperation in VANET Using QoS-OLSR Protocol

Omar Abdel Wahab · Hadi Otrok · Azzam Mourad

Published online: 17 October 2013

Abstract In this paper, we address the problem of cooperation among vehicles in VANET using QoS-OLSR protocol in the presence of selfish nodes. QoS-OLSR is a proactive protocol that considers the Quality of Service (QoS) of the nodes while electing the cluster-heads and selecting the Multi-Point Relay (MPRs) nodes. Cluster-heads and MPRs might misbehave on the roads by over-speeding or under-speeding. Classical and generous Tit-for-Tats are proposed to analyze the interaction among vehicles. However, both strategies are not able to enforce the cooperation due the fact that they (1) count on individual watchdogs monitoring, (2) rely on the node-to-node cooperation decision, (3) and ignore the high mobility and packet collisions. Therefore, we propose a Dempster–Shafer based Tit-for-Tat strategy that is able to improve the decision and regulate the cooperation in the vehicular network. This is done by (1) launching a cooperative watchdogs monitoring, (2) correlating the observations of the different watchdogs using Dempster–Shafer theory, and (3) propagating the decisions among clusters. Thereafter, we compare the Dempster–Shafer based strategy with several strategies derived from the original Tit-for-Tat. Simulation results prove that the Dempster–Shafer based strategy is able to maintain the survivability of the vehicular network in the presence of high mobility and packet collisions with minimal time and overhead.

Keywords Vehicular Ad hoc Network (VANET) · QoS-OLSR · Packet collision · Tit-for-Tat · Game theory · Dempster–Shafer · Information dissemination

O. A. Wahab · A. Mourad (✉)
Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon
e-mail: azzam.mourad@lau.edu.lb

O. A. Wahab
e-mail: omar.abdelwahab@lau.edu.lb

H. Otrok
Department of Electrical and Computer Engineering, Khalifa University of Science, Technology & Research, Abu Dhabi, UAE
e-mail: Hadi.Otrok@kustar.ac.ae

1 Introduction

Every year, millions of people around the world die in car accidents and many more are injured. Therefore, many countries resorted to the use of several safety information derivates such as speed limits and road conditions but still more work is required. Vehicular ad hoc networks (VANETs) are assumed, upon implementation, to support a wide variety of distributed applications starting from safety services such as collision avoidance systems down to commercial services such as context-aware advertisement and file sharing [30]. For example, a vehicular network may be used to warn drivers for possible traffic jams in order to reduce road congestions. It may be used as well to broadcast emergency alerts to drivers in order to avoid collisions [38].

Currently, most of the research in this field is focused on the implementation and deployment of such applications. Thus, the subject of message delivery among nodes has attracted lately the research community. A number of routing protocols [15,18,23,34,36] that can be applied to VANET has been advanced. However, the deployment of such protocols encounters several limitations. Indeed, although the routing protocol is good, the question is: "will the vehicles follow this protocol or not?". The Quality of Service Optimized Link State Routing (QoS-OLSR) protocol [31] is a proactive routing protocol modeled to cope with mobile ad hoc networks. It is based on electing a set of optimal cluster-heads and dividing the network into clusters. These heads are then responsible for selecting a set of designated nodes charged of transmitting the network topology information and forwarding the traffic flows. Such nodes are called *Multi-Point Relay* (MPR) nodes. However, some MPR nodes may, after being selected, refuse to send packets on behalf on other drivers and spend their resources for nothing. Instead, the driver chooses to over-speed the other vehicles (go beyond the maximum allowed speed) in order to get the destination as fast as he can or even to under-speed (drive below the minimum speed) for several purposes. Therefore, a detection mechanism that is able to detect and punish the misbehaving MPRs is needed. The existing detection mechanisms rely on the one-to-one relation in the sense that each node monitors its neighbors and takes the decision to cooperate or defect accordingly. Thus, these proposals are only able to guarantee the interest of individual nodes but not the interest of the whole network. Moreover, these mechanisms suffer from the ambiguous monitoring caused by the high mobility of nodes and the packet collisions. Therefore, we propose the Dempster–Shafer Tit-for-Tat strategy that is based on the cooperative detection. According to this strategy, all the neighboring nodes monitor the behavior of the MPR nodes. To make the decision cooperative and overcome the problems related to the ambiguous monitoring, we use the Dempster–Shafer to correlate the collected observations. After detecting the misbehaving nodes, we use the Tit-for-Tat strategy to regulate the cooperation by rewarding the cooperative nodes and punishing the selfish nodes.

We use, in this work, the Tit-for-Tat [21] strategy to regulate the cooperation among vehicles in VANET. In such a game, if a node refuses to help some other nodes at time t, these nodes will refrain from helping it back at time $t' > t$. However, this strategy suffers from the following limitations: (1) node-to-node cooperation decision, (2) ambiguous monitoring caused by high mobility and packet collisions, and (3) false alarms. In fact, the decision of cooperation in Tit-for-Tat as in all the reputation-based mechanisms is done locally between any pair of nodes by monitoring each other's behavior. Thus, if a node behaves selfishly it will be punished by its opponent solely. This ensures only the welfare of individual nodes but does not ensure the survivability of the network which will be full of defections and disconnections. Some nodes may behave selfishly for n times ($n > 1$) and behave well once to get paid back. Concerning the collisions and false alarms, it may happen, for example, that some packets

are not detected due to packets collisions. This collision may be caused intentionally and unintentionally. For instance, some nodes may transmit packets at the same time other nodes are transmitting in order to launch collision attack and forbid the accurate monitoring of watchdogs. This leads to false alarms by accusing cooperative vehicles to be selfish and vice versa. Furthermore, monitoring and declaring the selfish nodes is a challenging task in VANET regarding the high mobility of vehicles.

In this paper, we model the packet forwarding in VANET as a repeated non-cooperative game. We extend then this game to operate under noise considering thus the problem of high mobility and packets collisions. We also define in this game the collision problem in VANET and show the importance of considering this issue while designing any cooperation enforcement model. Thereafter, we advance three strategies based on (1) generous Tit-for-Tat, (2) Tit-for-k-Tats, (3) and Dempster–Shafer based Tit-for-Tat in order to compare their efficiency in dealing with the problem of cooperation in VANET. Simulation results reveal that the Dempster–Shafer based strategy (DS-based Tit-for-Tat) outperforms the strategies proposed in the literature since it uses a cooperative mechanism to build the decisions instead of relying on the one-to-one decision.

The strategy works as follows. First, some watchdogs are designed to monitor the behavior of the MPR nodes. Next, a voting process is launched among the watchdogs situating within the same transmission range. Thereafter, the head of each cluster aggregates the votes of the watchdogs situating within its cluster using the Dempster–Shafer theory. Finally, the cluster-head spreads the decision to all its members and warn the other clusters whenever a contact with them occurs for the purpose of reducing the implementation overhead and time. The basic idea of Dempster–Shafer is to give a weight for each vote according to the trustworthiness level attached to the voter node. Thus, the use of Dempster–Shafer is necessary to discount evidences from untrustworthy or uncertain observers upon building the final judgment.

In summary, our contribution is a cooperative model based on Dempster–Shafer Tit-for-Tat that can:

- Motivate the selfish vehicles (MPRs) to behave normally.
- Improve the detection of the misbehaving vehicle by the means of Dempster–Shafer.
- Detect the misbehaving vehicles taking into consideration the packet collisions and the high mobility using cooperative decision.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 states the tackled problem. Section 4 presents the packet forwarding model in VANET. Section 5 extends the model to consider the noise caused by packets collisions. Section 7 illustrates the simulation scenarios, describes the implementation details, and explains the Tit-for-Tat strategies with empirical results. Finally, Sect. 8 concludes the paper.

2 Related Work

Several schemes have been advanced to simulate the cooperation among nodes in ad hoc networks. These schemes can be grouped into two categories: credit-based techniques [10,19,20,39] and reputation-based techniques [7,21,24,25]. Credit-based schemes [22] offer incentives (virtual currency) for nodes versus performing networking functions. In a reputation-based approach, nodes monitor, detect, and then announce another node to be misbehaving. This announcement is then broadcasted all over the network, leading to discard

the misbehaving node from being used in all future routes [6]. In the following, we present the main contributions in both credit-based and reputation-based approaches.

2.1 Credit-Based Approaches

Lee et al. [19] proposed a method to control the commercial ad dissemination in VANETs called *receipt counting method*. In this method, the source node of a packet has to promise a fixed value for each receipt. However, this method entrains an overspending problem for the source nodes in the sense that the source node has no idea about the number of packets, which makes it unable to predict the total amount of payments.

Douceur et al. [10] proposed a mechanism called *lottree*. According to this method, one node in the network is selected periodically to get a payment. The selection of the node is accomplished in way to encourage the high participation and stimulate new entrants. The shortcoming of this mechanism is that the whole payment is granted for exclusively one winner, which would discourage the conservative nodes from collaborating due to the fact that they have poor chances to win.

FRAME [20] is composed of two phases: Weighted rewarding component and Sweepstake component. In the weighted rewarding component, the nodes are assigned weighted rewards based on their contributions. The sweepstake component gives the winner participating node a fixed amount of payment. Nevertheless, the limitation of this strategy is that the nodes are motivated to avoid the intermediate nodes and get connected directly to the destination with the intention to increase their contribution weight.

SPRITE [39] takes advantage of the Vickrey Clarke Groves (VCG) mechanism [1] to select the best available single path. This approach designs a game theory to specify the charges and credits and motivates then each node to truthfully declare its actions. However, the need for a Credit Clearance System is the shortcoming that encounters this approach.

Overall, the main idea of the credit-based approaches is that the nodes receive payments to serve others and give payments to get served. However, the shortcoming that limit the efficiency of these approaches are the lack of scalability and centralization, and the need for a tamper-proof hardware.

2.2 Reputation-Based approaches

In the Tit-for-Tat [21], a reputation value is assigned for each node and the nodes have the chance to increase their own reputations by cooperating with more reputable nodes. This reputation is used later to distribute the network services among nodes. However, this strategy suffers from three main problems. First, the decision of cooperation is restricted to the local relation of each pair of nodes. Second, it ignores the cases of high mobility and collisions that may hinder the monitoring process. Finally, this method ends up with a deadlock where no node is willing to cooperate with any other node.

Marti et al. [24] employed the watchdog and pathrater concepts in the Dynamic Source Routing (DSR) [17] protocol. This approach detects the misbehaving nodes but does not punish them. Instead, it prevents the detected misbehaving nodes from forwarding packets. Nonetheless, this would motivate the misbehavior in the network since the misbehaving nodes are rewarded for their behavior as their packets keep being forwarded by others while they do not have to transmit and spend resources.

In CONFIDANT [7], an alarm is sent to the nodes in the network whenever a misbehaving node is detected. The objective is to isolate these nodes from the whole network. However, the problem of such approach is how accurate or credible the alarms are.

In summary, in the reputation-based schemes, the nodes detect the misbehaving nodes and then propagate them all over the network, which results in discarding these preventing nodes from being used in all the future routes. One advantage of these approaches over the credit-based ones is that they do not rely on a specialized hardware. However, the reputation-based mechanisms suffer from several limitations such as: ambiguous collision, limited transmission power, false alarms, and non-cooperative cooperation/defection decision.

In this paper, we extend the concept of watchdogs in order to regulate the cooperation inside a vehicular network. The main difference with the previous reputation-based contributions is that the decision of cooperation in our model does not rely solely on the node-to-node relation where each node called watchdog monitors the behavior of its neighbor to decide whether to cooperate or not. The previous proposals are only able to guarantee the interest of individual nodes but not the interest of the whole network. In fact, a node may cooperate with some nodes and refrain from cooperating with the other nodes. In such a way, this node will be punished by some nodes and rewarded by others. Thus, this node will continue defecting whenever it believes that this achieves its interest; causing hence negative implications on the network. In our work, each 1-hop away neighbor of a node is set as a watchdog to monitor the behavior of this node. The decision of cooperation is based then on an aggregated collective decision of these watchdogs using Dempster–Shafer theory. Moreover, the set of selfish nodes is propagated within the cluster members and among the clusters. This guarantees that the selfish nodes will be punished by the different nodes instead of being punished by some nodes and served by others. Thus, we are controlling the cooperation inside the whole network instead of just regulating the relation between a pair of nodes. This ensures the continuity and the survivability of the network. Such mechanism is able as well to overcome the monitoring ambiguity caused either by the packets collisions or by the high mobility of vehicles since it is based on aggregated decision.

