A Cooperative Watchdog Model Based on Dempster-Shafer for Detecting Misbehaving Vehicles

Omar Abdel Wahab, Hadi Otrok[‡], Azzam Mourad[☆]
Department of Mathematics & Computer science,
Lebanese American University, Beirut, Lebanon
[‡]Department of Electrical & Computer Engineering,
Khalifa University of Science, Technology & Research, Abu Dhabi, UAE
Email:{omar.abdelwahab, azzam.mourad}@lau.edu.lb, Hadi.Otrok@kustar.ac.ae

Abstract

In this paper, we address the problem of detecting misbehaving vehicles in Vehicular Ad Hoc Network (VANET) using Quality of Service Optimized Link State Routing (QoS-OLSR) protocol. According to this protocol, vehicles might misbehave either during the clusters' formation by claiming bogus information or after clusters are formed. A vehicle is considered as selfish or misbehaving once it over-speeds the maximum speed limit or under-speeds the minimum speed limit where such a behavior will lead to a disconnected network. As a solution, we propose a two-phase model that is able to motivate nodes to behave cooperatively during clusters' formation and detect misbehaving nodes after clusters are formed. Incentives are given in the form of reputation and linked to network's services to motivate vehicles to behave cooperatively during the first phase. Misbehaving vehicles can still benefit from network's services by behaving normally during the clusters' formation and misbehave after clusters are formed. To detect misbehaving vehicles, cooperative watchdog model based on Dempster-Shafer is modeled where evidences are aggregated and cooperative decision is made. Simulation results show that the proposed detection model is able to increase the probability of detection, decrease the false negatives, and reduce the percentage of selfish nodes in the vehicular network, while maintaining the Quality of Service and stability.

Keywords: Vehicular Ad hoc Network (VANET), Dempster-Shafer, Cooperative Detection, Reputation, Passive Malicious Nodes, Selfish Vehicles.

1. Introduction

Vehicular Ad Hoc Network (VANET) [16], [20], [19], [9] is a new kind of ad hoc networks that is characterized by its highly mobile topology. Like Mobile Ad hoc

[☆]Corresponding Author Tel:+961 (1) 786456 ext. 1200; Fax:+961 (1) 867 098

Network (MANET), VANET encounters the problem of selfish nodes that may hinder the implementation of any protocol dedicated to it. However, dealing with these nodes in VANET is more challenging due to the increased ambiguity in the detection caused by the high mobility of vehicles. The Quality of Service Optimized Link State Routing (QoS-OLSR) protocol [10] 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 *MultiPoint Relay* (MPR) nodes. This protocol is an enhanced version of QOLSR [1] that prolongs the network lifetime by considering the energy of nodes while calculating the QoS function since the nodes, in MANET, have limited energy resources. However, the energy parameter has a minimal importance in VANET due the long battery lifetime of vehicles. In order to extend such a protocol to VANET, velocity and residual distance parameters must be added to the QoS function instead of the residual energy to improve the network stability.

According to this protocol, vehicles might misbehave either during the clusters' formation by claiming bogus information or after clusters are formed. A vehicle is considered as selfish or misbehaving once it over-speeds the maximum road limit or under-speeds the minimum road limit. Such a behaviour is considered as a *passive malicious* since vehicles do not aim to attack or impede the network functioning, but rather they tend to optimize their own gain neglecting the welfare of others [11]. They entail hence negative implications on the whole network such as the (1) increase in the percentage of MPRs, (2) decrease in the network stability, (3) increase in the clusters disconnections, and (3) increase in the average path length.

To address the above problems, we propose a two-phase model that (1) motivates vehicles to behave normally during clusters' formation and (2) detects misbehaving vehicles after clusters' formation. In phase one, incentives are given in the form of reputation where networks' services are offered based on vehicle's accumulative reputation. Misbehaving vehicles can still benefit from networks' services by behaving normally during the clusters' formation and misbehave after clusters are formed. Thus, the main challenge that we are addressing in this paper and as phase two of our model is the detection of misbehaving vehicles after clusters formation. This is done by the means of cooperative watchdog model based on Dempster-Shafer theory [4] where evidences are correlated cooperatively in order to improve the probability of detection and reduce the false alarms. Thus, we overcome the problem of ambiguity in the detection resulting from packets collision, high mobility of vehicles, and untrustworthy watchdogs. The cluster-members, including the cluster heads, are designated as watchdogs to monitor the behavior of their MPRs where the evidences of any suspicious MPR are shared among all. To overcome the problem of initial trust estimates that the Dempster-Shafer suffers from, we use the reputation calculated in phase one for this purpose. In summary, our contribution is a cooperative detection model based on Dempster-Shafer that is able to increase the probability of detection and reduce the false alarms.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Section 3 formulates the problem. Section 4 motivates the work. Section 5 explains the proposed approach in details. Section 6 explains the model used for simulation and presents empirical results. Finally, Section 7 concludes the paper.

2. Related Work

In the literature, several approaches have been proposed to motivate the cooperation in mobile ad hoc networks. These approaches can be classified into two categories: credit-based mechanisms [6], [13], [14], [21] and reputation-based mechanisms [3], [15], [17], [18]. In the credit-based approaches, the nodes receive incentives in terms of virtual currency versus their contributions in the network functions. In the reputation-based approaches, a monitoring process occurs to detect the misbehaving nodes. The detection results are then broadcasted all over the network in order to prevent the misbehaving nodes from being utilized in all the future routes [2]. In what follows, the main contributions in both credit-based and reputation-based approaches are discussed.

2.1. Credit-based approaches

The *receipt counting method* [13] was proposed by Lee et al. to control the commercial ad dissemination in VANETs. According to this method, the source of the packet undertakes a fixed value for each receipt. The shortcoming of this method is that the source does not know the number of network nodes in advance and is not able hence to predict the total amount of payments. This entrains an overspending problem for the source nodes.

Douceur et al. [6] resorted to the use of a lottery tree mechanism called *lottree*. This method is based on selecting periodically one node in the network to be the receiver of the payment. This selection is achieved in a way that guarantees to encourage high participation and to stimulate new participants. However, the lottery schemes suffer from the fact that only one winner will be selected to obtain the whole payment. This would discourage conservative nodes from participating regarding their poor chances to win.

FRAME [14] consists of two phases: Weighted rewarding component and Sweepstake component. The weighted rewarding component assigns weighted rewards for each vehicle according to its contribution. The sweepstake component grants the winner participating vehicle a fixed payment amount. However, this strategy encourages the sender nodes to avoid the intermediate nodes and get connected straight to the destination so as to gain more contribution weight.

In gross, the basic idea of the credit-based schemes is that nodes pay virtual money to get served and get paid to serve. Nonetheless, the lack of scalability, centralization, and the need for a tamper-proof hardware are the limitations that may encounter these schemes.

2.2. Reputation-based Approaches

Tit-for-Tat [15] associates the incentive mechanisms with the reputation concept so that cooperating with more reputable nodes enables the nodes from increasing their own reputation and benefiting hence from a larger set of services. However, this strategy encounters three main problems. First, the decision of cooperation is restricted to the local relation between each pair of nodes. Second, it neglects 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. [17] included the watchdog and pathrater concepts into the Dynamic Source Routing (DSR) [12] protocol. Their approach is based on preventing the detected misbehaving from forwarding packets instead of punishing them. However, according to this scheme the misbehaving nodes are remunerated vis-a-vis their behavior as their packets continue to be transmitted by others while they do not have to transmit and spend resources.

CORE [18] is a collaborative reputation mechanism that employs the watchdog concept. It defines three types of reputation: functional reputation (task specific behavior), subjective reputation (observations), and indirect reputation (positive reports by other nodes). A weight is assigned to each type of reputation to build an aggregated reputation used to judge a node. The weakness of CORE is that it considers only positive indirect reputation to avoid false accusation and denial of service attacks.

