LEBANESE AMERICAN UNIVERSITY

COOPERATIVE CLUSTERING MODELS FOR VEHICULAR AD HOC NETWORKS

By

OMAR MOHAMMAD ABDEL WAHAB

A thesis
Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

School of Arts and Sciences
June 2013

Lebanese American University

School of Arts and Sciences - Beirut Campus

Thesis Proposal Form

Name of Student: Omar Abdelwahab I.D.#: 201005232

Program / Department: M.Sc. Computer Science - Computer Science & Mathematics

On (dd/mm/yy): 20/3/2013 has presented a Thesis proposal entitled: Cooperative Clustering Models for Vehicular Ad hoc networks

in the presence of the Committee Members and Thesis Advisor:

Advisor: Azzam Mourad [redacted]
(Name and Signature)

Committee Member: (co-Advisor) Hadi Otrok [redacted]
(Name and Signature)

Committee Member: Rami Hawly [redacted]
(Name and Signature)

Comments / Remarks / Conditions to Proposal Approval:

Date: 27/3/13

Acknowledged by [redacted]
(Dean, School of Arts and Sciences)

cc: Department Chair
School Dean
Student
Thesis Advisor

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

LEBANESE AMERICAN UNIVERSITY
School of Arts and Sciences - Beirut Campus

Thesis Defense Result Form

Name of student: Omar Abdel Wahab I.D: 201005232

Program / Department: Computer Science & Mathematics

Date of thesis defense: June-12-2013

Thesis title: COOPERATIVE CLUSTERING MODELS FOR VEHICULAR AD HOC NETWORKS

Result of Thesis defense:

☑ Thesis was successfully defended. Passing grade is granted

☐ Thesis is approved pending corrections. Passing grade to be granted upon review and approval by thesis Advisor

☐ Thesis is not approved. Grade NP is recorded

Committee Members:

Advisor: Dr. Azzam Mourad
(Name and Signature)

Co-Advisor: Dr. Hadi Otrok
(Name and Signature)

Committee Member: Dr. Ramzi Haraty
(Name and Signature)

[redacted]

Advisor's report on completion of corrections (if any):

Changes Approved by Thesis Advisor: Dr. Azzam Mourad Signature: June 12, 2013

Date: 13 June 2013

Acknowledge by [redacted]
(Dean, School of ....SAS.....)

Cc: Registrar, Dean, Chair, Advisor, Student

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

LEBANESE AMERICAN UNIVERSITY

School of Arts and Sciences - Beirut Campus

Thesis Approval Form

Student Name: Omar Abdel Wahab I.D. #: 201005232

Thesis Title COOPERATIVE CLUSTERING MODELS FOR VEHICULAR AD HOC NETWORKS

Program : Masters in Computer Science

Department : Computer Science & Mathematics

School : School of Arts & Sciences

Approved by :

Thesis Advisor: Dr. Azzam Mourad Signature :

Co-Advisor : Dr. Hadi Otrok Signature

Committee Member : Dr. Ramzi Haraty Signature

Date : June 12, 2013

THESIS COPYRIGHT RELEASE FORM

LEBANESE AMERICAN UNIVERSITY NON-EXCLUSIVE DISTRIBUTION LICENSE

By signing and submitting this license, you (the author(s) or copyright owner) grants to Lebanese American University (LAU) the non-exclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video. You agree that LAU may, without changing the content, translate the submission to any medium or format for the purpose of preservation. You also agree that LAU may keep more than one copy of this submission for purposes of security, backup and preservation. You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant LAU the rights required by this license, and that such third-party owned material is clearly identified and acknowledged within the text or content of the submission. IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN LAU, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT. LAU will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Name: Omar Abdel wahab

Signature: [redacted]

Date: 12/06/2013

PLAGIARISM POLICY COMPLIANCE STATEMENT

I certify that I have read and understood LAU's Plagiarism Policy. I understand that failure to comply with this Policy can lead to academic and disciplinary actions against me.
This work is substantially my own, and to the extent that any part of this work is not my own I have indicated that by acknowledging its sources.

Name: Omar Abdelwahab

Signature: [redacted]

Date: 12/06/2013

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to all the people who helped me in accomplishing the thesis. A special gratitude I give to my supervisors Dr. Azzam Mourad and Dr. Hadi Otrok for their guidance and support. Moreover, I would like to thank the committee member, Dr. Ramzi Haraty, for his effort and precious advices. Last but not least, special thanks to my mom, dad, sister and brothers.

COOPERATIVE CLUSTERING MODELS FOR VEHICULAR AD HOC NETWORKS

OMAR MOHAMMAD ABDEL WAHAB

Abstract

In this thesis, we address the problem of clustering in Vehicular Ad Hoc Networks (VANETs) using QoS-OLSR protocol in the presence of selfish nodes. The QoS-OLSR is a clustering protocol that aims to prolong the network lifetime in Mobile Ad Hoc Networks (MANETs) by considering the bandwidth and energy parameters to calculate the Quality of Service (QoS) metrics. However, this protocol ignores the mobility metrics that characterize the vehicular topology. In fact, the high mobility of vehicles leads to frequent disconnections in the clusters and alters the QoS over the network. Moreover, the presence of selfish nodes would hinder the application of any clustering model. These nodes behave rationally and tend hence to maximize their gain regardless of the negative implications that may affect the network. Thus, this thesis is concerned with introducing a clustering model that aims to form stable clusters and maintain the stability during communications and link failures, while satisfying the Quality of Service requirements. This is achieved by (1) considering the high mobility metrics while computing the QoS, (2) using Ant Colony Optimization for MPRs selection, and (3) using MPR recovery algorithm that is able to select alternatives and keep the network connected in case of link failures. Moreover, the clustering model is accompanied with a Dempster-Shafer based model that detects the misbehaving vehicles and regulates the cooperation by (1) using cooperative watchdog model where evidences are collected by the different watchdogs and aggregated using Dempster-Shafer to make the final decisions, and (2) punishing the misbehaving vehicles by the different network nodes. Mathematical analysis and simulations are conducted to evaluate the performance of the proposed models.