3 Problem Statement

Selfishness is a normal behavior that is present in all the aspects of life and the VANETs are not an exception. After being elected, some MPRs may have a selfish thinking that pushes them to stop collaborating with other nodes. These nodes seek to realize their own objectives regardless of the bad consequences that may result. This thinking stems from the fact that the driver prefers to over-speed (go beyond the maximum allowed speed) the other vehicles and get his destination as earlier as he can. He considers the collaboration in the networking functions as a waste of time since he is spending his time sending packets on behalf of other vehicles without receiving any compensation. Some other vehicles can also under-speed (drive below the minimum speed) for several purposes. This type of attack is called *passive* since the purpose of the misbehaving nodes is to increase their own benefits and not to interrupt the normal operation of the network [16]. This behavior that aims only to satisfy the driver's demands, does not seek to harm the network functioning on purpose. However, this does not mean that such behaviors do not induce dangerous implications. Assume for example that a node serving as an MPR between two clusters decided to over or under speed. This may entail catastrophic implications in the sense that (1) the number of elected MPRs increases frantically due to the need of frequent MPRs reelections which increases the jamming over the network, (2) the network stability, measured as current number of nodes in each cluster divided by the previous number of nodes that was in it, deteriorates effectively and the number of clusters disconnections will hence be high, (3) the end-to-end

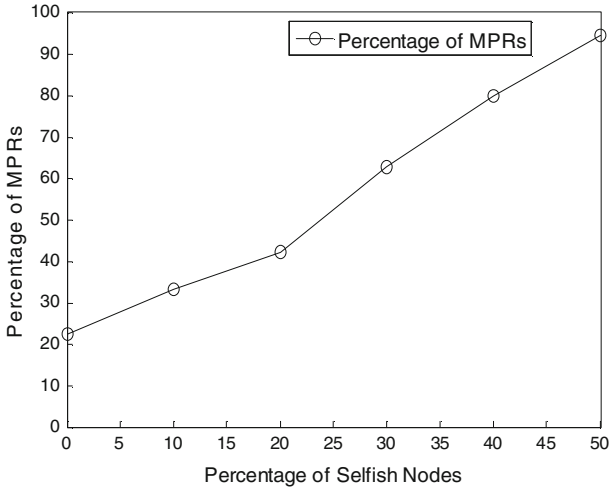


Fig. 1 Impact of the selfish nodes on the percentage of MPRs

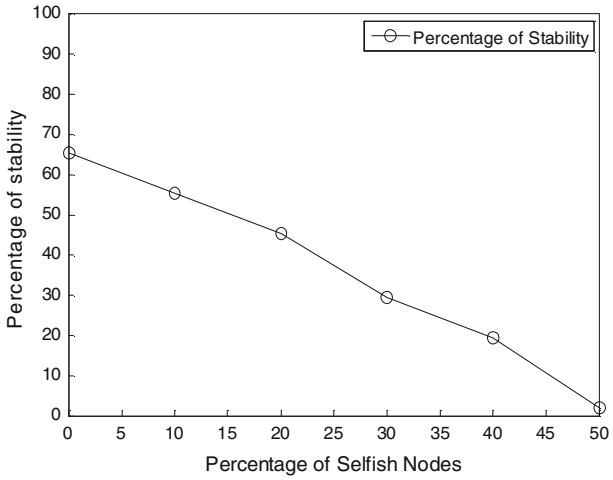


Fig. 2 Impact of the selfish nodes on the percentage of stability

delay or the average number of hops needed to transfer data between the source and the destination is strongly increased by the fact that the path will not stand up more than few seconds, (4) and the bandwidth allocation will suffer from recurrent disconnections. Figures 1, 2, 3 and 4 explains the impact of the selfish nodes on the network in terms of percentage of MPRs, percentage of stability, percentage of clusters disconnections, and average path length respectively. The number of nodes used to simulate these figures is 100. The percentage of misbehaving nodes used in the simulations ranges from 0 (without selfish nodes) to 50 % of the total nodes gradually (The selection of this interval and all the simulation details are explained in Sect. 7).

Figure 1 shows that the percentage of MPRs increases as the percentage of selfish nodes increases. This is due to the fact that the clusters will disconnect frequently due to the high mobility of the selfish nodes. Figure 2 reveals that the increase in the percentage of selfish

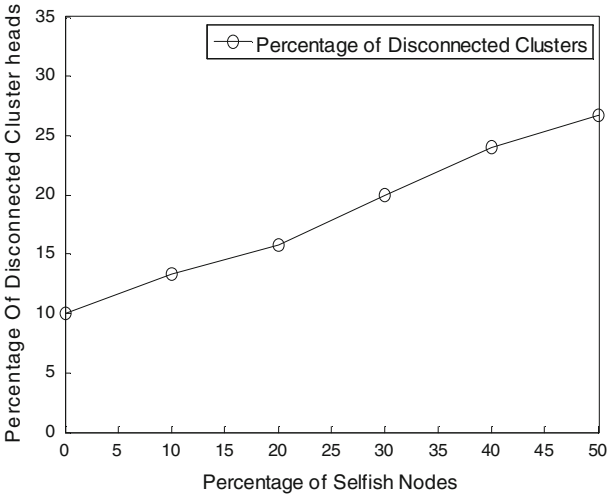


Fig. 3 Impact of the selfish nodes on the percentage of disconnections

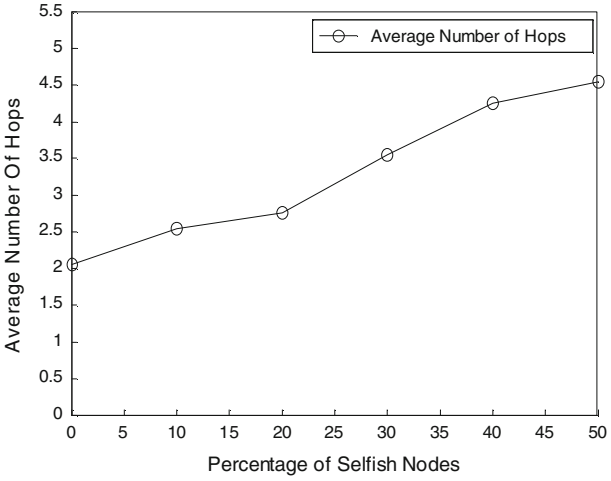


Fig. 4 Impact of the selfish nodes on the number of hops

nodes decreases the stability in the network. This is justified by the fact that the under-speeding vehicles remain for a long time in the same cluster whereas the over-speeding vehicles move very quickly to other clusters. In Fig. 3, we can notice that percentage of disconnected cluster-heads keeps increasing as long as the percentage of selfish nodes is increasing. This is a normal result of the decrease in the percentage of stability. Finally, Fig. 4 shows that the End-to-End delay will increase gradually with the increase of the percentage of selfish nodes. This is because the paths connecting the clusters will be frequently disconnected and the intended packets between the clusters will not be received on time.

Based on the above, it is highly important to develop a model that is able to regulate the cooperation among nodes in VANETs.

4 Game Model Without Collisions

Game theory [35] is a formal study of conflict and cooperation that applies whenever the actions of several peers are interdependent. In VANETs, vehicles are independent nodes, making decisions about cooperating or not. While building these decisions, nodes may behave selfishly paying attention for only their own interests. This makes the objectives of the different nodes conflicting (some nodes need to be served and others consider that their interests lie in being uncooperative). Thus, the application of game theory in dealing with selfish nodes in VANET may be straightforward, as game theory usually analyzes situations in which player purposes are in conflict. Therefore, we decided to model the cooperation among nodes in VANET as non-cooperative repeated game where the players are the set of MPR nodes responsible for relaying the packets. These nodes are assumed to be rational or selfish; namely, they seek to maximize their own payoff, not to cause damage for the other nodes. The game can be modeled as follows. The desired outcome of the game is achieved if the routing is done along a continuous path without any packet dropping. The players are the MPR nodes that cooperate in the packets forwarding inside the network. The group of players is a finite set that we denote by N and single players are indicated by $i \in N$. A_i is used to indicate the set of all potential actions of i while a_i denotes the action done by player i. Each player has to choose either to *forward* the packet or to *drop* it; thus, $A_i = \{Forward, Drop\}$.

Definition A Packet Relaying Game in VANET is

$$G = \langle N, \{d_i\}, \{G_i\} \rangle$$

where:

- N denotes the collection of players
- $0 \leq d_i \leq 1$ represents the dropping probability of player i
- G_i is the gain or payoff of player i

Since relaying consumes node's bandwidth, time, and storage space, *Forward* action should entail a cost. We assume this cost to be -1. *Drop* action, conversely, does not involve a cost. Additionally, successfully forwarded packets yield a gain of $\beta > 0$, whereas dropping the packets costs $-\beta$. In such a way, the game is characterized by the fact that the *Drop* action strictly dominates the *Forward* action. Indeed, when both players ignore each others decisions their best strategy resides in choosing to *drop* in the intention to avoid the $-\beta - 1$ cost (Table 1) which is the worst case since:

$$\beta > \beta - 1 > -\beta > -\beta - 1 \tag{1}$$

Thus, the strategy (*Drop*,*Drop*) represents the Nash Equilibrium [11] since no player can find its profit by deviating from it.

Lemma *The Nash Equilibrium in the Packet Forwarding Game represents the reciprocal defection, i.e.,* $d_i = 1$ *for* $i = 1, 2$ *is the unique Nash Equilibrium for the game* G.

Table 1 Payoff matrix of the packet relaying game

	F	D
F	$(\beta - 1, \beta - 1)$	$(-\beta - 1, \beta)$
D	$(\beta, -\beta - 1)$	$(-\beta, -\beta)$

Table 2 Payoff matrix of the Prisoner's Dilemma

	C	D
C	(R, R)	(S, T)
D	(T, S)	(P, P)

This leads us to the classical Prisoner's Dilemma [3] identified by the payoff matrix presented in Table 2, in addition to the following inequalities:

(1) $T > R > P > S$.
(2) $R > \frac{T+S}{2}$.

Hence, the packet forwarding game is equals to the Prisoner's Dilemma if and only if:

(1) Equation (1) is valid
(2) $\beta - 1 > \frac{-\beta-1+\beta}{2} \Longrightarrow \beta - 1 > -\frac{1}{2} \Longrightarrow \beta > \frac{1}{2}$

Since the Nash Equilibrium is achieved with the strategy (*Drop*,*Drop*), the rational player will always drop the packets if the game is played once. However, if the game is played infinitely this is not the case. Nonetheless, the packet forwarding game cannot resemble the classical version of Iterated Prisoner's Dilemma game [3]. This is due the fact that the interaction in the traditional Iterated Prisoner's Dilemma is basically synchronous, while the forwarding model necessitates an asynchronous interaction. Following the alternating game [29], the symmetry between the players is broken. In fact, two players can alternatively forward and receive packets. In such a case, the payoff values for one unit are like those in one round of simultaneous Prisoner's Dilemma:

Reward R = a + b
Punishment P = c + d
Temptation T = c + b
Sucker S = a + d

where a is a negative payoff representing the forwarding cost, b is a positive payoff representing the reward of being cooperated (served), c is null payoff representing the cost of dropping, and d is a negative payoff representing the cost of being defected (not served).

5 Game Model with Collisions

A major problem may face the implementation of the reputation-based strategies which is the packets collisions [2]. This problem that may prevent the players from successfully hearing a packet being forwarded could occur in different scenarios.