CONFIDANT [3] sends an alarm to the network nodes upon detecting a misbehaving node. This aims to isolate the misbehaving nodes from the network. Nonetheless, the credibility of the received alarms is not guaranteed.

Overall, in the reputation-based mechanisms, 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. However, these approaches have several disadvantages that may limit their efficiency such as: ambiguous collision, limited transmission power, false alarms, and non-cooperative monitoring.

3. Problem Statement

This paper tackles the problem of selfish or misbehaving MPR vehicles that misbehave by over-speeding the maximum road limit or under-speeding the road limit. To motivate the addressed problem, simulations related to such a behavior are done to show the impact on the network. This is done by modifying the speed of some vehicles accordingly and varying the percentage of these vehicles from 0% to 50%. For example, if the speed limits on a highway are set to be within 80 km/h and 120 km/h, then the average speed on this highway will be $\approx$ 100 km/h. According to this example, a vehicle is considered as misbehaving if it over-speeds/under-speeds by at at least 40 km/h compared to the average speed limit. The percentage of misbehaving vehicles used in the simulations ranges from 0% without selfish nodes and increase gradually to 50% of the total nodes. (The selection of this interval is explained in Section 6).

In Fig. 1 (a), as much as the MPRs over-speed/under-speed the other vehicles, they would be disconnected from their clusters. This will make the network disconnected and raises hence the need of electing new MPR nodes to re-connect the clusters, which justifies the increase in the percentage of MPRs as the percentage of selfish nodes increases. In Fig. 1 (b), the increase in the number of selfish/misbehaving vehicles will deteriorate the stability of the network gradually due to their speed compared to normal ones. For 0% selfish nodes, the percentage of stability keeps increasing as long as the number of nodes increases. This is because the network becomes more dense, the nodes closer to each other and connected by more MPRs (e.g. 100 nodes in an area of $3000 \times 1000m$ will be more connected than 30 nodes). In contrary, for the other percentages of selfish nodes (from 10% to 50%), as much as the number of nodes

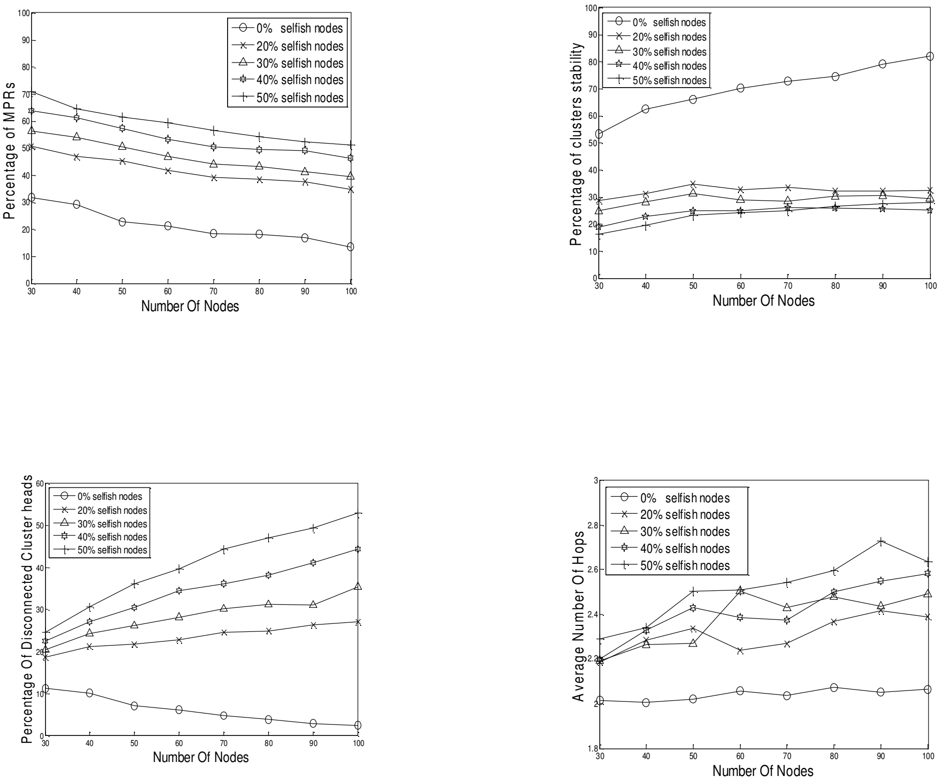

Figure 1: Impact of the selfish nodes on the (a) percentage of MPRs (b) percentage of stability (c) percentage of clusters disconnections (d) number of hops

increases, the percentage of stability is quite the same and will not increase. This is because some MPR nodes (according to the percentage of selfish nodes) connecting the nodes are over-speeding/under-speeding and not serving hence as relaying nodes, which will inhibit the increase in the connectivity and the stability of network. In Fig. 1 (c), the percentage of disconnected cluster-heads increases in conjunction with the increase of the percentage of selfish vehicles since the MPRs connecting these cluster-heads are over-speeding/under-speeding and leaving their clusters, which makes these cluster-heads disconnected from each others. In Fig. 1 (d), the End-to-End delay increases considerably as the percentage of selfish MPRs increases. This is because the selected routing paths will be broken in a short period of time since the MPRs, that are forming these paths, are misbehaving and causing link failures, which results in a delay

Table 1: Quality of Service Metrics

Notations and Quality of Service Metric Function
Let *i* be a node in the network. Let's define:
QoS(i) = Quality of Service Metric of node i
BW(i) = Available bandwidth of **i**
N(i) = Neighbors of *i*
VelRatio(i) = Ratio of velocity for *i*
DistRatio(i) = Ratio of remaining distance for *i*
Bandwidth Model
1 QoS(i) = $BW(i)$;
Proportional Bandwidth
2 QoS(i) = $\frac{BW(i)}{N(i)}$;
Proportional Bandwidth & Velocity Model (Prop. B-V)
3 QoS(i) = $\frac{BW(i)}{N(i)} \times VelRatio(i)$;
Proportional Bandwidth & Proportional Distance Model (Prop. B-DV)
4 QoS(i) = $\frac{BW(i)}{N(i)} \times \frac{DistRatio(i)}{VelRatio(i)}$;
Bandwidth-Connectivity & Proportional Distance Model (BCDV)
5 QoS(i) = $BW(i) \times N(i) \times \frac{DistRatio(i)}{VelRatio(i)}$;

in the packets' delivery.

If we generalize these facts on the whole network, the situation will be catastrophic. For that reason, it is indispensable to find a model that is able to deal with these nodes after clusters are formed. This raises the need for a detection model that can detect any misbehaving vehicle.

4. Clusters Formation

In this section, we present the Quality of Service (QoS) models used during clusters formation. Then, we show an illustrative example explaining how the QoS-OLSR [10] clustering algorithm work.

4.1. Quality of Service metrics Models

To ensure electing/selecting heads/MPRs having a good level of stability and Quality of Service, we propose several QoS models (Table 1). These models take into consideration the following metrics: bandwidth, connectivity, velocity, and residual distance. The bandwidth is considered to ensure the reliability, the connectivity is considered to increase the coverage of elected/selected cluster-heads/MPRs, while the velocity and distance parameters are considered to maintain the stability of the network. The residual distance represents the number of meters to reach the destination and it has two objectives: (1) group the vehicles into clusters with convergent residual distance, and (2) ensure to elect heads and MPRs with considerable distance to traverse.