Keywords: Vehicular Ad hoc Network (VANET); Quality of service (QoS); Ant Colony Optimization (ACO); QoS-OLSR; Game Theory; Dempster-Shafer; Tit-for-Tat.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

Safety systems are becoming nowadays an attractive topic for the research community with the increase in the number of traffic accidents and the complexity of the roads infrastructure. Vehicular Ad Hoc Network (VANET) is a new class of wireless networks that allows the communications among neighboring vehicles and between vehicles and nearby roadside infrastructure such as traffic lights and command centers. This technology offers a wide set of applications and services ranging from safety applications and traffic management systems to commercial and marketing services.

The basic point in such kind of networks is building efficient and secure communications. The clustering is one of the most important tasks in VANET that is concerned with organizing and optimizing the communications. It starts with the architecture of the network and covers the problems of routing and overhead. The existing clustering models dedicated to VANET such as [20, 21, 23, 24] are unilateral in the sense that they focus on

some VANET-specific features and ignore others such as bandwidth and connectivity. In other words, the current VANET clustering models consider exclusively the mobility parameters such as speed, distance, and direction, while ignoring the networking aspects of VANET. Safety, emergency, and multimedia applications of VANET require to assure a high level of Quality of Service (QoS) through the network. Practically, VANET requires real-time message propagation that is able to deliver data in a timely and accurate manner. For example, considering the case of safety applications, any delay in the message delivery may entrain dangerous and mortal accidents. Similarly, exchanging multimedia services such as files and video streams requires a high level of QoS.

Moreover, proposing a clustering model is not enough. Ensuring the proper implementation of the model is as important as the model itself. The interests of the model parties may contradict with the model objectives. Concretely, the vehicles are driven by humans who have selfish or rational thinking. During clusters formation, some drivers may derail the protocol principles and turn them to their advantage. Some others may prefer to save their time and resources by not following the model rules after clusters formation. Numerous contributions have been advanced to cope with these misbehaving vehicles. They can be categorized into two categories: incentive-based and detection-based mechanisms. Nonetheless, both of them suffer from several shortcomings that limit their efficiency such as lack of scalability and centralization, ambiguous collisions, and false alarms.

1.2 Problem Statement

In this thesis, we consider the case of Vehicular Ad hoc Network where a set of vehicles needs to form stable clusters and maintain the stability during the communications and in case of link failures. When achieving these goals, several problems arise. First, the high mobility of the vehicles may lead to a frequent and sometimes immediate disconnection of clusters. Suppose, for example a vehicle x, which is driving with a velocity of 120 km/h and willing to stop after 130 meters, has the highest QoS value in terms of bandwidth, connectivity and energy. If we use the existing QoS-based clustering algorithms such as QOLSR [2] and QoS-OLSR [17], they perform the clustering by electing a cluster-head for each set of neighbor nodes. The cluster-heads are responsible then of electing a set of nodes called Multi-points relay (MPRs) that connect the clusters. According to these algorithms, which consider only the energy and bandwidth parameters, the vehicle x will be elected as a cluster-head and has a high chance to be selected as MPR. However, this vehicle will stop after a short time and withdraw from the network, which would result in cluster disconnection. Second, the link failures in VANET are likely to occur. Thus, launching the MPRs selection whenever a failure happens would lead to wide overhead due to the exchange of a large set of messages.

Based on this, it is clear that the following objectives must be achieved to ensure the stability of the network. First, the clusters formation should take into consideration a trade-off between the Quality of Service (i.e. bandwidth, End-to-End delay, and Packet Deliver Ratio) and the mobility metrics (i.e. speed and residual distance). Second, there should be a MPR recovery algorithm that is able to provide quick alternatives and avoid the frequent

re-elections in case of link failures.

Moreover, each clustering model should be accompanied with the following question: "are the vehicles really willing to follow or to participate in the clustering model?". In fact, some vehicles (heads and 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 some driver prefer to over-speed (go beyond the maximum allowed speed) the other vehicles and get to the destination as earlier as they can. Others can also under-speed (drive below the minimum speed) for several purposes. Such behavior is known as passive attack since it would lead to denial of service due to not contributing in the routing process. This behavior, which 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 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 that 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 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, and (4) the bandwidth allocation will suffer from recurrent disconnections.

In this context, several approaches have been advanced to detect and punish the misbehaving nodes. The existing approaches assume that each node can serve as a watchdog to monitor the behavior of its neighbor node. In the light of its observations, the node then decides to cooperate or not. These mechanisms rely on the node-to-node relation and are able hence to penalize individual nodes, but not to regulate the cooperation in the whole network. In fact, several vehicles may cooperate with some nodes and refrain from cooperating with other nodes. Therefore, the observations of single watchdogs are not sufficient to judge the behavior of the other nodes. Moreover, the noise caused by the channel collisions and the high mobility of vehicles may affect the observations of the watchdogs. It may happen, for instance, that some packets are not received within the expected time due to intentional or unintentional network collisions.

In this case, the watchdog may accuse cooperative nodes to be misbehaving unjustly. Besides, some other watchdogs may accuse cooperative vehicles to be misbehaving unjustly with the intention of excluding them from being competitors in any future election procedure. All these problems contribute in decreasing the efficiency of detection and increasing the false alarms. Thus, the opinion of individual watchdog nodes is not sufficient. This raises the need for a cooperative approach that can improve the detection and regulate the cooperation in the presence of high mobility and channel collisions.

In summary, the problems tackled in this thesis can be listed as follows:

- Maintain the stability of the clusters during communications and in case of link failures without sacrificing the Quality of Service requirements.

- Improve the detection of misbehaving vehicles in the presence of channel collisions

and high mobility.