Scenario 1: Suppose that node V_3 is monitoring its 1-hop away neighbor, node V_2. As depicted in Fig. 5, node V_3 is located within the transmission range of node V_2 and therefore node V_3 can use promiscuous monitoring to detect whether node V_2 is forwarding packets as expected. Now assume that node V_3 has sent a packet to node V_2 to be forwarded later to node V_1 and is waiting to see if node V_2 will relay the packet to node V_1 or not. Simultaneously, vehicle V_3 is within the transmission range of vehicle V_4. If vehicle V_4 decided to forward some packets at the same time vehicle V_2 is transmitting vehicle V_3's packet to vehicle V_1, then vehicle V_3, which is monitoring vehicle V_2, will observe a collision of vehicle V_2's and V_4's transmission and will thus be unable to observe vehicle V_2's transmission. Vehicle V_2's transmission to V_1 might actually have been successful since node V_4 is out of range of

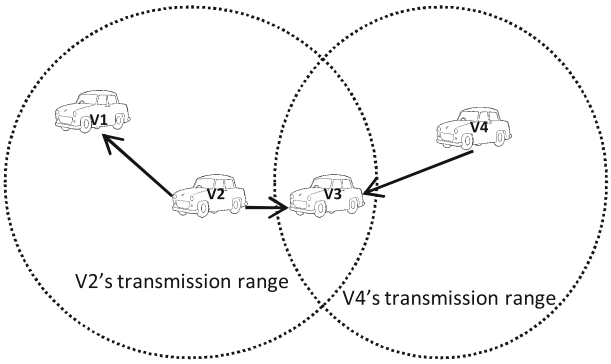


Fig. 5 Packets collision scenario

both vehicles V_1 and V_2. However, although V_2 forwarded the packet as expected, vehicle V_3 did not see that. Consequently, node V_3 may misleadingly accuse vehicle V_2 to be selfishly dropping the packet.

Scenario 2: The collision may occur also if at the same time vehicle V_2 attempts to forward a packet to vehicle V_1, vehicle V_1 relays a packet. That will cause a collision that forbids vehicle V_3 from determining whether it is within V_1's transmission range or not. If vehicle V_2 does not retransmit the packet, vehicle V_1 will not receive the packet. Thus, vehicle V_3 actually thinks that V_2 has successfully transmitted the packet and therefore will not be able to identify node V_2's malicious packet dropping behavior. Thus, vehicle V_2 can launch such collisions intentionally in order to hamper V_3's promiscuous monitoring. For instance, V_2 may wait until vehicle V_1 begins forwarding a packet to initiate the transmission for vehicle V_3's packet, generating thus an intentional collision.

We model this situation using a Prisoner's Dilemma game with Noise [37]. However, in such a game the real dropping probability d_i of a node is unknown to the other nodes due to the ambiguity caused by both high mobility and collisions. We incorporate therefore the notion of *perceived defection rate* [27] to prevent the nodes from overestimating d_i in order to earn an excuse for being uncooperative. Let γ indicate the probability at time t with which each node tries to transmit. The Perceived Defection of player i at stage k, is represented by $\hat{p}_i^{(k)}$, is:

$$\hat{p}_i^{(k)} = \gamma + (1 - \gamma) \times d_i^{(k)}$$

If the Tit-for-Tat strategy is applied, the situation will end up with a mutual deadlock where no node will cooperate with any other one. In fact, two players playing Tit-for-Tat will *"cooperate on the first move, then do what the opponent did in the last move"*. Thus, a strategy is Tit-For-Tat if:

- $d_i^{(0)} = 0$ (cooperate on the first move)
- $d_i^{(k)} = d_j^{(k-1)}$ for $k > 0$ (do what the opponent j did in the last move)

Thus, we can write the following equations:

Initially, the two players cooperate:

$$d_1^{(0)} = d_2^{(0)} = 0$$

The high mobility or the packet collisions will cause the perceived defection of the player to be:

$$\hat{p}_1^{(0)} = \hat{p}_2^{(0)} = \gamma$$

At stage 1, a mutual punishment will take place and the defection probability will be:

$$d_1^{(1)} = d_2^{(1)} = \gamma$$

The perceived defection will hence be:

$$\hat{p}_1^{(1)} = \hat{p}_2^{(1)} = \gamma + (1-\gamma) \times \gamma$$

At stage k, the dropping probability of each player will be:

$$d_1^{(k)} = d_2^{(k)} = 1 - (1-\gamma)^k$$

and the perceived defection will be:

$$\hat{p}_1^{(k)} = \hat{p}_2^{(k)} = 1 - (1-\gamma)^{k+1}$$

As the number of iterations in iterated Tit-for-Tat tends to infinity, we get:

$$d_1 = d_2 = \lim_{k\to\infty} d_1^{(k)} = \lim_{k\to\infty} d_2^{(k)} = 1 \tag{2}$$

We follow in this work the infinite backlog queuing model [26] where each node separates the packets originating from it from the transit packets originating from other neighbors by allocating an independent queue for each type of packets. Therefore we are able to assume in the above calculations that the traffic load γ is a constant that does not rely on the dropping probability d_i.

The Eq. 2 reveals that two playing Tit-for-Tat will end up with mutual punishment even when both players want to cooperate. A way to deal with this issue is by using a more generous strategy able to break the mutual retaliation problem. Such a strategy is called Generous TFT (GTFT) [4]. According to this strategy, a cooperative player will cooperate with another player at a regular basis of k movements regardless of their previous history. Moreover, only one cooperation in the past k decisions is enough to consider the other player cooperative. Although this approach is efficient with nodes that do not cooperate at all, it allows the selfish nodes to mimic the behavior of cooperative nodes by cooperating once every time they notice that their history become full of defections. Therefore, we propose in the following a Dempster–Shafer based Tit-for-Tat model that is able to accurately detect and punish the selfish nodes in VANET in the presence of collisions and high mobility and without giving the misbehaving nodes the chance to imitate the behavior of cooperative nodes.

6 Game Model Analysis

We give in the following a mathematical analysis to show that our proposed strategy can lead to a positive gain for the cooperative nodes. To do so, we will consider a network of "n" nodes where "c" of them are cooperative and "s" are selfish, with c = n − s and s = n − c. At each iteration, only one node (source) may demand a forwarding request. So, the probability of requesting is $p_r = \frac{1}{n}$. The other vehicles can either reject the request or cooperate and forward the packets. In the former case, the vehicle can be selfish or, simply, does not have enough

resources (bandwidth, storage space) or time with probability $1 - p_a$ where p_a represents the probability of responding to a request. The game parameters satisfying the conditions of the asynchronous repeated Prisoner's Dilemma game are presented as follows:

- Forwarding Cost: a $= -1$
- Drop Cost: c $= 0$
- Gain from a fulfilled request: b $= \beta = \frac{1}{pr}$
- Loss from a non-fulfilled request: d $= -\beta = -1$

We present an inequality, based on the entropy in information theory [9], to calculate the average gain for the cooperative vehicles. This inequality assures that the average gain for the cooperative vehicles is strictly positive:

$$p_r \times b + (1 - p_r)\{p_a \times a + (1 - p_a) \times c\} > 0$$

The above inequality is strictly positive since (1) all the parameters (a, b, and c) are ≥ 0, and (2) $1 - p_r > 0$ since $p_r < 1$. Moreover, the inequality assumes that (1) a vehicle is considered cooperative by the other vehicles and (2) the other cooperative vehicles have enough available resources and time to fulfill the requests of the requesting vehicle. If we take off the latter condition, the average gain for a cooperative node becomes:

$$p_r\{(1 - (1 - p_a)^{c-1}) \times b + (1 - p_a)^{c-1} \times d\} + (1 - p_r)\{p_a \times a + (1 - p_a) \times c\} > 0$$

The above inequality is strictly positive since (1) $1 - (1 - p_a) > 0$, (2) $1 - p_a > 0$, and (3) the only negative parameter is d and it is added to a value that is greater than it i.e., $(1 - (1 - p_a)^{c-1}) \times b > (1 - p_a)^{c-1} \times d$ since $b > d$. In this inequality:

- p_r represents the probability of requesting,
- $1 - (1 - p_a)^{c-1}$ means that the cooperative vehicles, except for the requesting vehicle $(c - 1)$, have enough resources and time to forward packets.
- $(1 - p_a)^{c-1}$ means that the cooperative vehicles, except for the requesting vehicle $(c - 1)$, do not have enough resources and time to forward packets.
- $(1 - (1 - p_a)^{c-1}) \times b$ represents the gain yielded by the requesting vehicle when it got served by the other cooperative vehicles $(c - 1)$ that have enough resources and time to cooperate.
- $(1 - p_a)^{c-1} \times d$ represents the loss imposed the requesting vehicle when it is not served by the other cooperative vehicles $(c - 1)$ that suffer from a lack of available resources and time.
- $p_r\{(1 - (1 - p_a)^{c-1}) \times b + (1 - p_a)^{c-1} \times d\}$ represents the gain or loss received by the requesting vehicles as a result of getting served or not.
- $1 - p_r$ represents the probability of non-requesting.
- $p_a \times a$ represents the cost of forwarding a packet.
- $(1 - p_a) \times c$ represents the cost resulting from dropping a packet for the vehicles that suffer from a lack of available resources and time.
- $(1 - p_r)\{p_a \times a + (1 - p_a) \times c\}$ represents the gain or loss received by the non-requesting vehicles as a result of forwarding or dropping the packets.

According to our Dempster–Shafer based Tit-for-Tat strategy, a MPR node is considered as cooperative if the belief in trustworthiness of this node is greater than 0.5. To compute the average gain for cooperative vehicles, we made 600,000 requests sequentially. The maximal number of requests which can be fulfilled is calculated by the following: Number of requests $\times$ ratio of cooperative nodes $= 600{,}000 \times \frac{c}{n}$. Since our strategy does not depend on the recent past history of the nodes but on an aggregated decision, which

calculates a belief for each node according to different observations to judge the nodes, the average gain of the cooperative nodes in our strategy is computed according to the following formula: $600{,}000 \times \frac{c}{n} \times [a \times (1-(1-p_a)^{c-1}) + b \times (1-(1-p_a)^{c-1}) + (1-p_a)^{c-1} \times d] = 600{,}000 \times \frac{c}{n} \times [(1-(1-p_a)^{c-1}) \times (b+a) + (1-p_a)^{c-1} \times d]$, where:

- $600{,}000 \times \frac{c}{n}$ represents the maximal number of requests that can be fulfilled.
- $a \times (1-(1-p_a)^{c-1})$ represents the cost received by the cooperative vehicles, except for the requesting vehicle $(c-1)$, as a result of forwarding the packets.
- $b \times (1-(1-p_a)^{c-1})$ represents the gain received by the requesting vehicle as a result of getting served by the other cooperative vehicles $(c-1)$ having enough resources and time to cooperate.
- $(1-p_a)^{c-1} \times d$ represents the loss imposed on the requesting vehicle as a result of not getting served by the other cooperative vehicles $(c-1)$ that do not have enough resources and time.

Let's consider now that n = 0, c = 4, and s = 6. Then, the gain for cooperative nodes, according to our strategy, will be: $600{,}000 \times \frac{14}{20} \times [(1-(1-\frac{1}{5})^{14-1}) \times (20+(-1)) + (1-\frac{1}{5})^{14-1} \times -1] = 420{,}000 \times [0.945 \times 19 + -0.055] = 7{,}518{,}000$.

To show the efficiency of the cooperative decision proposed in our strategy, we will compute, in the following, the average gain of the cooperative vehicles according to the Generous Tit-for-Tat strategy. This strategy, which relies on the node-to-node decision, assumes that a cooperative vehicle should fulfill a request to a requesting vehicle if this latter has been cooperative with it at least $\frac{k}{2}$ times in the last k actions. This generous strategy can stimulate the rational selfish vehicles to cooperate at least with probability of $\frac{p_a}{2}$, if they want to mimic the behavior of the cooperative vehicles and obtain some gain. However, this strategy cannot always maximize the total gain of the nodes even if selfish vehicles began to be cooperative. In this strategy, the rational vehicles may adjust their behavior to cooperate with a probability of $\frac{p_a}{2}$ since the nodes are considered cooperative if they cooperate with such a probability. In such a case, the selfish vehicles are still considered as cooperative even they are cooperating less. Practically, the gain of the cooperative vehicles according to this strategy would be reduced significantly to become $600{,}000 \times \frac{14}{20} \times [(1-(1-\frac{1}{10})^{14-1}) \times (20+(-1)) + (1-\frac{1}{10})^{14-1} \times -1] = 420{,}000 \times [0.746 \times 19 + -0.254] = 5{,}846{,}400$.