The residual distance parameter in the deployed systems can be obtained with the help of the Global Positioning System (GPS) that can save the most visited places for each vehicle. As an example, most of the vehicles on the road are for employees that are targeting their work or returning to home where the destination is known and thus the estimated residual distance can be computed. Similarly, adding the velocity parameter has two objectives: (1) group the vehicles into clusters with convergent velocity scale, and (2) ensure to elect heads and MPRs with reasonable velocity. The first objective contributes in prolonging the lifetime of the clusters, while the second reduces the link failures.

It is worth to note that the proposed QoS models guarantee the fairness among vehicles during elections/selections. In fact, we are proposing several QoS functions according to different set of combinations. These functions are not restricted to the location-based parameters (i.e., residual distance and velocity) or the performance-based parameters (i.e., bandwidth) but involve a tradeoff between several aspects. Practically, these functions are composed of various metrics related to the reliability (bandwidth), connectivity (number of neighbors), and stability (velocity, and residual distance). The fairness is ensured since all the nodes can participate the election/selection processes and benefit hence from the available bandwidth link (Section 5). As an example, let's consider that a MPR selection process occurs according to the Bandwidth-Connectivity & Proportional Distance Model (BCDV) QoS model, which gives the best results (Section 6). In this process, there are two competing vehicles having the following Quality of Service parameters values:

- Vehicle 1: bandwidth= 130, connectivity=3, distance ratio=0.3, velocity ratio=0.4. $QoS(1) = 130 \times 3 \times \frac{0.3}{0.4} = 292.5$.
- Vehicle 2: bandwidth= 150, connectivity=4, distance ratio=0.4, velocity ratio=0.4. $QoS(2) = 150 \times 4 \times \frac{0.2}{0.4} = 300$.

Thus, even Vehicle 1 has more residual distance, Vehicle 2 will have a higher QoS value and have hence more chances to be elected/selected as heads/MPRs since the QoS function is designed in way that guarantees the fairness among nodes by considering a tradeoff between several factors. Consequently, a node will not be excluded from the election/selection process if it suffers from a weakness in a certain parameter.

4.2. Clusters formation example

Table 2: QoS metrics values of nodes using the BCDV model

Nodes	1	2	3	4	5	6	12
QoS value	685.8	197	503.2	379.4	316.7	338.7	746.5
Nodes	7	8	9	10	11	13	14
QoS value	308.1	400	234.01	159.54	389.5	797.8	708.76

In this example, Bandwidth-Connectivity & Proportional Distance Model (BCDV) (Table 1) is used since it gives the best results in terms of number of MPRs, network

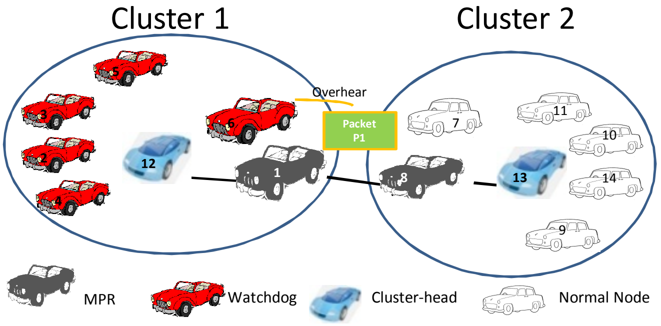

Figure 2: Vehicular ad-hoc Network example: A network of 14 nodes is used to illustrate how the algorithms of payment, reputation calculation, detection, and aggregation work

stability, end-to-end delay, and packet delivery ratio as shown in Section 6, where Table 2 shows the QoS values calculated for each node. In addition, Fig. 2 depicts a vehicular network composed of 14 nodes that need to form clusters by electing cluster-heads, and connect the clusters by selecting MPRs. The QoS-OLSR [10] clustering algorithm, which includes cluster-heads election and MPRs selection, works as follows. First, nodes broadcast *HELLO* messages containing their Quality of Service (QoS) values to their one hop neighbors. Then, each node votes for its neighbor having the local maximal QoS metric value to be the cluster-head. A node can as well vote for itself, if it has the maximal local QoS value. In our example, vehicles 12 and 13 are elected as cluster-heads since they have the local maximal QoS values among their 1-hop neighbors (746.5, and 797.8 respectively). Now, clusters are formed and nodes will join their elected cluster head. Thus, a cluster is formed by the nodes 1, 2, 3, 4, 5, 6, and 12 (Cluster 1), and another cluster is formed by the nodes 7, 8, 9, 10, 11, 13, and 14 (Cluster 2). Once the cluster-heads are elected, they are responsible for selecting a set of MPR nodes allowing them to connect and communicate with each other. The MPRs selection algorithm can be summarized as follows:

- The one-hop away cluster-heads are directly connected without the need for MPRs.
- For the 2-hop away cluster-heads, one MPR node is needed. Thus, the node having the highest QoS value and connecting the two cluster-heads will be selected as MPR.
- For the 3-hop away cluster-heads, two MPR nodes are needed. Thus, the nodes belonging to the path having the highest QoS value and connecting the two cluster-heads will be selected as MPRs. In this case, the local cluster-head could not select both MPR nodes since its *HELLO* messages cannot inform the 2-hop node that has been selected. Thus, one of the MPR nodes would be selected by the local cluster head and the other one have to be selected by the 3-hop away cluster-head.

In our example, suppose that the cluster-head 12 is willing to connect with cluster-head 13, which is 3-hop away, it has four choices: $\{6,7\}$, $\{6,8\}$, $\{1,7\}$, $\{1,8\}$ having respective QoS values of 646.8, 738.7, 966.6, and 1085.8. Thus, since the path $\{1,8\}$ has the maximal QoS value, the cluster-head 12 selects node 1 as MPR. Similarly, the cluster-head node 13 selects node 8 as MPR. Now, these two cluster-heads can communicate with each other through the path $1-8$, which presents the highest QoS value. Concerning the fairness of the MPRs selection mechanism, we are dealing with a dynamic topology in which the nodes are continuously moving, which makes their locations perpetually changeable. Moreover, we are considering a connected network topology, which allows the different nodes to be connecting two or more clusters at a time. However, the problem with this clustering protocol arises when the vehicles, which are assumed to be rational, refuse to cooperate in the clustering model either by broadcasting bogus information during clusters formation in order to be avoided from being selected or by misbehaving after getting an increase in the reputation. Such behavior will degrade the performance of the network and will lead to a disconnected network as shown in Section 3, which will break the objective of any clustering algorithm. Therefore, the issue of the cooperation in vehicular ad hoc networks is critical and is as important as the clustering algorithm itself.

5. The Two-Phase: Incentive and Detection Model

In this section, we describe the *VANET-DSD* model proposed to motivate and detect the selfish nodes in VANET. The model is composed of two phases: motivation phase and detection phase. The solution can be summarized as follows. Once elected, each cluster-head and MPR receives a payment from its voters. This payment is used to build a reputation for each node. Then, each node benefits from the network services according to its reputation value and the reputation is considered during elections to ensure electing a set of trusted heads and MPRs. After elections, some nodes are selected as watchdogs to monitor the behavior of the MPR nodes. These nodes decide according to their observations. Afterwards, the observations are shared among all nodes located in the same cluster so that each node aggregates all the observations using Dempster-Shafer to make the final decision.

5.1. Reputation Design

In this part, we design a reputation model that has two objectives: (1) motivating the truth-telling of vehicles during clusters formation, and (2) overcoming the problem of initial trust estimates in Dempster-Shafer. To achieve the first objective, the reputation is designed in a way to encourage the nodes being elected/selected as heads/MPRs so as to increase their share of network services. Concerning the second objective, the Dempster-Shafer theory used to aggregate the evidences in the detection model suffers from a serious problem, which is determining the initial trust estimates of the vehicles. These estimates may affect the results of the aggregation and therefore they have to be set in a thoughtful way. Thus, we use the reputation of each vehicle, which is the result of an accumulated payment model, to be its initial trust estimate.