- Regulate the cooperation in the presence of rational nodes that present forged evidences.

1.3 Objectives

The main goal of this thesis is a to develop a QoS-based clustering model that forms stable clusters, while maintaining the Quality of Service and elaborate a repeated game theory that ensures the faithful implementation of the clustering model. In summary, the objectives of our approach can be listed as follows:

- Form stable clusters and maintain the stability of the network while keeping a good level of Quality of Service.
- Detect the misbehavior after clusters formation and improve the accuracy of the decisions.
- Regulate the cooperation in the network by punishing the misbehaving vehicles and rewarding the cooperative ones.

1.4 Approach Overview and Contributions

In this thesis, we propose a clustering model that optimizes the communications in Vehicular Ad Hoc Networks. Our clustering model is able to reduce the communications overhead by assigning the routing responsibilities to a specialized set of nodes called cluster-heads

and Multi-Point Relays (MPRs) instead of flooding the network by unnecessary messages. These nodes are elected according to an ant colony optimization algorithm responsible for identifying the optimal set of nodes that achieves the best path. The optimal set is characterized by both high level of QoS and acceptable level of mobility. The QoS is considered to ensure that these nodes are eligible to assume the routing responsibilities and deliver the packets in an accurate and timely manner. Therefore, we consider the bandwidth and connectivity parameters while electing these nodes.

The high mobility in VANET is an important issue that cannot be neglected while electing the head and MPR nodes. In fact, to perform the routing responsibilities perfectly, the nodes have to be respecting the speed limits since driving with a very high or very low mobility will induce frequent clusters disconnections and cause packet losses. Therefore, we consider the distance and velocity parameters while designating the optimal set of nodes. The Ant Colony Optimization algorithm is responsible for launching a proactive discovery to select the best paths achieving the aforementioned parameters. Thus, the ACO algorithm is able to maintain the stability of the clusters during communications by selecting long-living paths without sacrificing the QoS requirements.

To guarantee the proper application of the clustering model, we consider the problem of selfish or misbehaving vehicles after clusters formations. These nodes may impede the implementation of the proposed clustering models by refusing to follow the protocol rules. Practically, some drivers may consider that their cooperation in the clustering missions contradicts their own interests in the sense that they spend their time and resources without receiving any return. Therefore, we propose a detection mechanism dedicated to identify the misbehaving nodes that perform their misbehavior in the network. The detection is

done in a cooperative manner in order to collect the larger possible set of evidences and increase hence the credibility of the decisions. The aggregation of these evidences is done then using Dempster-Shafer theory to prevent the untrustworthy and uncertain observations from spoiling the decisions, as well as to overcome the problems of ambiguity and false alarms caused by the channel collisions and high mobility.

In the light of the decisions resulting from the detection mechanism, we model a game theory based on the Tit-for-Tat strategy to regulate the cooperation in the network. This game aims to convince the vehicles that their interest lies by cooperating in the networking functions and that refraining from cooperating would result in a loss for them. This is done by punishing the misbehaving vehicles by the different network nodes and rewarding the cooperative vehicles by the different nodes. Thus, the vehicles would tend to cooperate with the different network nodes to avoid the severe punishment resulting from the cooperative detection model.

The contributions of the thesis can be summarized as follows:

1. Extend the network lifetime and maintain the QoS requirements by introducing a QoS-based clustering algorithm that considers the mobility metrics.

2. Enhance the End-to-End delay and the Packet Delivery Ratio by selecting the MPR nodes using Ant Colony Optimization (ACO).

3. Reduce the communications overhead by introducing a MPR recovery algorithm that is able to select alternative MPRs in case of link failures.

4. Detect the misbehaving vehicles after clusters formation by using cooperative watchdogs monitoring for evidences collection and Dempster-Shafer theory for evidences

aggregation.

5. Regulate the cooperation in VANET by rewarding the cooperative vehicles and punishing the misbehaving ones.

6. Reduce the time and overhead of detection by using the information dissemination approach to propagate the detection results.

1.5 Thesis Organization

The remaining of the thesis is organized as following:

In Chapter 2, we give an overview on the main concepts that form our thesis such as: Ant Colony Optimization, Dempster-Shafer, and repeated game theory. We present then the related works in the fields of clustering, routing, and security in both mobile and vehicular ad hoc networks.

In Chapter 3, we present our proposed clustering model. After the clusters formation, we develop a routing algorithm based on Ant Colony Optimization. To better explain how the clustering and routing algorithms work concretely, we give an illustrative example that mimics a real vehicular network. Finally, we evaluate the performance of the proposed algorithms using mathematical analysis and simulation results. In addition, we explain in details the scenario and parameters followed during simulations.

In Chapter 4, we present the VANET-DSD model that is proposed to deal with the selfish vehicles. Thereafter, we evaluate the performance of the proposed model using simulations and give an illustrative example to show how this model works.

In Chapter 5, we provide the conclusion of this thesis, recapitulate its contributions, announce the plans for future work, and present the list of publications and submissions derived from this thesis.

Chapter 2

Background & Related Work

2.1 Introduction

We present in this chapter a general idea on the concepts that form our models. Our clustering model is based on Ant Colony Optimization algorithm that is responsible for a proactive discovery to select the optimal set of nodes that are able to guarantee the best path in terms of both Quality of Service and mobility. We explain hence how the Ant Colony Optimization (ACO) algorithm works and we give an illustrative example to show the efficiency of such algorithm for the routing issues. Moreover, the thesis uses the Dempster-Shafer theory to aggregate the evidences while detecting the misbehaving vehicles. Therefore, we provide an overview on this theory and show the motivation behind using it for evidences aggregation. To regulate the cooperation in VANET and ensure the faithful application of the proposed clustering model, a repeated game theory is employed. Thus, we give a definition of the game theory and show the importance of this field to analyze the interaction

among vehicles and enforce the cooperation. Finally, a summary of the main research contributions in the areas of clustering, routing, and security in both MANET and VANET is provided to show the need of our contribution and how it is different.