7 Tit-for-Tat Strategies

In this section, we explain the settings and introduce the assumptions that we considered when formulating the game. We describe then the details of the implementation and the scenarios we followed during the simulations. Thereafter, we analyze the behavior of the VANET nodes using different Tit-for-Tat strategies in order to select the best strategy able to enforce the cooperation in VANET in the presence of high mobility and collisions. This can be achieved by increasing the gain of cooperative node and decreasing the gain of selfish nodes.

7.1 Set-up and Simulation Scenarios

Consider that we have a group of N players (MPRs) in a packet relaying game, each node is a member of one cluster at a time and the routing is done according to a clustered-based QoS-OLSR protocol [5]. Each player is able to:

(1) Forward a packet.
(2) Drop a packet because of the inability to forward such as lack of bandwidth, transmission power, or time.
(3) Drop a packet although it is able to relay it. Such behavior is known as "selfish" or "rational". These nodes represent a threat for the stability and the functioning of the network as shown in the Sect. 3.

The game will run for around 1 week (600,000 iterations) where each iteration represents a second. At each iteration (t), only one node (source) may demand a forwarding request. So, at time t, a randomly selected plauer i, makes a request r. The relay nodes (MPRs) can either decline the request or cooperate by forwarding the packets. In the former case, a node j can be selfish or, simply, does not have sufficient resources (bandwidth, storage space, time). In the latter case, a player j decides to cooperate and forward the packet according to the past history of node i (the expression of this fact differs between the proposed strategies). According to their cooperation in the game, the gain of the nodes is calculated.

The following asynchronous Prisoner's Dilemma game is followed while evaluating the different strategies:

– We have a total of 20 MPRs where the percentage of selfish nodes varies from 0 to 50 %.
– At each iteration, a non-requesting cooperative vehicle j would relay a packet with a probability of $P_a = \frac{1}{5}$.
– 600,000 requests are made sequentially.
– In each iteration, a particular source node is chosen randomly to make a request. Thus, the probability of requesting is $P_r = \frac{1}{20}$, for any given node.
– This player may request one or more packets to be forwarded. If a node receives more than one packet at a time it will save them into the transit queue according to the infinite backlog queuing model (Sect. 5).

In the following, we define the game parameters that can satisfy the conditions of the asynchronous repeated Prisoner's Dilemma game:

– Forwarding Cost: a $= -1$
– Drop Cost: c $= 0$
– Gain from a fulfilled request: b $= \beta = \frac{1}{p_r}$
– Loss from a non-fulfilled request: d $= -\beta = -1$

Note that the parameter a is given a negative value to represent the cost of responding to a request since it requires resources (bandwidth, storage space) and time to relay a packet. The parameter c is hence greater than a, which means that dropping the packet would be more beneficial for the rational vehicle if the game is played one-shot. Furthermore, $c - a$ is less than $b - d$ showing that the cost of serve (cooperate) is less than the benefit of being served. Therefore, for the longer term, rational users are better off cooperating with each other. Recall that the parameter b which is equals to β is satisfying the aforementioned constraint allowing our packet forwarding game to be equivalent to a Prisoner's Dilemma game ($\beta > \frac{1}{2}$ and $\beta > \beta - 1 > -\beta > -\beta - 1$).

The gain of the cooperative nodes is affected by the behavior of the selfish nodes. To show impact of such behavior on the gain of cooperative vehicles, we consider five different scenarios.

– Scenario 1: There are 100 vehicles and all of them are cooperative.
– Scenario 2: There are 100 vehicles, 80 % of them are cooperative and 20 % are selfish.
– Scenario 3: There are 100 vehicles, 70 % of them are cooperative and 30 % are selfish.

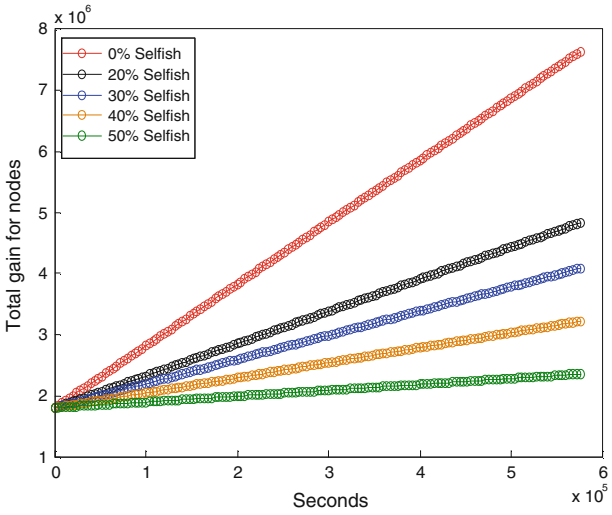


Fig. 6 The optimal upper bounds

– Scenario 4: There are 100 vehicles, 60 % of them are cooperative and 40 % are selfish.
– Scenario 5: There are 100 vehicles, 50 % of them are cooperative and 50 % are selfish.

The number of selfish nodes used in the simulations varies from 0% to 50% of the total nodes. For 0 % of selfish nodes, we simulate the behavior of cooperative nodes. From 10 to 50 % of selfish nodes, the impact of the misbehaving nodes will be catastrophic on the network as depicted in the Sect. 3. Above 50 %, the misbehaving nodes will form the majority and their negative impact begins to diminish gradually since they can meet again, form new clusters, and resume the network functions anew.

Figure 6 describes the impact of the existence of selfish nodes on the gain of the cooperative vehicles. As depicted in the figure, this gain will decrease gradually as long as the percentage of selfish nodes is increasing. This loss can be turned into gain if the selfish users were somehow forced to cooperate. Here lies the importance of developing a cooperation enforcement model that can stimulate the nodes cooperating and achieving their common interests.

7.2 Implementation Details

Matlab 9 [13] has been used to simulate the Tit-for-Tat strategies, the without Dempster–Shafer model (averaging model), and the with Dempster–Shafer model. VanetMobiSim [14] is also used as traffic simulator. It is an open source that is able to generate realistic mobility parameters dedicated to VANET. VanetMobiSim employs XML code to represent the network features such as number of nodes, topography, velocity, duration and time steps. It supports both micro-mobility and macro-mobility features. Macro mobility model refers to road topology namely the number of lanes, the traffic light constraints, speed limits, etc whereas micro mobility is concerned more by driving behavior [12]. This code is parsed then using Matlab to be used later for mobility representation. The multi-lane highway topology [32] was used in our simulations. The minimum speed on this highway is set to 60 km/h and the maximum speed is set to 120 km/h. The reputation of all the nodes is set initially to 100

Table 3 Simulation parameters

Parameter	Value
Simulation area	3,000 × 1,000 m
Number of nodes	Between 30 and 100
Transmission range	300 m
Packet size	1 kb
Idle Time	Random value in [0.1]
Link bandwidth	2 Mbps
Available bandwidth	Idle time × Link bandwidth
Pause time	10 s
Initial reputation	100
Minimum speed	60 km/h
Maximum allowed speed	120 km/h

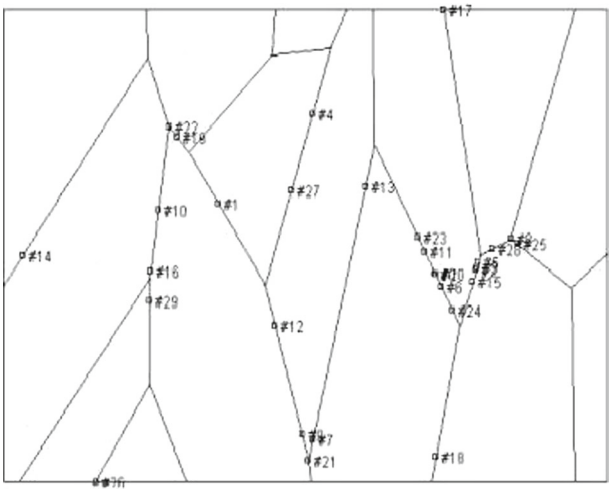


Fig. 7 Graph of streets of our vehicular movement simulations

and is updated continuously according to the payment algorithms. The simulation parameters are summarized in Table 3.

Figure 7 illustrate the generated graph streets used to simulate the models.

7.3 Traditional Tit-for-Tat

According to this strategy, the node starts by cooperating, and then imitates the behavior of its opponent in the prior iterations. In an iterated game, we assume that each player j maintains the historic records $H_{ji}(_k)$ of the last K actions with another player i. Each $H_{ji}(t)$ value corresponds to the cooperation decision $D_{ij}(t)$ taken by player i to cooperate or not with j at time t. Then depending on the historic record kept from $H_{ji}(1)$ to $H_{ji}(k)$ player j will make a decision $D_{ji}(t)$ to cooperate with i or not. If the accumulated value from $H_{ji}(1)$ to $H_{ji}(k)$ is bigger than $\frac{k}{2}$, player i will be considered cooperative and player j will try to cooperate with player i; otherwise, player j will defect. However, the cooperation decision depends as

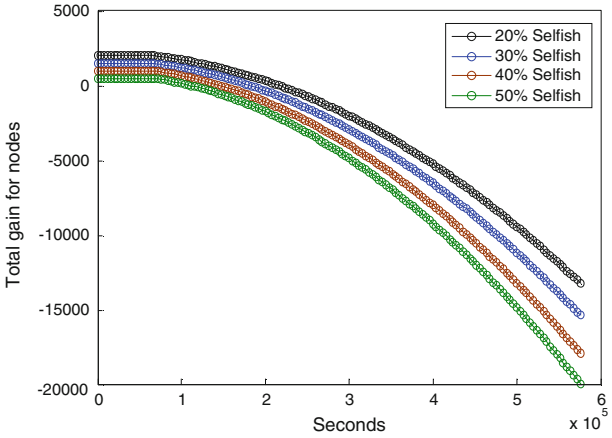


Fig. 8 Classical Tit-for-Tat

well on other factors such as the storage space and the available resources. Let $R_{ji}(t)$ be the forwarding request made from i to j at time t. Formally, vehicle j cooperates by responding to $i's$ request $R_{ji}(t)$ if (1) the current transit queue of j is not full i.e. $Q(j) < C(j)$ where $Q(j)$ is the current transit queue of i and $C(i)$ is the storage capacity of i, (2) j has $B(j)$ available resources (bandwidth), and (3) node i was cooperative with $j's$ requests in the last k iterations i.e. $\max_{1 \leq h \leq k} H_{ji}(h)$. This can be interpreted by the following equation:

$$D_{ij}(t) = min \left\{ Q(j) < C(j), B(j), \max_{1 \leq h \leq k} H_{ji}(h) \right\}$$

Figure 8 illustrates the progress of total gain of cooperative nodes over the time. It reveals that that this gain begins by an increase until reaching 1 h and 30 min (100 min). Starting from this time, the payoff of the cooperative nodes reaches a deadlock and begins to decrease as proven in the Sect.5. In fact, at this time each vehicle will have a bad history of all other nodes and will hence refrain from cooperating at all. This justifies the continuous decrease of the gain till the end caused by the loss from a non-fulfilled forwarding request.