The reputation value is set initially to 100 for all the nodes and is increased continuously whenever a node receives a payment from its voters/selectors. The payment is

Table 3: Quality of Service Metrics

Notations and Quality of Service Metric Function
Let i be a node in the network. Let's define:
QoS(i) = Quality of Service Metric of node i
BW(i) = Available bandwidth of **i**
N(i) = Neighbors of i
VelRatio(i) = Ratio of velocity for i
DistRatio(i) = Ratio of remaining distance for i
R(i) = Reputation of i
Bandwidth Model
6 QoS(i) $= BW(i) + \mathbf{R}(\mathbf{i})/\sum\mathbf{R}(\mathbf{N}(\mathbf{i}));$
Proportional Bandwidth
7 QoS(i) = $\frac{BW(i)}{N(i)} + \mathbf{R}(\mathbf{i})/\sum\mathbf{R}(\mathbf{N}(\mathbf{i}));$
Proportional Bandwidth & Velocity Model (Prop. B-V)
8 QoS(i) = $\frac{BW(i)}{N(i)} \times VelRatio(i) + \mathbf{R}(\mathbf{i})/\sum\mathbf{R}(\mathbf{N}(\mathbf{i}));$
Proportional Bandwidth & Proportional Distance Model (Prop. B-DV)
9 QoS(i) = $\frac{BW(i)}{N(i)} \times \frac{DistRatio(i)}{VelRatio(i)} + \mathbf{R}(\mathbf{i})/\sum\mathbf{R}(\mathbf{N}(\mathbf{i}));$
Bandwidth-Connectivity & Proportional Distance Model (BCDV)
10 QoS(i) $= BW(i) \times N(i) \times \frac{DistRatio(i)}{VelRatio(i)} + \mathbf{R}(\mathbf{i})/\sum\mathbf{R}(\mathbf{N}(\mathbf{i}));$

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). The payment of cluster-heads is explained in Algorithm 1.

On the other hand, the MPR node that connects the 2-hop away cluster heads should be paid by each of the two head nodes according to Algorithm 2.

The payment received by the MPR nodes connecting 3-hop away cluster heads is established according the minimum QoS value of the new interconnecting path once the actual selected MPR node has been taken away. The payment of these MPRs is explained in Algorithm 3.

The reputation value of a node represents the cumulative payment received by this node. The reputation accumulates over the time. Thus, we denote the reputation of a node x by: $R_{t+1}(x) = R_t(x) + P(x)$. In such a way, 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. Moreover, the vehicles benefit from the network services according to their reputation values. Thus, the access to the network resources for the selfish nodes will be restricted. For example, if the available bandwidth in the network is 2000Mb/s and there are four neighbor nodes having reputation values of 123, 115,108, and 154 respectively. The total reputation in the network is then 123 + 115 + 108 + 154 = 500. Thus, the reputation ratios of the nodes are $\frac{123}{500}$, $\frac{115}{500}$, $\frac{108}{500}$, and $\frac{154}{500}$ respectively. The first node yields a bandwidth

Algorithm 1: Cluster-heads Payment Algorithm

```
Initialization:
Let $x$ be an elected cluster-head node.
Let $R_t(x)$ be the reputation of node $x$ at time $t$.
Let $P(j)$ represent the payment offered by node $j$.
Let $N_1(x)$ represent the two-hop away nodes from $x$.

procedure HEADPAYMENT
    for each $j \in N_1(x) \cup \{x\}$ do
        $P(j) = QoS(x) - \max\{QoS(k) | k \in N_1(j) \cup \{j\}\}$
        $R_{t+1}(x) = R_t(x) + P(j)$
    end for
end procedure
```

Algorithm 2: Payment Algorithm for MPRs Connecting 2-hop Clusters

```
Initialization:
Let $CH_2(u)$ be the 2-hop away nodes from $u$.
Let $x$ be an elected MPR node for the nodes in $CH_2(k)$.
Let $u$ be an elected cluster head.
Let $w$ be an elected cluster head.
Let $R_t(x)$ be the reputation of node $x$ at time $t$.
Let $P(u)$ be the payment offered by head node $u$.
Let $N_1(x)$ represent the one-hop away nodes from $x$.

procedure TWOHOPMPRPAYMENT
    The path $(u,x,w)$ maximizes $QoS(x)$ among all paths  connecting $u$ to $w$.
    $P(u) = QoS(x) - \max\{QoS(j) | j \in N_1(u) \bigcap N_1(w)\}$.
    $P(w) = QoS(x) - \max\{QoS(j) | j \in N_1(u) \bigcap N_1(w)\}$.
    $R_{t+1}(x) = R_t(x) + P(u) + P(w)$
end procedure
```

Algorithm 3: Payment Algorithm for MPRs Connecting 3-hop Clusters

```
Initialization:
Let $CH_3(k)$ be the 3-hop away nodes from $k$.
Let $x$ and $y$ be elected MPR nodes for the nodes in $CH_3(k)$.
Let $k$ be an elected cluster head.
Let $l$ be an elected cluster head.
Let $R_t(x)$ be the reputation of node $x$ at time $t$.
Let $P(k)$ be the payment offered by the head node $k$.

procedure THREEHOPMPRPAYMENT
    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 $min(QoS(x_2),QoS(y_2)) < min(QoS(x_1),QoS(y_1))$.
    $R_{t+1}(x) = R_t(x) + P(k) + P(l)$.
    $R_{t+1}(y) = R_t(y) + P(k) + P(l)$.
end procedure
```

share of $\frac{123}{500} \times 2000$. The bandwidth share of the second node will be $\frac{115}{500} \times 2000$. The bandwidth share of the third node is $\frac{108}{500} \times 2000$, while the share of the fourth node will be $\frac{154}{500} \times 2000$ knowing that $\frac{123}{500} \times 2000 + \frac{115}{500} \times 2000 + \frac{108}{500} \times 2000 + \frac{154}{500} \times 2000 =$ 2000Mb/s. Thus, each node tends to increase its reputation value in order to increase its share of network resources. In such a way, we guarantee that the nodes will reveal their true QoS values during elections in order to get elected and rewarded.

In order to elect the trusted set of heads and MPRs, the *reputation* of each node is added to the QoS function. Thus, the QoS models become as shown in Table 3. Note that we divide the reputation value of each node by the sum of reputations of its neighbor nodes to ensure the fairness and to increase the competitiveness among nodes during elections.

5.2. Detection Mechanism

After being selected as MPRs, some nodes may behave selfishly by refusing to cooperate in the networking functions such as packet forwarding. These nodes seek to over-speed/under-speed the other nodes in order to realize their own goals. Such behavior degrades the performance of the network dramatically as shown in Section 3. Therefore, we need a detection mechanism that is able to identify such nodes. Several detection mechanisms [15], [17], [3] are proposed in the literature to detect the selfish nodes. However, these mechanisms are non-cooperative which makes any decision to be unilateral and sometimes untrustworthy. Moreover, these mechanisms suffer from the problems of ambiguity and false alarms caused by packets collisions and high mobility. We propose, in this section, a nested cooperative detection mechanism composed of four algorithms: monitoring, sharing, aggregation, and contact dissemination. The mechanism can be summarized as follows. First, the cluster-members, including the cluster-head, are designated as watchdogs for their MPR nodes to collect evidences on the suspected ones. Thereafter, the evidences are shared among all the nodes. Then, each node aggregates the evidences using Dempster-Shafer theory to construct the final decision. Finally, the cluster-heads exchange the decisions with each other to reduce the detection time and overhead.