2.2 Ant Colony Optimization

Ant Colony Optimization is a probabilistic approach that is used to solve several discrete optimization problems. This approach inherits the normal behavior of ants that tend to find the shortest route while searching for food. In fact, the ants that move randomly towards the food indicate the other ants the shortest path to follow by depositing chemical substance called pheromone. Thereafter, paths with higher pheromone values are chosen to be followed. Thus, the shortest path will be continuously reinforced by more pheromone values since it will get marched repeatedly by the ants [12]. In contrary, the pheromone trails of the not marched paths will decrease due to the evaporation process. The evaporation is important to avoid the convergence for local optimal solution since without evaporation, the routes marched by the first ants will be extremely attractive for the following ants. The behavior of ants has attracted the researchers due to its dynamic nature that makes it adaptive to changes in real-time applications. Therefore, it has been widely used to solve many problems such as vehicle routing, Traveling Salesman Problem (TSP), machine scheduling, telecommunication networks, ad hoc networks routing, and personnel placement in airline companies.

To illustrate how the Ant Colony Optimization algorithm works, we show in the following how ACO can be applied to solve the TSP, which is a routing problem. Given a set

of cities and the distances separating each pair of cities, the problem of TSP is concerned with finding the shortest path that visits each city just once and turns back to the original city. This problem could be modeled as a construction graph where the cities represent the vertices and the distances separating the cities are the edges. The ACO solution works as follows: initially, every ant begins from a randomly selected vertex (city). Next, ants choose their next vertex to get in a probabilistic manner according to the highest pheromone value. This process continues until each ant has visited all the vertices on the graph only once. Now, pheromone values are updated on all the edges according to the quality of solution to which they belong so that the pheromone values for shorter routes would be greater than other routes. Thereafter, the pheromone values begin to evaporate and only the short tours will be reinforced by more pheromone values. This process is repeated for a specified number of iterations and the best discovered tour is maintained as final solution. This process is illustrated in Fig. 1.

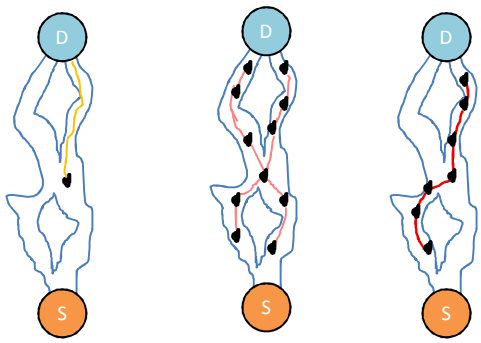

Figure 1: Ant Colony Optimization

In this thesis, the clusters in VANET have to communicate with each other in order to exchange packets and information. Therefore, a routing algorithm is needed. To guarantee

choosing the optimal paths in terms of QoS, mobility, and delay, an Ant Colony Optimization algorithm is used where proactive discovery by ant agents is initiated the find the best paths.

2.3 Dempster-Shafer

Dempster-Shafer is a mathematical theory elaborated by Arthur P. Dempster and Glenn Shafer [44]. This theory combines evidences from independent sources to come up with a degree of belief (belief function). It relies on two main ideas: (1) acquiring degrees of belief from subjective probabilities, and (2) combining these beliefs.

To illustrate how these two ideas work. As an example, let us assume that the subjective probabilities for the trustworthiness of Doctor John are known. The probability that John is trustworthy is 0.9 while the probability that he is untrustworthy is 0.1. Suppose that John says that a patient, Bob, suffers from diabetes. This allegation must be true if John is trustworthy but not necessarily false if he is untrustworthy. Thus, his testimony alone justifies a 0.9 degree of belief that Bob suffers from diabetes, but only a zero degree of belief (not a 0.1) that Bob is healthy. This zero does not imply that we are sure that Bob does not suffer from diabetes, but simply implies that John's testimony gives no reason to believe that Bob is healthy. The 0.9 and the zero together form a belief function.

In order to explain how the combination rule for degrees of belief works, we suppose that we know another doctor, called Alice, and that Alice is trustworthy with probability of 0.9 and untrustworthy with probability of 0.1. Assume that Alice witnesses as well that Bob suffers from diabetes. Since the trustworthiness of John is independent from

the trustworthiness of Alice, we may multiply the probabilities of these two events. The probability that both are trustworthy is $0.9 \times 0.9 = 0.81$. The probability that neither John nor Alice is trustworthy will be $0.1 \times 0.1 = 0.01$. Finally, the likelihood that at least one is trustworthy is $1 - 0.01 = 0.99$. Since both doctors said that Bob suffers from diabetes, at least of them being trustworthy means that Bob really suffers from diabetes, and we may hence assign this event a degree of belief of 0.99 as explained before.

On the other hand, if John's and Alice's testimonies were contradictory in the sense that john says that Bob suffers from diabetes while Alice says that he does not. Now, both cannot be right and hence cannot be both trustworthy. The prior probabilities that only John is trustworthy, only Alice is trustworthy, and that neither is trustworthy are 0.09, 0.09, and 0.01, respectively, and the posterior probabilities (given that not both are trustworthy) are $\frac{9}{19}$, $\frac{9}{19}$, and $\frac{1}{19}$, respectively. Hence we have a $\frac{9}{19}$ degree of belief that Bob suffers from diabetes (because John is trustworthy) and a $\frac{9}{19}$ degree of belief that Bob is sound and healthy (because Alice is trustworthy). Thus, Dempster-Shafer gives a weight for each evidence according to the trustworthiness level of the person giving the evidence and is necessary hence to discount evidences from untrustworthy observers upon aggregating the different testimonies. This appealing feature motivated us to use Dempster-Shafer while detecting the misbehaving vehicles since we are proposing a cooperative detection mechanism where evidences from different sources need to be aggregated. Therefore, Dempster-Shafer can be a solution to come up with reliable and credible decisions.