7.4 Generous Tit-for-Tat

The classical version of Tit-for-Tat strategy suffers from several limitations. First, this strategy will end up with a mutual deadlock where no node will cooperate with any other node as proven before. Moreover, according to this strategy, a vehicle can, intentionally or unintentionally, (1) betray its opponent (false positive), (2) cooperate in error (false negative), or (3) get misinterpreted (collisions). To overcome the problems related to deadlock, false positives and false negatives, several enhancements have been made to the original Tit-for-Tat. Generous Tit-for-Tat (GTFT) [28], is a variation of the traditional Tit-for-Tat. This strategy forgives periodic defections with a certain probability. Thus, a cooperative GTFT player j will cooperate with player i at a regular basis of k movements regardless of their previous history from $H_{ji}(1)$ to $H_{ji}(k)$. Moreover, only one cooperation in the past k decisions is enough to consider the other player cooperative, instead of $\frac{k}{2}$ cooperations in the previous classical Tit-for-Tat model. Let $f_{ji}(t)$ be a fulfilled request by vehicle j to vehicle i at time t. The GTFT corresponds to the following equation:

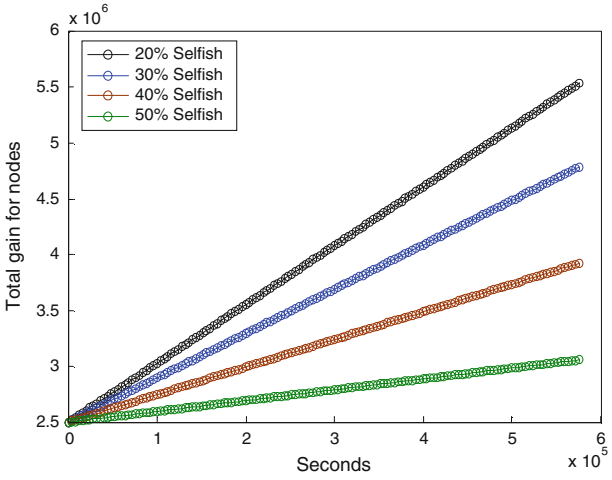


Fig. 9 Generous Tit-for-Tat

$$D_{ji}(t) = \begin{Bmatrix} min\left\{ Q(j) < C(j), B(j), \quad \max\limits_{1 \le h \le k} H_{ji}(h) \right\} \\ if\, H_{ji}(h) \neq \emptyset, \quad for\, some\, h, \\ f_{ji}(t), \quad every\, k\, requests(bonus), \\ 0, \quad otherwise \end{Bmatrix}$$

Figure 9 reveals that the total gain of the cooperative nodes is somehow close to the optimal upper bound compared to the traditional Tit-for-Tat. The figure shows also that this strategy does not cause a deadlock as observed in the traditional Tit-for-Tat. This is due to the generous characteristics preventing the cooperative users from having mutual bad history of each others in the sense that only one cooperation in the short past history is needed to consider a node cooperative. The generous strategy is good in the case of having selfish users that do not cooperate at all. Indeed, even the generous behavior results in them getting served every k turns, the cumulative loss of the nodes of not getting served is much higher which results in the drop of their total gain over the time. However, selfish nodes may try to mimic the behavior of cooperative vehicles. Thus, every time a selfish node notices that its history is full of defections, it cooperates once. Such behavior will break the strategy objectives and make the selfish nodes indistinguishable from the cooperative ones. This gives the selfish nodes a gain higher than the cooperative ones since these nodes are saving their resources and getting a gain similar to the cooperative nodes as depicted in Fig. 10. Consequently, the rational vehicles will find that their interest lies in the defection. Thus, the game goes on vicious circle.

7.5 Tit-for-k-Tats

Tit-for-Two-Tats [4] is a new form of generous Tit-for-Tat. The difference between these two strategies is the degree of generosity the strategy follows. In the traditional form of Tit-for-Tat, a node responds by defecting once it detects that its opponent has defected in the previous round. This has the effect of producing mutual retaliation which would result in a poor outcome for both players. A Tit-for-Two-Tats player will forgive first defection in

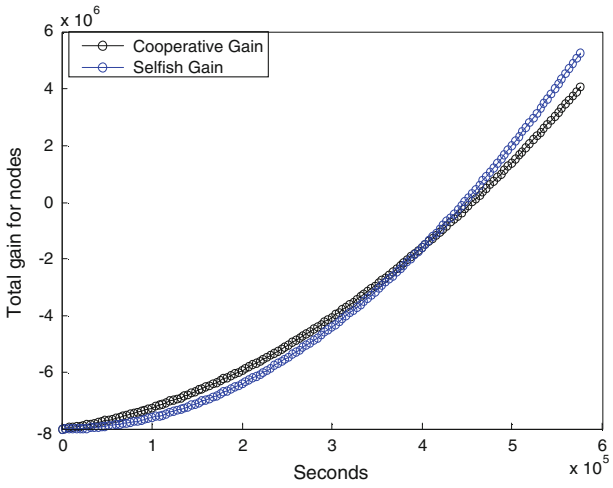


Fig. 10 Comparison between the selfish and cooperative gain in generous Tit-for-Tat with 20 % selfish nodes

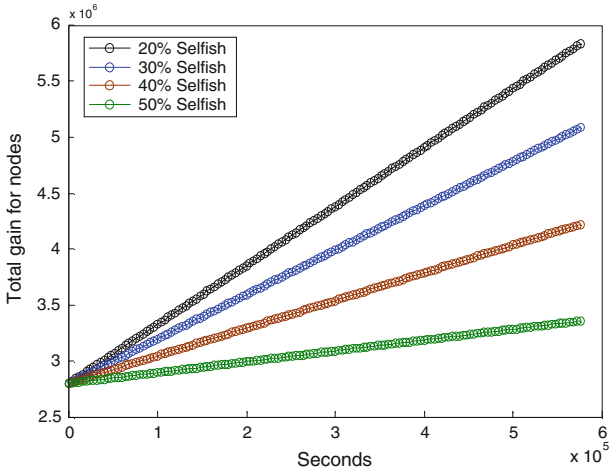


Fig. 11 Tit-for-Two-Tats

order to break the deadlock of the Tit-for-Tat strategy. Then, if the opponent defects twice consecutively, the Tit-for-Two-Tats bearer will defect in response.

Figure 11 depicts that the gain yielded by the Tit-for-Two-Tats strategy is close to the gain generated by the Generous Tit-for-Tat. Figure 12 shows the impact of Tit-for-Two-Tats on the gain of both cooperative and selfish nodes. It reveals that the strategy will end up with the selfish nodes having a higher gain than the cooperative nodes.

Therefore, we extended this approach by varying "k" to study the impact of increasing the number of "tats". We vary "k" from 2 to 10. The name of the strategy changes with the variation of the number of tats to be respectively: Tit-for-Four-Tats, Tit-for-Six-Tats, Tit-for-Eight-Tats, and Tit-for-Ten-Tats. Note that in the following we take the average gain for each strategy and we group them into tables because of space constraints. Table 4 reveals that the average gain of cooperative nodes in the different strategies is close somehow to the optimal

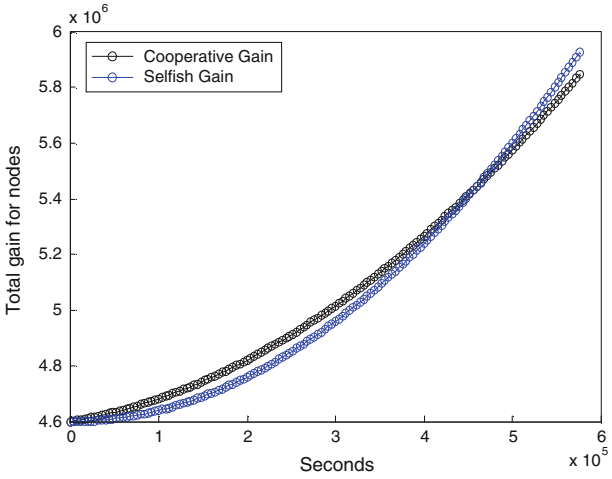


Fig. 12 Tit-for-Two-Tats with 20 % selfish nodes

Table 4 Average gain of cooperative nodes with different percentages of selfish nodes

K	Percentage of selfish nodes			
	20 %	30 %	40 %	50 %
2	1,969,000	1,537,100	1,105,100	673,100
4	2,098,900	1,666,700	1,234,700	802,700
6	2,185,300	1,753,100	1,321,100	889,100
8	1,969,000	1,537,100	1,105,100	673,100
10	2,746,700	2,314,700	1,882,700	1,450,700

Table 5 Comparison between the average gain of cooperative nodes and selfish nodes

K	Gain of cooperative nodes	Gain of selfish nodes
2	1,776,700	1,969,000
4	1,976,650	2,098,900
6	2,068,800	2,185,300
8	2,309,300	2,401,100
10	2,676,250	2,746,700

upper bound and that this gain increases as the number of tats "k" increases. That is, increasing the number of "tats" increases the generosity of the strategy. However, by looking at Table 5, we notice that the average gain of selfish nodes in the different strategies exceeds the average gain of cooperative nodes and that increasing the number of tats is able only to delay the time at which the gain of selfish nodes will exceed the gain of cooperative nodes but not to prevent it. This is justified by the fact that the selfish nodes will try repeatedly to cooperate in "k" requests (according to the number of tats used in the strategy) among the "n" requests in order to avoid being punished. Thus, by cooperating "k" times and saving resources (defecting) "k-n" times the gain of the selfish nodes will exceed the gain of cooperative nodes that cooperate and spend their resources "n" times.

7.6 Dempster–Shafer Based Tit-for-Tat

After being elected as MPRs, some nodes prefer to do not cooperate in the packets forwarding for selfish purposes. These nodes have dramatic implications on the network. Several approaches have been advanced in the literature to stimulate the cooperation of these nodes. The traditional version of Tit-for-Tat strategy is not able to deal with this problem since it will end up with a mutual deadlock where no vehicle will cooperate with any other one. The Generous Tit-for-Tat, in turn, which was proposed to prevent the deadlock caused by the traditional Tit-for-Tat, is still insufficient to solve the problem. In fact, the selfish nodes may exploit the generosity of this strategy to mimic the behavior of well-behaving nodes in order to avoid the punishments. This will give the selfish nodes a gain higher than the cooperative ones and will push the rational nodes to behave selfishly. The Tit-for-K-Tats is able to delay the time when the gain of selfish nodes exceeds the gain of cooperative nodes but not to prevent it. Moreover, all these strategies do not operate neither under high mobility nor under packets collisions. In order to overcome these limitations, we propose a Dempster–Shafer based Tit-for-Tat strategy, which is made up of five phases: reputation calculation, watchdogs monitoring, votes aggregation, Tit-for-Tat cooperation regulation, and information dissemination. The strategy can be summarized as follows. First, all the cluster nodes monitor the behavior of their MPR nodes and exchange their observations. Then, the cluster-head aggregates the collected observations using Dempster–Shafer theory and spreads the results. According to the Tit-for-Tat strategy, if the belief in trustworthiness of the MPR is > 0.5, the other nodes will cooperate with this MPR, which results in a gain for it. Otherwise, they will defect, which results in a loss for it. The general architecture of the Dempster–Shafer Tit-for-Tat strategy is depicted in Fig. 13.