Monitoring : This algorithm aims to identity the suspicious nodes. It is derived from the *watchdog* concept [17] where the members in each cluster, including the cluster-head, are appointed as watchdogs to monitor the behavior of the MPR nodes and ensure that they are cooperating well. These nodes can overhear the communications between nodes locating in their transmission range. Thus, if a node W can overhear the incoming and outgoing transmissions from/to a MPR M, then W may be designated as a watchdog to monitor M's behavior. To do so, each watchdog node specifies an expected time for each packet to be sent. After the expiry of this time, the watchdog, that maintains a buffer of recently sent and packets, will compare each overheard packet with the packet in the buffer to see if there is a match. If so, then the packet was delivered correctly and the watchdog will mark the sender MPR as "good". Otherwise, it will not mark this MPR as "selfish" automatically but it will accuse it to be "suspicious" awaiting the observations from the other watchdogs to make the final decision.

However, some out of control factors may affect the work of watchdogs. It may happen, for instance, that some packets are not received within the expected time due to network collisions or high mobility. In this case, the watchdogs may accuse cooperative nodes to be misbehaving unjustly. The opinion of only one or few watchdog nodes is thus not sufficient. Here lies the importance of launching a cooperative detection and sharing the observations among vehicles.

Sharing : In this algorithm, each node shares its evidences with the other nodes locating in its clusters so that they can aggregate all the gathered evidences and come up with an aggregated final decision.

Aggregation : In order to build a final decision, the nodes have to do an aggregation function. they can merely calculate the average of the received evidences or even follow the simple majority-decision rule. However, the aggregation function should take into account that some untrustworthy evidences may affect the final decision. Namely, the watchdogs may say that the MPR is good while it is not if a plot between these two nodes took place. Similarly, some watchdogs may accuse good MPRs to be misbehaving unjustly with the intention of excluding them from being competitors in any future selection procedure. Therefore, there must be a distinction between trustworthy and untrustworthy evidences.

To do so, we propose an aggregation algorithm based on Dempster-Shafer theory [4]. This theory has proved its efficiency in such kind of problems where evidences from independent sources need to be combined to come up with an aggregated decision. Due to its effectiveness in this area, it has been widely used in many critical fields like investigating crimes and diseases. This theory suffers, however, from the problem of determining the initial estimates of the nodes' trustworthiness. We overcome this issue by using the vehicles' reputations calculated in the motivation mechanism for this purpose. The reputation gives an accurate estimation of the trust level of the vehicles since it is a result of cumulative payments offered to the truth-teller vehicles. The aggregation algorithm works as follows. Initially, each vehicle L is assigned a

Algorithm 4: Detection Algorithm - Cooperative Monitoring

```
Initialization:
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.

procedure COOPERATIVEMONITORING
    for each watchdog $w$ do
        Set an expiry time $E_t$ for forwarding packet $p$;
        if t:=:$E_t$ then
            if p:=s:=d then
                $w$ marks $M$ as “good”;
            else
                $w$ marks $M$ as “suspicious”;
            end if
        end if
    end for
end procedure
```

Algorithm 5: Detection Algorithm - Sharing

```
Initialization:
Let $C_i$ be the cluster members of cluster $C$.
Let $Evidences(S)$ be the set of evidences collected by vehicle $S$.

procedure SHARING
    for each vehicle $X \in C_i$ do
        for each vehicle $Y \in C_i$ and $Y \neq X$do
            $Evidences(X) := Evidences(X) \cup Evidences(Y)$
        end for
    end for
end procedure
```

trustworthiness probability α according to its reputation value so that:

$$\alpha(L) = \frac{Reputation(L)}{\sum_{j=1}^{n} Reputation(j)} \tag{1}$$

where n represents all the neighbor nodes belonging to the same cluster as L. Note that dividing by the reputation values of the neighboring nodes ensures the fairness and increases the competitiveness among the nodes to increase their reputations. Let's define a power set Ω composed of three main elements: hypothesis $H = C$ stating that an MPR M is cooperative; hypothesis $\bar{H} = S$ that it is selfish; and hypothesis $U = \Omega$ that M is either cooperative or selfish. This latter hypothesis is important to express the uncertainty in the decisions when some watchdogs are not sure if an MPR is cooperative or not. The probability of cooperation assigned to the node being judged is equal to the trustworthiness probability of the node giving the judgment. This means that if node X, which is trustworthy with probability α, states that node Y is cooperative, then the primary probability assignments of node X are:

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

In contrary, if the node X claims that Y is selfish, then the basic probability assignments of node X are:

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

The combination rule for the gathered evidences is expressed in terms of *belief in trustworthiness* function:

$$bel(H) = \sum_{j:A_j \subset H} m(A_j) \tag{2}$$

where H represents a hypothesis. The above function may be resolved by combining each pair of beliefs. This can be done as follows [5]:

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

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

Where:

$$K = \sum_{B \cap C = \varnothing} m_1(B)m_2(C) \tag{3}$$

Dempster-Shafer generates a judgment value between 0 and 1 expressing the degree of belief in that judgment. Thus, the use of Dempster-Shafer is important to exclude evidences from untrustworthy or uncertain observers upon building the final judgment by giving more weight to the trusted evidences to the detriment of the untrustworthy evidences.

Algorithm 6: Detection Algorithm - Votes Aggregation

```
Initialization:
Let $m_i$ be the set of nodes in the cluster $m$.
Let $X$ be the node taking the decision $m$.
Let $M$ be a MPR in $m$ in being judged
Let $belief(T)$ denotes the belief in trustworthiness of $M$.
Let $N_i$ be the number of nodes in $m$.
Let $D_i(j)$ be the decision of vehicle $i$ on vehicle $j$.

procedure VOTESAGGREGATION
    for each node $X$ do
        Calculate $belief(T) := \sum_{i=1}^{N_i} m_i(H)$.
        if $belief(T) \geq 0.5$ then
            $D_H(M) := cooperative$
        else
            $D_H(M) := selfish$
        end if
    end for
end procedure
```

Contact Dissemination : The overhead and time of the detection algorithm is somehow high. In fact, the nodes should perform 3 algorithms: monitoring, sharing, and aggregation. To overcome this issue, contact dissemination principle is used to make clusters share the belief in trustworthiness of the nodes. Thus, the selfish nodes will be punished by all of the vehicles (who share the belief of this node) without the need of launching the monitoring, sharing, and aggregation algorithms repeatedly. The contact dissemination phase works as follows. After building the aggregated decisions, the cluster-head has to broadcast these decisions to the other cluster-heads whenever a contact with them occurs. These cluster-heads, in turn, disseminate this information to all their cluster members. Thus, these nodes will no longer cooperate with the propagated selfish nodes if these latter fall later in their transmission range without launching a new monitoring, sharing and aggregation algorithms. Thus, instead of lunching a new detection process for a node marked already as selfish, the nodes can save their time and refrain from dealing these nodes directly thanks to the cooperative dissemination. This idea allows also reducing the detection overhead caused by the exchange of a large number of messages.