Another important characteristic of Dempster-Shafer is that it supports uncertain evidences. Suppose, for example, that Alice and John witness that a thief got in Bob's home. However, they might both heard the voice of a noise coming from a cat and thought it is

coming from a thief. To express this uncertainty, Bob can consider three evidences: (1) evidence for Alice's trustworthiness, (2) evidence for John's trustworthiness, and (3) evidence for the possibility of the presence of a cat. Now, Bob can combine these three items of evidences through Dempster's rule of combination to come up with the final decision taking into consideration that both observers may be unreliable. This feature can be exploited to improve the detection for misbehaving vehicles in VANET and overcome the problem of ambiguity caused by the high mobility of the vehicles and the channel collisions.

2.4 Repeated Game Theory

Game theory is a formal study of conflict and cooperation that applies whenever the actions of several peers are interdependent in the sense that the strategy of one game's component depends on the action of another game component [35]. The motivation behind using game theory is arriving at optimal decision. Consider, for example, that a company decides to reduce prices in order to augment its profit. Without considering the other players' (companies) actions, this action may be counterproductive and the company will lose money if the other companies apply a policy of price cuts. Here lies the importance of considering the different parties' strategies upon building any strategy. Game theoretical concepts have been widely used to solve problems in the fields of economy, biology, military, and computer science. To be clear and meaningful, each game should describe seven principal elements: players, actions, information, strategies, outcomes, payoff, and equilibrium. The players are the game parties that are responsible for making decisions. The actions are the set of options from which the players have to choose. The information represents the

learning of the player upon making decision. The strategies describe the set of principles that control the decisions of the player at each stage of the game. The outcomes are the expected or desired output of the game such as increase in profits. The payoffs describe the utilities yielded by the player in a specific outcome. Finally, the equilibrium represents a stable solution in which no player has an interest to take unilateral decisions and change his strategy.

Repeated game is a type of game theory in which players repeat their actions over and over again. To show the importance of using repeated game models, we present a motivating example based on the Prisoner's Dilemma [37]. The Prisoner's Dilemma models the investigation in a crime where two prisoners suspected to be committed a crime together are arrested. The investigator isolates them and suggests a deal saying: (1) if one of them confesses against the other one, the confessor will get free (payoff: 0) and the offender will spend 4 years in prison (payoff: 4), (2) if they both confess, they will bear a less cruel punishment by being jailed 3 years (payoff: 3), (3) if they both decline to confess, they will both bear a reduced sentence lack of evidences (payoff: 1). This deal can be summarized in the following bimatrix:

	Confess	Don't Confess
Confess	$(3, 3)$	$(0, 4)$
Don't Confess	$(4, 0)$	$(1, 1)$

Table 1: Payoff matrix of the Prisoner's Dilemma

The question is: how should the prisoners behave in such game? Each prisoner will have the following thinking:

- If the other prisoner confesses, I have to confess (since 3 years are less than 4 years).

- If the other prisoner refuses to confess, I have to confess (since getting free is better than 1 year in jail)

If the game is played one-shot, the best strategy for both players is to confess whatever the opponent did; thus staying 3 years in jail. However, if the game is played repeatedly, then the previous actions of each other become observable and they will know each other's decisions. Then, they may get a better result by not confessing together (1 year in jail). Here lies the dilemma of the prisoners.

In VANET, the vehicles have to make decisions about cooperating with each other. In making these decisions, nodes may behave selfishly, seeking exclusively for 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 may be appropriate, as game theory analyzes situations in which player objectives are in conflict. Moreover, the vehicles' decision depends on the other vehicles' decisions. Therefore, the repeated games are the best to model such situation.

2.5 Related Work

We provide, in this section, an overview of the main contributions in the fields of clustering, routing, and security in Mobile and Vehicular Ad Hoc Networks. We show then their limitations and motivate the need for our contributions.

2.5.1 Clustering in VANET

The communication in vehicular ad hoc networks entrains a high level of overhead, collision, and contention. In order to ensure efficient communications and mitigate the channel collision, overhead, and contention; there should be wireless backbone architecture able to elect some nodes to assume the network responsibilities. One solution is to gather the nodes into clusters and elect for each cluster a specified node to serve as cluster-head. The function of the cluster-head is to achieve both intra-cluster coordination, and inter-cluster communication. The intra-cluster coordination involves the coordination among the nodes within each cluster. In the inter-cluster communication, the cluster members charge the cluster-head to communicate with the other cluster-heads on behalf on them. The clustering imposes several challenges that should be taken into consideration such as: Which node has to be elected as cluster-head? How the election procedure is done? What are the requirements of the cluster-heads? How to increase and maintain the clusters lifetime?

Based on these challenges, several clustering algorithms for VANET have been proposed trying to answer these questions. In the following, we present an overview on the main contributions in this context.

APROVE [21] uses the Affinity Propagation algorithm to perform a clustering that minimizes the distance and the mobility between cluster-heads and members. The affinity metric is composed of responsibility and availability factors. Responsibility signals how compatible is one node to become exemplar while availability signals the willingness of the node to become exemplar.

Modified DMAC [23] was proposed on top of the original Basagni's Distributed and

Mobility-Adaptive Clustering algorithm. Its basic idea is to increase the stability and avoid re-clustering of the group of vehicles moving in different directions using a freshness parameter. In this algorithm, each node has to know its moving direction, current position, and velocity.

The authors in [24] propose a multi-hop clustering that uses the relative mobility between multi-hop away nodes. The beacon delay is used to calculate this metric. The clusterhead is elected according to the smallest aggregate mobility value. This approach considers also the problem of re-clustering by postponing it for some time.