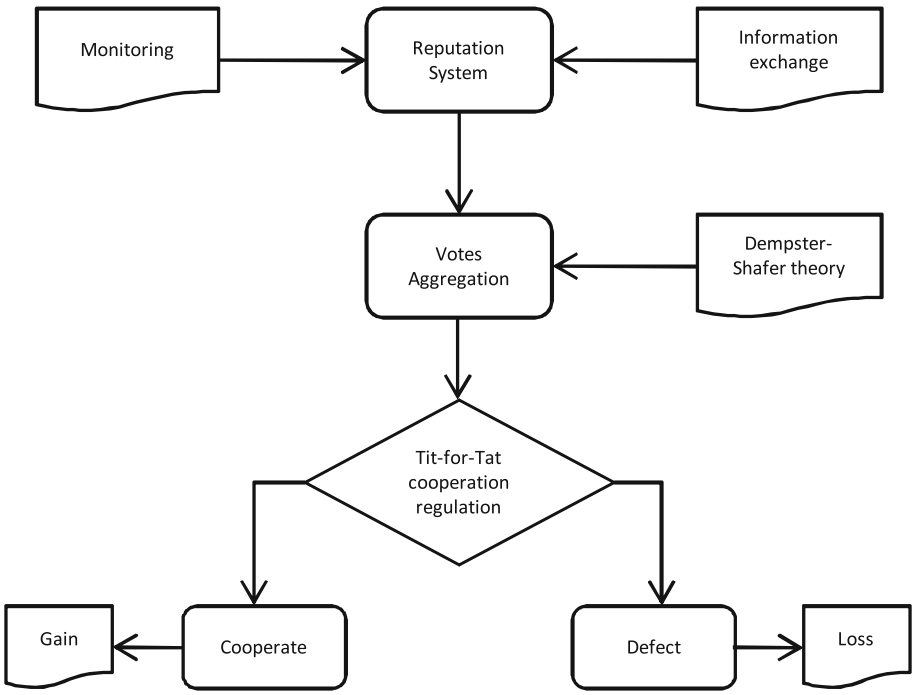


Fig. 13 Dempster–Shafer Tit-for-Tat model

7.6.1 Reputation Calculation

Based on the reward and punishment principle, each node is assigned a value called *reputation*. This value is set initially to 100 for all the nodes and is increased continuously whenever a node receives a payment from its voters. The payment is received by the nodes once elected as cluster-heads or MPRs. The payment of heads is expressed as the difference between the QoS value of the voted node (cluster-head) and the QoS value of the next best candidate among its neighbor nodes (the node having the next maximal local QoS value other than the head) so that:

- $R_{t+1}(x) = R_t(x) + P(j)$
  where:
- $P(j) = QoS(x) - \max\{QoS(k) | k \in N_1(j) \cup \{j\}\}$
- x is an elected Cluster-head node.
- $j \in N_1(x) \cup \{x\}$ represents all the 1-hop away nodes from x
- $R_t(x)$ expresses the reputation of node x at time t.
- $P(j)$ represents the payment advanced by node j in the network.

On the other hand, the MPR node that connects the 2-hop cluster heads should be paid by each of the two head node so that:

- $R_{t+1}(x) = R_t(x) + P(u) + P(w)$
  where:
- u is an elected cluster head.
- w is an elected cluster head.
- $CH_2(u)$ are the 2-hop away nodes from u.
- x is an elected MPR node for the nodes in $CH_2(k)$.
- $R_t(x)$ is the reputation of node x at time t.
- $P(u)$ be the payment offered by head node u.
- $P(u) = (QoS(x) - \max\{QoS(j) | j \in N_1(u) \bigcap N_1(w)\}$.
- $P(w) = (QoS(x) - \max\{QoS(j) | j \in N_1(u) \bigcap N_1(w)\}$.
- The path (u, x, w) maximizes $QoS(x)$ among all paths connecting u to w.

The payment value received by MPR nodes connecting 3-hop cluster head is established according the minimum QoS value of the new interconnecting path once the actual selected MPR node has been taken away. Thus:

- $R_{t+1}(x) = R_t(x) + P(k) + P(l)$.
- $R_{t+1}(y) = R_t(y) + P(k) + P(l)$.
  where:
- k is an elected cluster head.
- l is an elected cluster head.
- $CH_3(k)$ are the 3-hop away nodes from k.
- i is an elected MPR node for the nodes in $CH_3(k)$.
- $R(i)$ is the reputation of node i.
- $P(k)$ is the payment offered by head node k.
- The path (k, x_1, y_1, l) maximizes $min(QoS(x_1), QoS(y_1))$ among all paths connecting k to l.
- The path (k, x_2, y_2, l) maximizes $min(QoS(x_2), QoS(y_2))$ among all paths connecting k to l and $min(QoS(x_2), QoS(y_2)) < min(QoS(x_1), QoS(y_1))$.

The reputation accumulates over the time. Thus, we denote the reputation of a node x by: $R_{t+1}(x) = R_t(x) + P(x)$. Thus, the cooperative nodes will be continuously increasing their

reputation values. In contrary, if a selfish node decides to cooperate for only a short period, its reputation will gradually evaporate. Thereafter, the nodes are granted the network services proportionally to their reputation values. Thus, the access to the network resources for the selfish nodes will be limited. Note that dividing by the reputation values of the neighboring nodes ensures the fairness among the nodes to have the same chance of getting services. For example, if the available bandwidth in the network is 1,000 Mb/s and there are three neighbor nodes having reputation values of 109, 130, 116 respectively. The total reputation in the network is then $109 + 130 + 116 = 355$. Thus, the reputation ratios of the nodes are $\frac{109}{355}$, $\frac{130}{355}$, and $\frac{116}{355}$ respectively. The first node will get a bandwidth share of $\frac{109}{355} \times 1{,}000$. The share of the second node will be $\frac{130}{355} \times 1{,}000$ while the share of the third node will be $\frac{116}{355} \times 1{,}000$ with $\frac{109}{355} \times 1{,}000 + \frac{130}{355} \times 1{,}000 + \frac{116}{355} \times 1{,}000 = 1{,}000\,\text{Mb/s}$.

7.6.2 Watchdogs Monitoring

In this phase, the cluster-members are set as watchdogs to monitor the behavior of the MPR nodes forwarding packets on behalf of them. These watchdogs maintain a buffer of recently sent packets and received packets to see if there is a match. If so, this means that the MPR node has forwarded the packets. Otherwise, this MPR has potentially misbehaved and could be a selfish node. The algorithm of watchdogs monitoring is presented in Table 6. Thus, the watchdog will not consider this MPR selfish directly but will wait for the evidences from other observers to take a final decision. In fact, some out of control factors may affect the work of watchdogs namely the high mobility of vehicles and the collision problem depicted in the Sect. 5. Some vehicles may, for example, increase their speed to prevent the watchdogs from detecting whether they are transmitting the packets or not. Furthermore, it may happen, for instance, that some packets are not received within the expected time due to packets collisions. In these cases, the watchdogs may accuse innocent nodes to be misbehaving unjustly and vice versa. Moreover, some MPRs will cooperate with some nodes and defect with other nodes. Thus, these nodes are rewarded by some watchdogs and punished by others. In such a way, the selfish nodes will find a balance between cooperating and defecting in order to maximize their gain. Therefore, relying only on the opinion of individual watchdogs is able only to punish some nodes temporarily but not to regulate the cooperation. The aforementioned reasons raise the need for a cooperative detection that aggregates evidences from independent observers to come up with a final decision. Therefore, a votes' aggregation phase is presented in what follows (Table 7).

7.6.3 Votes Aggregation

In this phase, the observations from the different watchdogs are aggregated to form up a final unified decision. This can be done by launching a local voting process among the watchdogs situating in the same cluster. Nonetheless, the aggregation technique should take into account that some nodes may be intentionally or unintentionally untrustworthy. Namely, in addition to the deception caused by the collisions, some watchdogs may be selfish themselves and give false information to satisfy some egoist objectives. In fact, the voter watchdog may say that a MPR is cooperative while it is not if a plot between these two nodes took place. Similarly, some other voters may accuse cooperative MPRs to be misbehaving unjustly with the intention of excluding them from being competitors in any future election procedure. Therefore, there must be a distinction between trustworthy and untrustworthy voters (Table 7).

To do so, we have chosen the Dempster–Shafer theory [33] of evidences to be used while aggregating the votes of the different watchdogs. Dempster–Shafer is a mathematical model

Table 6 Watchdogs monitoring phase

```
   Let M be an elected MPR node.
   Let w be a neighbor watchdog for M.
   Let E_t be the expiry time to forward a packet.
   Let t be the current time.
   Let s be the packet source node.
   Let d be the packet destination node.
   Let p be the packet to send.

   For each watchdog w
1    Set an expiry time E_t for forwarding packet p
2    if t = E_t then
3       if p=s=d then
4         w marks M as "cooperative"
5       else
6         w marks M as "suspicious"
7       end
8    end
   End For
```

Table 7 Votes aggregation

```
   Let C be a cluster.
   Let H be the cluster-head of the cluster C.
   Let M be an MPR being judged.
   Let W be the set of watchdogs in C.
   Let H_x(i) be the opinion of node i in node x.
   Let belief(x) be the belief in trustworthiness in node x.

   For each watchdog w in W
1    If H_M(w)=cooperative
2         Set the voting flag to 1
3    else
4         Set the voting flag to 0
5    End if
   End For

   For each cluster-head H ∈ C
6    Calculate belief(M)
7    If belief(M) > 0.5 then
8         declare M as cooperative
9    else
10        declare M as selfish
11   End if
   End For
```

that is characterized by considering the uncertain evidences and by its ability to aggregate the evidences from independent sources. The motivations behind using Dempster–Shafer can be summarized as follows:

- The usefulness of Dempster–Shafer in representing and combining different types of evidences coming from independent sources.
- The fact that Dempster–Shafer represents uncertain evidences, which makes it appealing to model the ambiguity in the detection caused by the high mobility of vehicles and the channel collisions.

– The good reputation of Dempster–Shafer in many critical fields like investigating crimes and diseases.

The proposed algorithm works as follows. Initially, each node L is assigned a trustworthiness probability α according to its reputation value.

$$\gamma(N) = \frac{Rep(N)}{\sum_{j=1}^{n} Rep(N)} \tag{3}$$

where $Rep(N)$ is the reputation of node N and n represents all the neighbor nodes belonging to the same cluster as N. Note that dividing by the reputation values of the neighboring nodes ensures the fairness among the nodes. Let's define a power set Ω composed of three main elements: hypothesis $H = Cooperative$ stating that L is cooperative; hypothesis $H = Selfish$ that it is selfish; and hypothesis $U = \Omega$ that L is either cooperative or selfish. The latter hypothesis represents the uncertainty in the decisions when some watchdogs suffer from an ambiguity in the detection. The basic probability assignment (bpa), denoted by m, defines a mapping of the power set to the interval between 0 and 1, where the bpa of the null set is 0 and the summation of the bpas of all the subsets of the power set is 1. The value of the bpa for a certain set B is equal to the trustworthiness probability of the node giving the judgment. In other words, if vehicle X, which is trustworthy with probability γ, claims that vehicle Y is cooperative, then the primary probability assignments of node X are:

– $m_1(H) = \gamma(X)$
– $m_1(\bar{H}) = 0$
– $m_1(U) = 1 - \gamma(X)$

On the other hand, if the vehicle X states that Y is misbehaving, then the bpa of node X will be:

– $m_1(H) = 0$
– $m_1(\bar{H}) = \gamma(X)$
– $m_1(U) = 1 - \gamma(X)$.