Algorithm 7: Detection Algorithm - Contact dissemination

```
Initialization:
Let $H_1$ be a cluster head of cluster $C1$.
Let $H_2$ be a cluster head of cluster $C2$.
Let $S$ be a selfish node in cluster $C1$.
Let $SelfishSet(H_1)$ be the set of selfish nodes detected within the cluster $C1$.
Let $SelfishSet(H_2)$ be the set of selfish nodes detected within the cluster $C2$.

procedure ContactDissemination
    $SelfishSet(H_1) := S$
    if new contact between $H_1$ and $H_2$ occurs then
        $SelfishSet(H_2) := SelfishSet(H_2) \cup SelfishSet(H_1)$
        $SelfishSet(H_1) := SelfishSet(H_1) \cup SelfishSet(H_2)$
    end if
end procedure
```

Table 4: Reputation values of nodes using Reputation value calculation algorithm

Cluster 1								
Nodes	1	2	3	4	5	6	12	*Total*
Initial Reputation	100	100	100	100	100	100	100	700
New Reputation	177.8	100	100	100	100	100	160.7	838.5
Trust α	0.21	0.12	0.12	0.12	0.12	0.12	0.19	1

Cluster 2								
Nodes	7	8	9	10	11	13	14	*Total*
Initial Reputation	100	100	100	100	100	100	100	700
New Reputation	100	161.3	100	100	100	189.04	100	850.34
Trust α	0.12	0.18	0.12	0.12	0.12	0.22	0.12	1

5.3. Illustrative Example

In this part, we continue the example presented in Section 4 to show how the payments are done, the reputations are calculated, and the cooperative detection is modeled. The initial reputation values of all the nodes shown in Fig. 2 are set to 100 as shown in Table 4. Nodes 12 and 13, which have the local maximal QoS values in their clusters, are elected as cluster-heads for clusters 1 and 2 respectively. After being elected as cluster-heads, nodes 12 and 13 receive a payment. The payment is calculated as follows. Node 12 will receive a payment value of Payment(12) = QoS(12) - QoS(1) = $746.5 - 685.8 = 60.7$ to yield a new reputation of Rep(12) = $100 + 60.7 = 160.7$. Similarly, the node 13 will receive a payment of Payment(13) = QoS(12) - QoS(1) = $797.8 - 708.76 = 89.04$ to yield a new reputation value of Rep(13) = $100 + 89.04 = 189.04$. Afterwards, a MPR selection algorithm takes place according to QoS-OLSR selection algorithm. Nodes 1 and 8 are selected as MPRs according to this algorithm. These MPRs receive also a payment from their voter nodes once selected. According to the example, the MPRs 1 and 8 connecting the 3-hop away cluster-heads 12 and 13 should be paid. We need to find the path connecting 12 and 13 and having the second best QoS. In this case, the path is 1-7 composed of nodes 1 and 7. The payment of the MPRs will be hence the QoS difference between the two path so that: Payment(1)=Payment(8)=min(QoS(1),QoS(8)) - min(QoS(1),QoS(7)) = 400-308.1=91.9. Thus, the new reputation value of node 1 becomes Rep(1) = 100+91.9 = 191.9. Similarly, the node 8 will get a reputation of Rep(8)= 100+91.9 = 191.9.

Now, the nodes 2, 3, 4, 5, 6, and 12 will serve as watchdogs to monitor the behavior of the MPR node 1. These nodes can overhear all the incoming/outcoming packets from/to node 1 since this latter falls in their transmission ranges. Suppose that the node 1 has to send a packet $p1$ to the node 8. The watchdog nodes estimate the expected time the packet should take in order to reach its destination, let's say *30 ms*. Then, after the expiry of this delay, the watchdogs check if the packet has been received to the potential destination using the buffer they maintain. If they find that the packet was received, they mark the node 1 as "good". Otherwise, they mark the node 1 as "suspicious". Suppose that watchdogs 3 and 6 reported that vehicle 1 is suspicious. Then, all the watchdogs share their observations to make the final decision on this MPR. They have now to aggregate the observations using Dempster-Shafer. We give, in the following, an example of how the aggregation is done between two watchdogs. Assume in our example that the first watchdog claims that vehicle 1 is selfish with a probability of 0.99 and that this watchdog is uncertain of its decision with probability of 0.01 (denoted by $m_1(S)$ and $m_1(U)$, respectively). The second watchdog states that 1 is cooperative with a probability of 0.99 and is uncertain of its decision with probability of 0.01 (denoted by $m_2(C)$ and $m2(U)$, respectively). The beliefs are then represented as follows:

- **Watchdog 1** :

  $m_1(S) = 0.99$ (Vehicle 1 is selfish)
  $m_1(U) = 0.01$ (watchdog 1 is uncertain)
  $m_1(C) = 0$ (M is cooperative)

- **Watchdog 2** :

  $m_2(C) = 0.99$ (Vehicle 1 is cooperative)
  $m_2(S) = 0.01$ (Vehicle 1 is selfish)
  $m_2(U) = 0$ (watchdog 2 is uncertain)

The combination of the beliefs with the two watchdogs is summarized in Table 5.
Using Equations 2 and 3:

- Multiplying the beliefs from intersected row and column yields the combined probability , e.g., $m_{12}(S) = (0.99)(0.01) = 0.0099$.
- The empty intersections represent a conflict.

Table 5: Dempster Combination of Watchdog 1 and Watchdog 2

W1 W2	Selfish=0.99	Cooperative=0	Uncertain=0.01
Selfish=0.01	$m_1(S)m_2(S) = 0.0099$	$m_1(C)m_2(S) = 0$	$m_1(U)m_2(S) = 0.0001$
Cooperative=0.99	$m1(S)m2(C) = 0.9801$	$m_1(C)m_2(C) = 0$	$m_1(U)m_2(C) = 0.0099$
Uncertain=0	$m_1(S)m_2(U) = 0$	$m_1(C)m_2(U) = 0$	$m_1(U)m_2(U) = 0$

- The single nonzero value is for the combination of $Selfish, m_{12}(S) = (0.99)(0.01) =$ 0.0099.

- To calculate K, we multiply the empty intersections that represent conflicts. Using Equation 3, K = (0.99)(0.99) + (0.01)(0.01) + (0.01)(0.99) = 0.9901.

- Using Equation 2, $m_1(S)m_2(S) = (0.99)(0.01)/[1 - 0.0099] = 1$.

The basic probability assignment for the selfishness of vehicle 1 turns out $Bel(S) = 1$ although there is many conflicting beliefs. The vehicle 1 is marked then as selfish. Now, the cluster head node 12 spreads this decision to the cluster-head node 13 whenever a contact between them occurs to may, in its turn, inform its cluster members (7,9,10,11,14) in order to accelerate the detection procedure. Thus, if the vehicle 1 gets the cluster scope of any of the Cluster 2 members, they will directly refrain from electing it or cooperating with it without the need of new monitoring and voting mechanisms.

6. Simulation Results

In this section, we explain in details the simulation scenario and parameters used to build our simulations. We present as well the simulation results yielded after comparing our proposed QoS models. We compare also the Dempster-Shafer aggregation model against averaging model. We call “With DS” the Dempster-Shafer model and “Without DS” the averaging model.

6.1. Simulation Scenario and Parameters

MATLAB [8] network simulator and VanetMobiSim [7] traffic simulator have been used to simulate the different models. VanetMobiSim is an XML-based traffic simulator that allows the user to define the vehicular network features such as number of nodes, topography, velocity, duration, and time steps. VanetMobiSim supports both micro-mobility and macro-mobility features. Macro mobility model cares of the macroscopic aspects that affect the vehicular traffic such as road topology, intersections, number of lanes, traffic light constraints, and speed limits. Micro mobility is concerned more by the driving behavior such as acceleration, deceleration, and behavior in presence of traffic signs [7]. A simulation area of 3000 × 1000m is used to simulate a set of nodes varying from 30 to 100. The screenshot of this area is presented in Fig. 3. The multi-lane highway topology is used to simulate the traffic. The minimum allowed speed on this highway was set to 60 km/h, while the maximum speed was 120 km/h. After the simulation has been completed, VanetMobisim generates a file containing some important features such as time, velocity, and position. We parse hence this file to use these parameters to simulate the vehicular network using MATLAB. The transmission ranges used for the simulations vary from 150 to 300. The simulation scenario is summarized in Table 6.