In [15], the authors use complex metric composed of traffic conditions, connection graph, and link quality. Before assigning a node to a cluster, a check on the node's reliability is done using the membership lifetime counter. This has the advantage of avoiding needless re-clustering.

Presented clustering algorithms are proposed for different purposes such as clusters stability and overhead minimization. However, these algorithms ignore the Quality of Service which is important for safety, emergency, and multimedia services in VANET [22]. The Quality of Service relies primarily on connectivity, reliability, and end-to-end delay. Thus, we propose a new clustering protocol called VANET QoS-OLSR that is able to maintain the stability of the vehicular network while achieving a tradeoff between QoS requirements and mobility constraints.

2.5.2 Routing in VANET

After the clusters are formed, it is important to develop a routing algorithm able to achieve the communications among clusters. Routing refers to the process of carrying a packet from source to destination. In ad hoc networks, this process encounters several challenges. These challenges range from the dynamic topology of the network, scalability, limited physical security, to bandwidth and energy constraints. Based on these challenges, several routing protocols are presented MANETs and VANETs.

Routing based on Ant Colony Optimization

Routing Algorithm using Ant Agents for MANETs (RAAM) [18] was proposed to reduce the End-to-End delay. This can be done by creating multiple ant colonies that will travel through different paths to select the optimal one. Nevertheless, the overhead is the shortcoming that encounters this algorithm.

Ant-Colony-Based Routing Algorithm (ARA) [14] gets several paths from source to destination to transfer the packets. The drawback of ARA is that it cannot respond directly to topology change because of its passive nature. *Probabilistic Emergent Routing Algorithm* (PERA) [4] is, in contrary, an active method that periodically broadcasts ants so as to avoid the local best solution. However, the overhead of the routing table and the periodic broadcasts is a drawback that faces PERA.

The idea of AntHocNet [10] is to achieve a dynamic traffic loading balance for the whole network in order to reveal the importance of the Quality-of-Service issue. Nevertheless, AntHocNet suffers from several limitations such as the long search time and the early

convergence for large scales.

Routing based on Multi-point relay nodes

The classical *OLSR* [6] protocol has been modeled to cope with Mobile Ad hoc Networks (MANETs). Its basic idea is to elect a cluster-head for each group of neighbor nodes and divide hence the network into clusters. These heads then select a set of specialized nodes called MultiPoints relay (MPRs). The function of the MPR nodes is to reduce the overhead of flooding messages by minimizing the duplicate transmissions within the same zone.

QOLSR [2] was design on top of *OLSR* to consider the Quality of Service of the nodes during the election of heads and the selection of MPRs. In fact, QOLSR focuses on choosing optimal paths satisfying the QoS constraints. Though, the QOLSR is unable to deal with Vehicular Ad Hoc Networks since it considers exclusively the nodes' bandwidth ignoring thus some other important metrics such as mobility.

Then came *QoS-OLSR* [17], a cluster-based protocol that aims to prolong the network lifetime. When electing heads and choosing MPRs, this protocol considers, in addition to the bandwidth, some metrics that may affect the network lifetime such as the residual energy. Nevertheless, the QoS-OLSR has many limitations that make it inadequate to achieve the VANET requirements since it ignores the mobility of nodes while computing the QoS.

In summary, vehicular ad hoc networks have some characteristics that make them unique among other types of networks. Practically, VANET is characterized by the very high mobility of its nodes and the frequent disconnections. Numerous routing protocols have been proposed for MANETs, and some of them could be applied to VANETs. Nevertheless, simulation results proved that they suffer from bad performances due to the specific

features of VANET such as: rapid vehicles movement, dynamic packets exchange and high speed of nodes. Thus, finding and maintaining routes is a very challenging task in VANETs. In this work, we propose a routing algorithm based on Ant Colony Optimization that is able to select the best path in terms of QoS and mobility; thus maintaining the stability without sacrificing the performance.

2.5.3 Security issues in VANET

One additional question should be added to the clustering and routing challenges, which is: are the vehicles willing really to contribute in the clustering and routing processes by serving as cluster-heads and MPRs? In fact, proposing clustering and routing protocols for Vehicular Ad Hoc Networks is not enough. Practically, there should be a clear vision of how to ensure the proper implementation of these protocols. However, some vehicles may have objectives that contradict the proper functioning of these protocols. These selfish or misbehaving nodes seek to achieve their objectives neglecting the welfare of the whole network. Several contributions are proposed in the literature to deal with these nodes. These contributions can be categorized into: incentive-based approaches, and detection-based approaches.

Incentive-based Approaches

The *receipt counting method* [26] 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. [25] 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 [27] is made up 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. Nonetheless, 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.

Zhong et al. advanced an incentive scheme called *SPRITE* [28] that exploits the VCG mechanism [33] to choose the best available single path. It utilizes the game theory to establish the charges and credits, and encourages then each node to state its actions truthfully. Nevertheless, this method necessitates a Credit Clearance System.

In gross, the basic idea of incentive-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.

Detection-based Approaches

Tit-for-Tat [30] associates the incentive mechanisms with the reputation concept so that cooperating with more reputable nodes enable the nodes from increasing their own reputation and benefiting hence from a larger set of services. 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. [31] integrated the watchdog and pathrater concepts into the Dynamic Source Routing (DSR) [34] protocol. Their approach suggests preventing the detected misbehaving from forwarding packets instead of punishing them. However, according to this scheme the misbehaving nodes are remunerated vis-à-vis their behavior as their packets continue to be transmitted by others while they do not have to transmit and spend resources.

CONFIDANT [29] is based on sending an alarm to the network nodes whenever a misbehaving node is detected in the purpose of isolating these nodes from the network. Yet, the credibility of the received alarms is not guaranteed.

Overall, in the detection-based schemes, 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, despite the advantage of these kinds of approaches that they need no specialized hardware, they have several disadvantages that may limit their efficiency such as: ambiguous collision, limited transmission power, false alarms, and non-cooperative cooperation/defection

decision.