To aggregate the different evidences, we calculate the following *belief in trustworthiness* function:

$$bel(H) = \sum_{j:A_j \subset H} m(A_j) \tag{4}$$

where H represents a hypothesis. The above function may be resolved by combining each pair of beliefs. This can be done as follows [8]:

$$m_1(H) \oplus m_2(H) = \frac{1}{K}[m_1(H)m_2(H) + m_1(H)m_2(U) + m_1(U)m_2(H)]$$
$$m_1(\bar{H}) \oplus m_2(\bar{H}) = \frac{1}{K}[m_1(\bar{H})m_2(\bar{H}) + m_1(\bar{H})m_2(U) + m_1(U)m_2(\bar{H})]$$

where:

$$K = \sum_{B \cap C = \varnothing} m_1(B)m_2(C) \tag{5}$$

We give in the following an example of how the aggregation is done between three watchdogs. Assume that there are two watchdogs A, and B judging a MPR x. The watchdog A has a reputation ratio of 0.9 whereas watchdog B has a reputation ratio of 0.2. A says that x is cooperative whereas B claims that it is not. The beliefs are then represented as follows:

Table 8 Dempster combination of watchdog A and watchdog B

W_B	W_A		
	Cooperative = 0	Selfish = 0.2	Uncertain = 0.8
Cooperative = 0.9	$m_A(C)m_B(C) = 0$	$m_A(S)m_B(C) = 0.18$	$m_A(U)m_B(C) = 0.72$
Selfish = 0	$m_A(C)m_B(S) = 0$	$m_A(S)m_B(S) = 0$	$m_A(U)m_B(S) = 0$
Uncertain = 0.1	$m_A(C)m_B(U) = 0$	$m_A(S)m_B(U) = 0.02$	$m_A(U)m_B(U) = 0.08$

Table 9 Dempster combination of watchdogs A, B, and C

W_C	W_{AB}		
	Cooperative = 0.878	Selfish = 0.024	Uncertain = 0.097
Cooperative = 0	$m_{AB}(C)m_C(C) = 0$	$m_{AB}(S)m_C(C) = 0$	$m_{AB}(U)m_C(C) = 0$
Selfish = 0.2	$m_{AB}(C)m_C(S) = 0.1756$	$m_{AB}(S)m_C(S) = 0.0048$	$m_{AB}(U)m_C(S) = 0.0194$
Uncertain = 0.8	$m_{AB}(C)m_C(U) = 0.7024$	$m_{AB}(S)m_C(U) = 0.0192$	$m_{AB}(U)m_C(U) = 0.0776$

– **Watchdog A:**

$m_A(C) = 0.9$ (Vehicle 1 is cooperative)
$m_A(U) = 0.1$ (watchdog 1 is uncertain)
$m_A(S) = 0$ (M is selfish)

– **Watchdog B:**

- $m_B(C) = 0$ (Vehicle 1 is cooperative)
- $m_B(U) = 0.2$ (Vehicle 1 is selfish)
- $m_B(S) = 0.8$ (watchdog 2 is uncertain)

– **Watchdog C:**

$m_C(C) = 0$ (Vehicle 1 is cooperative)
$m_C(U) = 0.2$ (Vehicle 1 is selfish)
$m_C(S) = 0.8$ (watchdog 2 is uncertain)

The combination of the beliefs with the two watchdogs is summarized in Table 8.

- $K = m_A(C)m_B(C) + m_A(C)m_B(U) + m_A(U)m_B(C) + m_A(S)m_B(S) + m_A(S)m_B(U) + m_A(U)m_B(S) + m_A(U)m_B(U) = 0 + 0 + 0.72 + 0 + 0.02 + 0.08 + 0 = 0.82.$
- $m_A(C) \oplus m_B(C) = 1/K[m_A(C)m_B(C) + m_A(C)m_B(U) + m_A(U)m_B(C)] = 1/0.82[0 + 0 + 0.72] = 0.72/0.82 = 0.878.$
- $m_A(S) \oplus m_B(S) = 1/K[m_A(S)m_B(S) + m_A(S)m_B(U) + m_A(U)m_B(S)] = 1/0.82[0 + 0.02 + 0] = 0.02/0.82 = 0.024.$
- $m_A(U) \oplus m_B(U) = 1/K[m_A(U)m_B(U)] = 1/0.82[0.08] = 0.08/0.82 = 0.097.$

Then, we combine the combined observations of watchdogs A and B with the observations of watchdog C (Table 9).

- $K = m_{AB}(C)m_C(C) + m_{AB}(C)m_C(U) + m_{AB}(U)m_C(C) + m_{AB}(S)m_C(S) + m_{AB}(S)m_C(U) + m_{AB}(U)m_C(S) + m_{AB}(U)m_C(U) = 0.8376.$

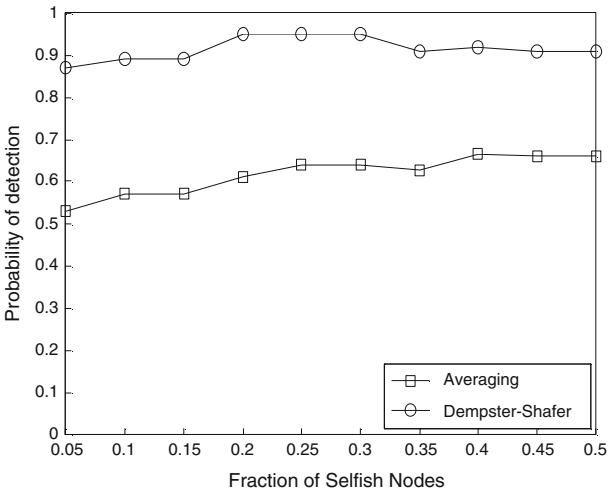


Fig. 14 Probability of detection

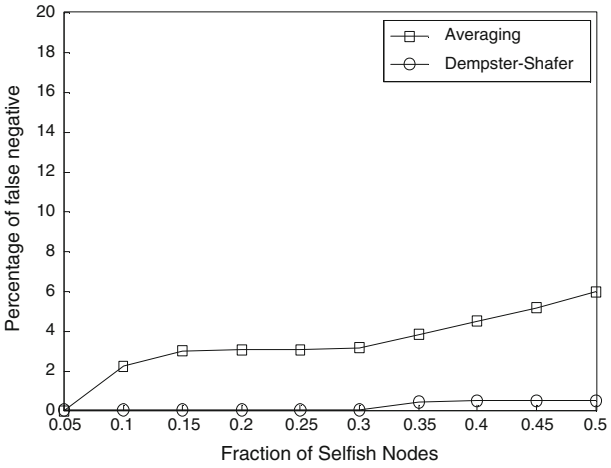


Fig. 15 Percentage of false negatives

- $m_{AB}(C) \oplus m_C(C) = 1/K[m_{AB}(C)m_C(C) + m_{AB}(C)m_C(U) + m_{AB}(U)m_C(C)] = 0.7024/0.8376 = 0.838$.
- $m_{AB}(S) \oplus m_C(S) = 1/K[m_{AB}(S)m_C(S) + m_{AB}(S)m_C(U) + m_{AB}(U)m_C(S)] = 0.0434/0.8376 = 0.052$.
- $m_{AB}(U) \oplus m_C(U) = 1/K[m_{AB}(U)m_C(U)] = 0.0776/0.8376 = 0.093$.

The basic probability assignment for the trustworthiness of MPR x is still high (0.838) although the majority of nodes, which are less trustworthy, reported that x is selfish. Therefore, the use of Dempster–Shafer is able to increase the probability of detection of selfish nodes and decrease the false alarms as shown in Figs. 14 and 15 In the following, simulations are conducted to compare two models: the Dempster–Shafer model and the averaging model.

The probability of detection is computed according to the following formula: *Detection probability = number of selfish nodes detected/number of existing selfish nodes*. It is used

Table 10 Cooperation regulation phase

```
Let i be an elected MPR node.
Let j be an elected MPR node.
Let R be a forwarding request from j to i.

1   if belief(j) > 0.5
2     i fulfills R
3   else
4     i drops R
5   End if
```

to evaluate the efficiency of any proposed detection mechanism. Figure 14 shows that the Dempster–Shafer can increase this probability up to 20 %. This is justified by the fact that Dempster–Shafer disregards the untrustworthy evidences upon building the final decisions. Moreover, Dempster–Shafer gives a weight for each evidence according to the trustworthiness level of the node giving this evidence. To do so, we propose payment mechanism to build a reputation for each node. Then, we use this reputation as the basic trustworthiness level for the nodes. This helps on giving accurate estimates on the trust level of each node and enhancing hence the probability of detection.

False negative occurs when an actual attack cannot be detected. Figure 15 reveals that using Dempster–Shafer for votes aggregation is able to minimize considerably the percentage of false negatives. This is due to the fact that Dempster–Shafer discounts the untrusted evidences upon building the final decisions and prevents them from beating the trusted evidences even they constitute the majority. Moreover, using the vehicles' reputation built through payment mechanisms to give weights for the collected evidences gives a realistic assessment on the behavior of the vehicles, which ensures the faithful application of the aggregation mechanism.

7.6.4 Tit-for-Tat Cooperation Regulation

In this phase, the cooperation among the nodes is decided according to the aggregated decision; that is, a MPR node i will cooperate with another MPR node j if the belief in trustworthiness of node j is greater than 0.5. Otherwise, it will defect (Table 10).

7.6.5 Information Dissemination

The information dissemination phase can be summarized as follows: after aggregating the final decision, each cluster-head broadcasts the results of the voting to its cluster members. Moreover, it has to propagate the detection results to the other cluster-heads when it gets connected with them. These cluster-heads, in their turn, disseminate these results to their cluster members, which will refrain from cooperating with the propagated misbehaving nodes without initiating new monitoring and voting phases. This process allows decreasing the overhead and reducing the implementation time of the strategy (Table 11).

Figure 16, illustrates the progress of the model implementation time as the percentage of selfish nodes increases in both cases "Without information dissemination" and "With information dissemination". It is obvious that the information dissemination is able to reduce the implementation time of the model up to 0.3 s.

This idea allows also reducing the overhead caused by the exchange of a large number of messages. In fact, applying the proposed strategy requires broadcasting messages to propagate the initial observations of the watchdogs, voting messages to announce the watchdogs'

Table 11 Information dissemination

Let's define:
CH_1: cluster head of cluster C_1.
CH_2: cluster head of cluster C_2.
S: selfish node detected in cluster C_2.
$Selfish(CH_1)$: set of selfish nodes detected within C_1.
$Selfish(CH_2)$: set of selfish nodes detected within C_2.
M_{12}: MPR node connecting CH_1 to CH_2.
$Path(x, y, z)$: path connecting cluster-heads x and z through MPR y.

1 $Selfish(CH_2) = S$
2 if $Path(CH_1, M_{12}, CH_2)$ then
3 $Selfish(CH_2) = Selfish(CH_2) \cup Selfish(CH_1)$
4 $Selfish(CH_1) = Selfish(CH_1) \cup Selfish(CH_2)$
5 end if

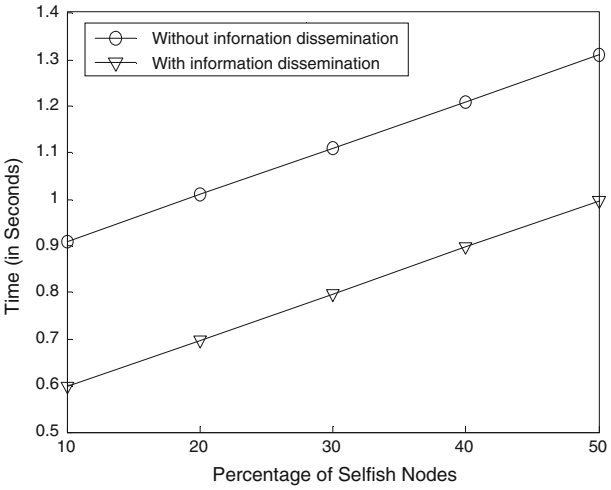


Fig. 16 Model implementation time

opinions, other messages for the cluster-head to propagate the decision to all its cluster members, and other bunch of messages for the cluster-head to warn the other cluster-heads whenever a contact between them occurs. We assume that all these messages are broadcasted 2-hop away. Thus the total overhead of the model is $N_i + N_i + N_i + N_i = 4N_i$ where N_i represents the number of 2-hop away neighbor nodes. By adopting the information dissemination concept, the propagation of watchdogs' initial observations and votes phases are eliminated which reduces the overhead to be $N_i + N_i = 2N_i$ with $2N_i < 4N_i$.