Table 6: Simulation Parameters

Parameter	Value
Aggregation Models	Averaging and Dempster-Shafer
Number of nodes	$30, 40, 50, 60, 70, 80$, and 100
Percentage of selfish nodes	$0\%, 20\%, 30\%, 40\%$, and 50%
Transmission range	300 m
Topology	Multi-lane highway
Packet Size	1 kb
Idle Time	Random value in $[0..1]$
Link Bandwidth	2Mbps
Available Bandwidth	$Idle\ Time \times Link\ Bandwidth$
Initial Reputation	100
Hello messages	18 messages are sent per minute
Minimum Speed	60 km/h
Maximum Speed	120 km/h

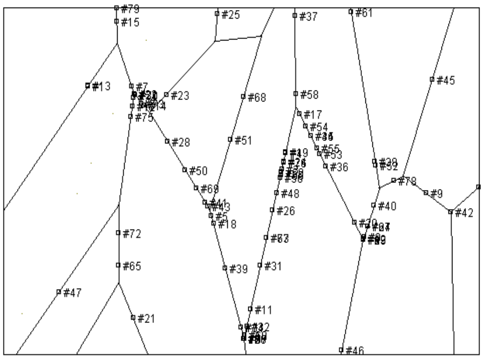

Figure 3: Screenshot of the vehicular movement simulation using VanetMobisim

The number of selfish nodes used to simulate the aggregation models vary from 10% to 50% of the total nodes. Within this interval, the impact of the selfish nodes will be catastrophic on the network as depicted in the section 3. For 0% of selfish nodes, there is no need for detection. Similarly, above 50% the misbehaving nodes form the majority and their negative impact begins to diminish gradually since they can form new clusters and resume the networking functions again.

6.2. Simulation Results

In this section, we compare first the proposed Quality of Service models (Table 1) in order to find the best set of combinations that is able to maintain the performance, stability, and trust. We show also the efficiency of the motivation mechanism in terms of percentage of selfish nodes. We present then a detailed comparison between the averaging aggregation model and Dempster-Shafer aggregation model. The first model computes the average of the different observations to judge a suspected node whereas the second model uses the Dempster-Shafer theory to aggregate the votes.

6.2.1. Comparison between QoS metrics models

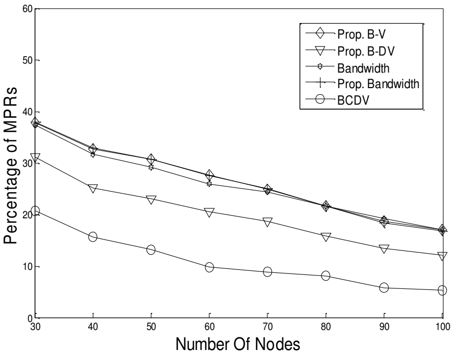

Figure 4: Percentage of MPRs: This aspect represents the percentage of selected MPR nodes. The decrease of this aspect decreases the overhead and jamming over the network

In this part, we present a comparison between the QoS metrics models presented in Table 1 in order to find the best set of combinations. Fig. 4 shows that the Bandwidth-Connectivity & Proportional Distance (BCDV) model is able to decrease the percentage of MPR nodes. This is due to the fact that the BCDV model multiplies the number of neighbors or the connectivity index by the other QoS metrics, while this index is divided by the other QoS metrics in the other models (refer to Table 1). Note that decreasing the percentage of MPRs is important to reduce the jamming and overhead in the network. Concerning the stability of the clusters, which relies fundamentally on the distance and velocity parameters, Fig. 5 reveals that the BCDV model shows an improved percentage of stability since this model multiplies the QoS function by the residual distance and divides it by the vehicle's velocity. Fig. 6 shows that BCDV is able to reduce the end-to-end delay by decreasing the average number of hops between sources and destinations. Fig. 7 compares the packet delivery ratio factor yielded by the different QoS models. This factor represents the total number of packets received by the destination over the total number of packets sent by the source. According to Fig. 7, the BCDV model increases this ratio compared to the other models. This is

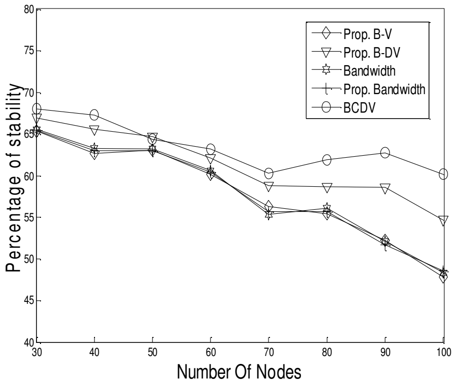

Figure 5: Percentage of Stability: This aspect is used to measure the clusters lifetime and evaluate the efficiency of considering the high mobility parameters

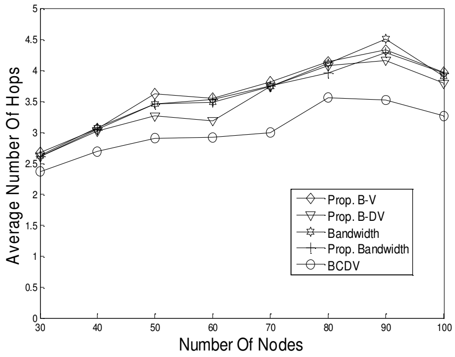

Figure 6: Average Number of Hops: This aspect is used to study the End-to-End delay

because BCDV increases the connectivity, maintains the stability, and reduces the End-to-End delay. Moving to the percentage of selfish nodes in the network, Fig. 8 shows that all the models give almost the same percentage since all these models consider the reputation of the nodes in their QoS functions. Overall, the Bandwidth-Connectivity & Proportional Distance (BCDV) model is preferred to enhance the network performance, Quality of Service, overhead, stability, and trust.

To study the impact of the proposed motivation mechanism, we compare in Fig. 9 the BCDV model in two scenarios: (1) without motivation: means without the motivation mechanism, and (2) with motivation: after applying the motivation mechanism. The figure reveals that adding the reputation is able to reduce the percentage of selfish nodes in the network up to 40%.

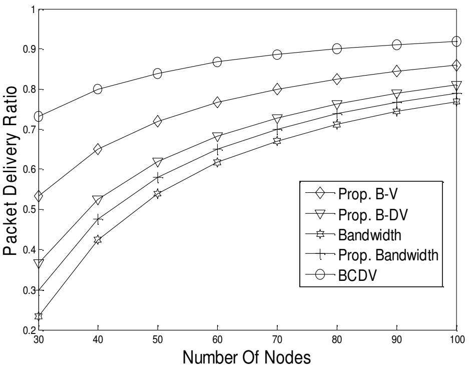

Figure 7: Packet Delivery Ratio: This aspect measures the level of delivered data to the destination

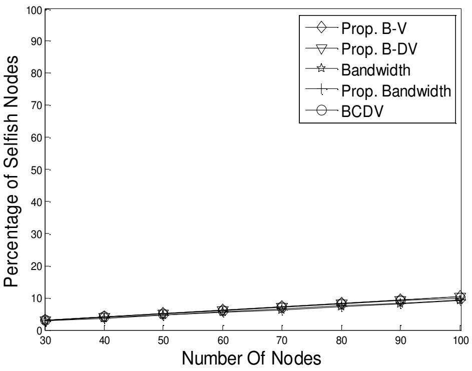

Figure 8: Percentage of Selfish Vehicles: This aspect reflects the percentage of selfish nodes in the network. This figure is used to study the best set of metrics that is able to reduce this percentage

6.2.2. Probability of detection

The probability of detection is obtained by dividing the number of detected selfish nodes by the real number of selfish nodes. This aspect measures the efficiency of the detection model. As depicted in the Fig. 10, using Dempster-Shafer as an aggregation model increases the probability of detection up to 20%. This result is expected since Dempster-Shafer discounts the untrustworthy and uncertain votes upon building the final judgement which augments the accuracy of the decisions. By discarding the untrustworthy and uncertain votes, the Dempster-Shafer model is increasing the number of detected selfish nodes and is able hence to increase the probability of detection.