2.6 Conclusion

We explained in this chapter the Ant Colony Optimization algorithm, Dempster-Shafer theory, and repeated game theory that constitute the core of the thesis. We presented then the related works in the fields of clustering, routing, and security in both mobile and vehicular ad hoc networks. We showed the clustering algorithms are unilateral since they focus either on the QoS requirements or on the mobility metrics and are insufficient hence to form stable and robust clusters in VANET. Moreover, the existing routing algorithms are unable to select and maintain the optimal paths in terms of QoS, stability, delay, and overhead. Concerning the misbehaving vehicles, the existing approaches have several limitations that make them inefficient to deal with the selfish nodes such as: lack for scalability and centralization, ambiguous collisions, and false alarms.

Chapter 3

QoS-based Clustering Model for Vehicular Ad Hoc Networks

3.1 Introduction

This chapter addresses the problem of clustering in Vehicular Ad hoc Networks (VANETs). Several clustering algorithms have been proposed for VANET and MANET. However, the mobility-based algorithms ignore the Quality of Service requirements that are important for safety, emergency, and multimedia services of VANET, while the QoS-based algorithms ignore the high speed mobility constraints since they are dedicated for Mobile Ad hoc Networks (MANETs). Our solution is a new QoS-based clustering algorithm that considers a tradeoff between QoS requirements and high speed mobility constraints. The goal is to form stable clusters and maintain the stability during communications and link failures, while satisfying the Quality of Service requirements. When achieving this goal, the following problems arise:

- The high mobility of vehicles
- The link failures after clusters formation.

These problems are solved by: (1) considering the high mobility metrics while computing the QoS, (2) using Ant Colony Optimization for MPRs selection, and (3) using MPR recovery algorithm able to select alternatives and keep the network connected in case of link failures. Mathematical analysis and simulation results are used to evaluate the performance of the proposed protocol and prove its efficiency.

The remainder of this chapter is organized as follows. Section 3.2 explains the proposed protocol and describes its three components. Section 3.3 gives an illustrative example to show how the proposed protocol works. Section 3.4 analyzes the performance of the proposed protocol mathematically. Section 3.5 describes the scenario used during simulations. Section 3.6 presents empirical results. Finally, Section 3.7 concludes the chapter.

3.2 VANET QoS-OLSR Protocol

In this section, we describe the VANET QoS-OLSR protocol proposed to maintain the stability of the vehicular network. We explain its two components: the QoS-based Clustering, and the MPR recovery. Thereafter, we give an illustrative example explaining how our protocol works. The protocol can be summarized as follows. First, the cluster-head election algorithm elects a set of optimal cluster-heads. Next, the elected cluster-heads select a set of optimal MPR nodes responsible for transmitting the packets and connecting the clusters. Finally, the MPR recovery algorithm deals with link failures by selecting alternative MPRs.

Table 2: Notations

Symbol		Significance
N	:⇔	Set of nodes in the network.
$N_2(i)$	:⇔	2-hop away nodes from node i.
k	:⇔	Source cluster-head.
d	:⇔	Destination cluster-head.
$m(k)$	:⇔	Number of 1-hop away nodes from k.
$QoS(i)$	:⇔	Quality of Service value of node or path i.
$D(i)$	:⇔	End-to-End delay of path i.
$Pheromone(i)$	:⇔	Pheromone value of path i.
P	:⇔	Set of all paths leading to d.
$Prob(i)$	:⇔	Probability of pheromone for path i.
$MPRSet(i)$	:⇔	Set of MPRs selected by head i.
s	:⇔	Nodes Visited Stack.
$s(i)$	:⇔	i^{th} element of s.

3.2.1 QoS-based Clustering

A QoS-based clustering model for VANET is proposed. The clustering model relies on two algorithms, the cluster-head election algorithm and the MPRs selection algorithm. In the following, we present the notations and the details of these algorithms.

The Quality of Service Metric Models

To enhance the stability and the quality of service, we propose several Quality of Service (QoS) models. In the case of MANET, each node chooses its cluster-head according to several parameters such as proportional bandwidth, and residual energy. In our case, the Vehicular Ad hoc Network topology imposes new parameters to adopt in addition to bandwidth and connectivity namely the vehicle's mobility represented by residual distance and velocity. Therefore, we suggest five different QoS models according to different combinations of the QoS metrics. The bandwidth is considered to ensure the reliability, the connectivity is considered to increase the coverage of cluster-heads and MPRs, while the velocity and distance parameters are considered to maintain the stability of the network. The models are presented in Table 3.

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
Bandwidth Model
QoS(i) = *BW(i)*;
Proportional Bandwidth
QoS(i) = $\frac{BW(i)}{N(i)}$;
Proportional Bandwidth & Velocity Model (Prop. B-V)
QoS(i) = $\frac{BW(i)}{N(i)} \times VelRatio(i)$;
Proportional Bandwidth & Proportional Distance Model (Prop. B-DV)
QoS(i) = $\frac{BW(i)}{N(i)} \times \frac{DistRatio(i)}{VelRatio(i)}$;
Bandwidth-Connectivity & Proportional Distance Model (BCDV)
QoS(i) = $BW(i) \times N(i) \times \frac{DistRatio(i)}{VelRatio(i)}$;

The *VelRatio* of a node is the velocity ratio for this node. It is calculated according to Algorithm 1.