Figure 17 shows that the DS-based Tit-for-Tat strategy is as good as the generous strategy for the cooperative vehicles in the sense that their gain is close to the optimal upper bounds. This can be justified by the fact that these nodes will not be punished due to the high detection probability of the real selfish nodes and the null percentage of false alarms resulting from the use of Dempster–Shafer. Furthermore, Fig. 18 demonstrates that the DS-based strategy is able to regulate the cooperation inside the network by rewarding the cooperative nodes and punishing the selfish nodes. In the figure, we notice that the gain of cooperative nodes keeps

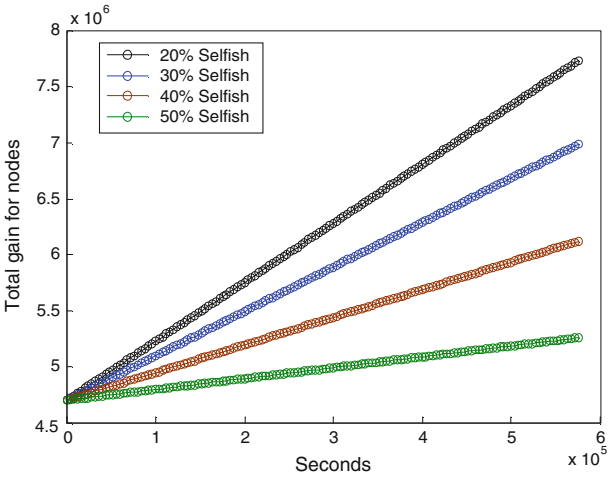


Fig. 17 Dempster–Shafer based Tit-for-Tat

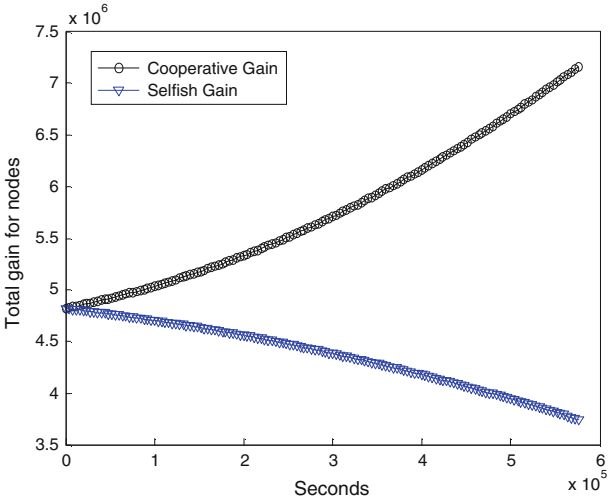


Fig. 18 Dempster–Shafer based Tit-for-Tat with 20 % selfish nodes

increasing along the time due to the rewards received by the different nodes while the gain of selfish nodes keeps decreasing along the time due to the continuous punishments imposed by not only single nodes, but by the different network nodes instead.

The figure shows that the gain of cooperative nodes is around 5×10^6 after 1,500 min of simulations, while the loss of selfish nodes is around -2×10^6. We can notice that the penalizing cost is large, which raises the need for a cooperative model that improves the detection and avoids the malicious use of our model. In the literature, the work related to game theory in VANET or MANET was all related to one to one monitoring relation and thus the above problems were not considered. In our model, a novel cooperative monitoring is proposed based on Dempster–Shafer theory to increase the credibility of the decisions and ensure hence the fairness of both punishments and rewards.

8 Conclusion

In this paper, we modeled the cooperation in the packets forwarding in VANET as a non-cooperative repeated game. We extended then this game model to consider the ambiguity caused by packets collisions and high mobility. We present then three strategies based on Tit-for-Tat to deal with the problem of selfish nodes. The strategies are: (1) generous Tit-for-Tat, (2) Tit-for-k-Tats, (3) and Dempster–Shafer based Tit-for-Tat. Simulation results reveal that the Dempster–Shafer based Tit-for-Tat strategy outperforms the strategies proposed in the literature since it uses a cooperative mechanism to build the decisions instead of relying on the one-to-one decision. The Dempster–Shafer based Tit-for-Tat is composed of five phases: reputation calculation, watchdogs monitoring, votes aggregation, Tit-for-Tat cooperation regulation, and information dissemination. Empirical results show that the proposed model is able to break the deadlock issue of the traditional Tit-for-Tat as well as to regulate the cooperation inside the vehicular network by rewarding the cooperative nodes and punishing the selfish nodes. They show also that our model increases the probability of detection of selfish nodes in a considerable manner and reduces the false negatives to a negligible percentage while maintaining a minimized implementation time and overhead.

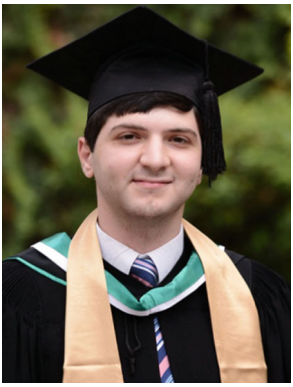
Acknowledgments This work was supported by National Council for Scientific Research (CNRS)—Lebanon, Lebanese American University (LAU), and Khalifa University of Science, Technology & Research (KUSTAR).

References

1. Anderegg, L., & Eidenbenz, S. (2003). Ad hoc-VCG: A truthful and cost-efficient routing protocol for Mobile ad hoc networks with selfish agents. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 245–259.
2. Anjum, F., & Mouchtaris, P. (2007). *Security for wireless ad hoc networks*. Hoboken: Wiley.
3. Axelrod, R. (1981). The Emergence of Cooperation among Egoists. *The American Political Science Review*, *75*, 306–318.
4. Axelrod, R. (1984). *The evolution of cooperation*. New York: Basic.
5. Badis, H., & Al Agha, K. (2005). QOLSR, QoS routing for ad hoc wireless networks using OLSR. *European Transactions on Telecommunications*, *16*, 427–442.
6. Balakrishnan, K., Deng, J., & Varshney, P.-K. (2005). TWOACK: Preventing selfishness in Mobile ad hoc networks. In *Proceedings of IEEE wireless communications and networking conference (WCNC'05)*, (Vol. 4, pp. 2137–2142).
7. Buchegger, S., & Le Boudec, J.-Y. (2002). Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, (pp. 226–236).
8. Chen, T.-M., Venkataramanan, V. (2005). Dempster-shafer theory for intrusion detection in ad hoc networks. *IEEE Internet Computing, 9*, 35–41.
9. Cover, T.-M., & Thomas, J.-A. (2006). *Elements of information theory* (2nd ed.). Hoboken: Wiley.
10. Douceur, J.-R., & Moscibroda, T. (2007). Lottery trees: Motivational deployment of networked systems. In *Proceedings of the 2007 conference on applications, technologies, architectures, and protocols for, computer communications*, (pp. 121–132).
11. Felegyhazi, M., Hubaux, J. P., & Buttyan, L. (2006). Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *IEEE Transactions on Mobile Computing, 5*, 463–476.
12. Fiore, M., Harri, J., Filali, F., & Bonnet, C. (2007). Vehicular Mobility Simulation for VANETs. In *Simulation symposium, annual*, (pp. 301–309).
13. Gilat, A. (2008). *MATLAB: An introduction with applications*. Hoboken: Wiley.
14. Harri, J., Filali, F., Bonnet, C., & Fiore, M. (2006). VanetMobiSim: Generating realistic mobility patterns for VANETs. In *Proceedings of the 3rd international workshop on vehicular, ad hoc networks*, (pp. 96–97).
15. Jain, S., Fall, K., & Patra, R. (2004). Routing in a delay tolerant network. *SIGCOMM Computer Communications Review*, *34*, 145–158.

16. Jawandhiya, P., Ghonge, M., Ali, M.-S., & Deshpande, J.-S. (2010). A survey of mobile ad hoc network attacks. *International Journal of Engineering Science and Technology*, *2*, 4063–4071.
17. Johnson, D.-B., & Maltz, D.-A. (1996). Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, *353*, 153–181.
18. Jones, E., Li, L., & Ward, P. (2007). Practical routing in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, *6*, 943–959.
19. Lee, S., Pan, G., Park, J., Gerla, M., & Lu, S. (2007). Secure incentives for commercial ad dissemination in vehicular networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, (pp. 150–159).
20. Li, F., & Wu, J. (2009). FRAME: An innovative incentive scheme in vehicular networks. In *Proceedings of the 2009 IEEE international conference on communications*, (pp. 4638–4643).
21. Lian, Q., Peng, Y., Yang, M., Zhang, Z., Dai, Y., & Li, X. (2008). Robust incentives via multi-level Tit-for-Tat: research articles. *Concurrency and Computation: Practice and Experience*, *20*, 167–178.
22. Liu, K., Deng, J., Varshney, K., & Balakrishnan, K. (2007). An acknowledgment-based approach for the detection of routing misbehavior in MANETs. *IEEE Transactions on Mobile Computing*, *6*, 536–550.
23. Marasigan, D., & Rommel, P. (2005). MV routing and capacitybuilding in disruption tolerant networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, (Vol. 1, pp. 398–408).
24. Marti, S., Giuli, T.-J., Lai, K., & Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, (pp. 255–265).
25. Michiardi, P., Molva, R. (2002). CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the IFIP TC6/TC11 sixth joint working conference on communications and multimedia security*.
26. Milan, F. (2006). Achieving cooperation in multihop wireless networks of selfish nodes. In *Workshop on game theory for networks* (GameNets 2006).
27. Milan, F., Jaramillo, & J., Srikant, R. (2006). Performance analysis of Reputation-based mechanisms for multi-hop wireless networks. In *Proceedings of 40TH conference on information sciences and systems (CISS 2006)*, (pp. 12–17).
28. Nowak, M.-A., & Sigmund, K. (1992). Tit for tat in heterogeneous populations. *Nature*, *355*, 250–253.
29. Nowak, M.-A., & Sigmundl, K. (1994). The alternating prisoner's dilemma. *Journal of Theoretical Biology*, *168*, 219–226.
30. Nzouonta, J., Rajgure, N., Wang, G., Borcea, C. (2009). VANET routing on city roads using real-time vehicular traffic information. *IEEE Transactions on Vehicular Technology, 58*, 3609–3626.
31. Otrok, H., Mourad, A., Robert, J.-M., Moati, N., & Sanadiki, H. (2011). A cluster-based model for QoS-OLSR protocol. In *IWCMC*, (pp. 1099–1104).
32. Ramakrishnan, B. (2012). Performance analysis of AODV routing protocol in Vehicular ad-hoc network service discovery architecture. *ARPN Journal of Systems and Software, 96*, 65–72.
33. Shafer, G. (1976). *A mathematical theory of evidence*. Princeton, NJ, USA: Princeton University Press.
34. Spyropoulos, T., Psounis, K., & Raghavendra, C.-S. (2005). Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, (pp. 252–259).
35. Turocy, T.-L., & Stengel, B. (2001). *Game theory*. Technical report.
36. Vahdat, A., & Becker, D. (2000). Epidemic routing for partially-connected ad hoc networks. Technical report.
37. Wu, J., & Axelrod, R. (1995). How to cope with noise in the iterated prisoner's dilemma. *Journal of Conflict Resolution*, *39*, 183–189.
38. Zhao, J., & Guohong, G. (2008). VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. In *IEEE Transactions on Vehicular Technology*, (Vol. 57, pp. 1910–1922).
39. Zhong, S., Yang, Y., & Chen, J. (2003). Sprite: A simple, cheat-proof, credit-based system for Mobile ad hoc networks. In *Proceedings of INFOCOM*, (pp. 1987–1997).

Author Biographies



Omar Abdel Wahab is a M.Sc. student in computer science at the Lebanese American University (LAU). He holds a bachelor degree in computer science from the Lebanese University. The topics of his research activities are computer security, network security and vehicular ad hoc networks.

Hadi Otrok holds an assistant professor position in the department of computer engineering at Khalifa University. He received his Ph.D. in Electrical and Computer Engineering (ECE) from Concordia University, Montreal, Canada. His research interests are mainly on network and computer security. Also, he has interest on resources management in virtual private networks and wireless networks. His Ph.D. thesis was on "Intrusion Detection System (IDS)" using Game Theory and Mechanism Design. Throughout his Masters degree, he worked on "Security Testing and Evaluation of Cryptographic Algorithms". Before joining Khalifa University, Dr. Otrok was holding a postdoctoral position at the École de technologie supérieure (University of Quebec). He is serving as a technical program committee member for different international conferences and regular reviewer for different specialized journals.

Azzam Mourad is an assistant professor in the department of computer science and mathematics at the Lebanese American University (LAU). He holds Ph.D. degree in electrical and computer engineering from Concordia University, Canada and M.Sc. degree in computer science from Laval University, Canada. The main topics of his current research activities are web services security, web services engineering, aspect-oriented programming, ad-hoc network security, information security, software security hardening and security engineering. Dr. Mourad is currently serving as technical program committee member of several international conferences and reviewers for different international journals. In the past, he served as Postdoctoral fellow at Concordia University.