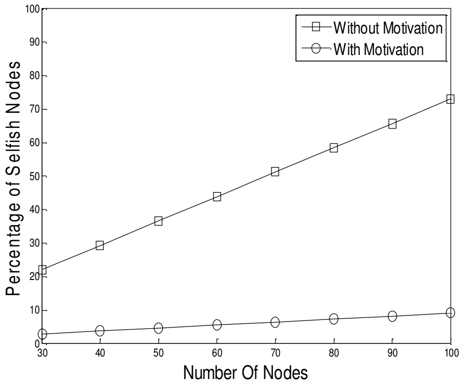

Figure 9: Percentage of Selfish Vehicles: This aspect reflects the percentage of selfish nodes in the network. This figure is used to study the impact of adding the reputation to the QoS metrics function on this percentage

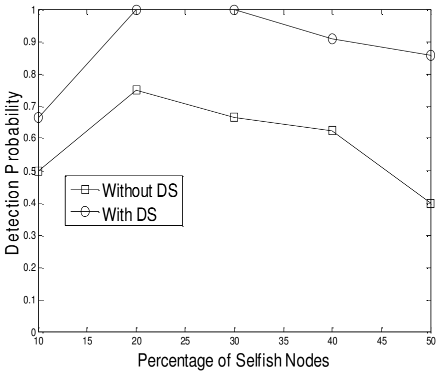

Figure 10: Probability of detection: This aspect reflects the number of detected selfish nodes out of the real number of selfish nodes. It is used to study the impact of using the cooperative detection and the Dempster-Shafer on the detection efficiency

6.2.3. False Negatives

False negative represents a failure to detect an actual attack. This value is increased whenever an existing attack is not detected. As shown in Fig. 11, the "Without Dempster-Shafer" model allows some breaches to occur in this context. In fact, this model allows the selfish node to build some alliances with some watchdog nodes to gain their votes and acquit themselves. In contrary, the Dempster-Shafer model gives a zero percentage of false negatives. This is due to the fact that the reputation value

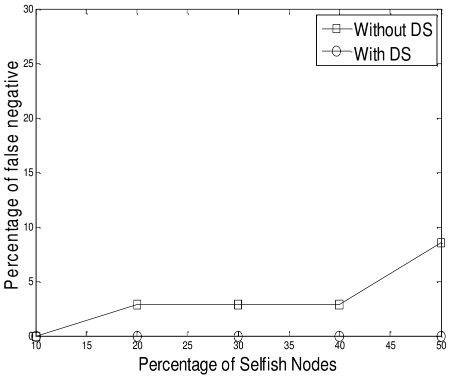

Figure 11: Percentage of false negatives: This aspect reflects the failure to detect an actual attack. It is used to study the impact of using the cooperative detection and the Dempster-Shafer on the false alarms

built through a payment mechanism affects the weight of each vote. This value gives an accurate assessment of the nodes' trust level since it is a result of an accumulated payment model. This leads to prevent the inaccurate votes from beating the accurate ones. Thus, even the majority of the nodes reported the false decision, the weighting remains for the trustworthy observations. This ensures that all the misbehavior actions will be detected and hence the false negative percentage will be null.

7. Conclusion

This work addressed the problem of misbehaving nodes in Vehicular Ad Hoc Networks. We showed that the presence of these nodes has a negative impact on the network stability, lifetime, overhead, and delay. Therefore, we proposed a two-phase model that is able to motivate the cooperation during clusters' formation and detect the misbehaving vehicles after the clusters are formed. A vehicle is considered as selfish or misbehaving when it over-speeds or under-speeds the maximum/minimum road speed limit. Giving incentives will not stop such behavior but will ensure the clusters formation. Thus, the main challenge was the detection of misbehaving vehicles. The detection is done in a cooperative manner where evidences from different watchdogs are gathered and aggregated using Dempster-Shafer. The decisions are then broadcasted among clusters to reduce the detection time and overhead. Simulation results show that the proposed model is able to increase the probability of detection up to 40%, minimize the false negatives, and reduce the percentage of selfish nodes up to 30% while maintaining the network stability and performance.

References

[1] H. Badis and K. Agha. QOLSR, QoS Routing for Ad Hoc Wireless Networks Using OLSR. *European Transactions on Telecommunications*, 16(5):427–442, 2005.

[2] K. Balakrishnan, J. Deng, and P. K. Varshney. TWOACK: Preventing selfishness in mobile ad hoc networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC '05)*, volume 4, pages 2137–2142, 2005.

[3] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 226–236, 2002.

[4] T. M. Chen and V. Venkataramanan. Dempster-Shafer Theory for Intrusion Detection in Ad Hoc Networks. *IEEE Internet Computing*, 9:35–41, 2005.

[5] T.-M. Chen and V. Venkataramanan. Dempster-shafer theory for intrusion detection in ad hoc networks. *IEEE Internet Computing*, 9, 2005.

[6] J. R. Douceur and T. Moscibroda. Lottery trees: motivational deployment of networked systems. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 121–132, 2007.

[7] M. Fiore, J. Harri, F. Filali, and C. Bonnet. Vehicular Mobility Simulation for VANETs. *40th Annual Simulation Symposium ANSS07*, 07:301–309, 2007.

[8] A. Gilat. *MATLAB: An introduction with Applications*. Wiley, 2008.

[9] J.-T. Isaac, J.-S. Camara, S. Zeadally, and J.-T. Marquez. A Secure Vehicle-to-roadside Communication Payment Protocol in Vehicular Ad Hoc Networks. *Computer Communications*, 31(10):2478 – 2484, 2008.

[10] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In *Proc. of the Multi Topic Conference Conference (International)*, pages 62–68, 2002.

[11] P. Jawandhiya, M. Ghonge, M.S.Ali, and J. Deshpande. A survey of Mobile Ad hoc Network Attacks. *International Journal of Engineering Science and Technology*, 2:4063–4071, 2010.

[12] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, Chapter 5:153–181, 1996.

[13] S. Lee, G. Pan, J. Park, M. Gerla, and S. Lu. Secure Incentives for Commercial Ad Dissemination in Vehicular Networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 150–159, 2007.

[14] F. Li and J. Wu. Frame: An innovative incentive scheme in vehicular networks. In *Proceedings of the 2009 IEEE international conference on Communications*, pages 4638–4643, 2009.

[15] Q. Lian, Y. Peng, M. Yang, Z. Zhang, Y. Dai, and X. Li. Robust incentives via multi-level Tit-for-Tat: Research Articles. *Concurr. Comput. : Pract. Exper.*, 20:167–178, 2008.

[16] S. Lim, C. Yu, and C.-R. Das. Cache Invalidation Strategies for Internet-based Vehicular Ad Hoc Networks. *Computer Communications*, 35(3):380 – 391, 2012.

[17] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, 2000.

[18] P. Michiardi and R. Molva. 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*.

[19] H. Yoo and D. Kim. Repetition-based Cooperative Broadcasting for Vehicular Ad-hoc Networks. *Computer Communications*, 34(15):1870 – 1882, 2011.

[20] S. Yousefi, E. A. andR. Elazouzi, and M. Fathy. Improving Connectivity in Vehicular Ad Hoc Networks: An Analytical Study. *Computer Communications*, 31(9):1653–1659, 2008.

[21] S. Zhong, Y. Yang, and J. Chen. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks. In *Proceedings ofI NFOCOM 2003*, pages 1987–1997, 2003.