For example, if a car travels at 60 mph (96.56 km/h) on a trip and at 100 mph (160.93 km/h) on return trip. Then, the average total speed of the entire trip would be, Total average speed = 2*60*100/(100+60) = 75 mph (120.7 km/h). The *velocity(i)* can be any number between 80 and 120, and the *VelRatio* for nodes respecting the average speed will be ≤ 1, which increases the QoS value for these nodes (if we divide by velocity). In contrary, the nodes violating the speed limits will have a *VelRatio* > 1 and then a reduced QoS value. Similarly, the *DistRatio* of a node is the ratio of residual distance towards the destination. The calculation procedure of this ratio is explained in Algorithm 10. The distance parameter in the deployed systems can be obtained with help of the Global Positioning System (GPS)

Algorithm 1 Velocity Ratio Calculation

```
Initialization:
D = distance traveled by the car in each direction
t_1 = time spent on onward trip
t_2 = time spent on return trip
Total distance traveled by the car = D+D= 2D
Total time = t_1+t_2
AvgSpeed:= Total distance/Total time = 2D/(t_1+t_2).
procedure VELOCITYRATIOCALCULATION
    for each node i ∈ N do
        Velocity(i):= random integer between Minimum and Maximum speed
        VelRatio(i):=Velocity(i)/AvgSpeed
    end for
end procedure
```

Algorithm 2 Distance Ratio Calculation

```
Initialization:
MaximumDistance:= the distance between source and destination;
procedure DISTANCERATIOCALCULATION
    for each node i ∈ N do
        CurrentPosition(i): the current position of i
        ResidualDistance(i):=MaximumDistance-CurrentPosition(i)
        DistanceRatio(i):=ResidualDistance(i)/MaximumDistance
    end for
end procedure
```

Efficiency of Adding Mobility Metrics

Several contributions addressed the problem of QoS in Mobile Ad hoc Networks. The main proposed metrics in these contributions [1, 3, 5] were the connection duration, packet delivery ratio, end-to-end delay, and jitter. However, these schemes do not take into consideration the vehicular topology. Therefore, we suggest adding two new metrics dedicated to the VANET topology namely the velocity and the residual distance. Considering the residual distance 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. 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. Therefore, adding these VANET-dedicated parameters to the other important network-dedicated factors such as bandwidth and connectivity ensures to have a stable and reliable Vehicular Ad Hoc Network.

The Cluster-Head Election Algorithm

In the following, we model a cluster-head election algorithm that allows to electing a set of optimal cluster-heads and dividing the network into clusters. The algorithm works as follows. The nodes broadcast *HELLO* messages containing their QoS values two-hop away. Then, each node votes for its neighbor having the local maximal Quality of Service metric value. A node can as well vote for itself, if it has the maximal local QoS value. The nodes use their special *HELLO* messages, called *Election* messages, to locally broadcast their votes. Once the election procedure is done, the elected node acknowledges to serve as a cluster-head by sending an *Ack* message containing its public key. This message is sent also 2-hop away. Thereafter, the elected cluster heads act as MPR nodes for their electors. They should hence broadcast *TC* messages containing their electors. This algorithm is described in Algorithm 3.

Note that some modifications need to take place to the classical *HELLO* message. The first one is adding a flag, the *H* flag, to signal that a node has been designated as a cluster-head. The second is to add a new neighbor type in the link code. This *H _NEIGH* flag denotes that a neighbor has been elected as a cluster head. The *Election* messages are used by the nodes to indicate the neighbors for which node this neighbor has voted for.

Algorithm 3 Cluster Head Election Algorithm

```
procedure CLUSTERHEADELECTION
    for each node i ∈ N do
        broadcast HELLO message containing QoS(i) 2-hop away
        Let k ∈ N_2(i) ∪ {i} be s.t.
            QoS(k) := max{QoS(j) | j ∈ N_2(i) ∪ {i}}
        vote for k through the Election messages
        MPRSet(i) := {k}
    end for
    for each elected head k ∈ N do
        broadcast an Ack message 2-hop away
    end for
end procedure
```

Ant Colony Optimization Basic Notations

Ant Colony Optimization [12] imitates the real behavior of ants seeking for food. Ants search in the environment of anthill; when the food is found, they turn back to their home depositing a chemical substance called *pheromone*. Thus, the other ants that can smell this substance will follow the same path which will successively get passed. The shortest path will remain consequently followed among various paths due to the continuous reinforcement by pheromone trails.

In this work, we exploit this swarm intelligence algorithm to optimize the communications among clusters in a cluster-based QoS-OLSR protocol. To do so, some ant agents called *ANT-HELLO* are responsible for gathering information about all the paths and come up with an optimal choice in this context. The goodness of a path is estimated using the *pheromone value*. All pheromone values are set initially to 100 and are updated periodically according to the ants' observations. The nodes preserve probabilistic routing tables containing the probability of choosing a neighbor as the next hop for any destination. These tables are updated periodically by the ant agents based on the quality of paths. The quality of paths is expressed, in turn, in terms of Quality of Service and End-to-End delay.

An important element of the ACO, which is used to enhance the future solutions, is the *pheromone evaporation*. It is done according to the following equation [11]: $\tau_i = \lambda \times \tau_i + (1 - \lambda) \times q_i$ where λ is a smoothing factor between 0 and 1, and q_i is the measured route quality. The efficiency of the evaporation process can be summarized as follows. The pheromone trails start to evaporate as the time evolves. Thus, the goodness probability represented by the pheromone value will begin to disappear piecemeal unless they are reinforced by more ants. The optimal path will hence get marched by more ants than the other paths. This would increase its pheromone density. Thus, the evaporation phenomenon is important to avoid the convergence to local optimal solutions.

The MPR Nodes Selection Algorithm

Once elected, the cluster-heads are charged to select a set of optimal MPR nodes. This set of nodes is responsible for interconnecting the clusters and forming a connected network. The MPRs selection algorithm assumes that a flag indicating node's QoS value is added to the ANT-HELLO message.

The MPRs selection algorithm works as follows. Consider a case where two cluster-heads want to establish a communication between each other by selecting a set of MPR nodes. Initially, the source cluster-head sets the ANT-HELLO messages type to 0 indicating that these messages will be *forwarded* to the destination cluster-head. It then sends "m" messages (m is the number of 1-hop away neighbors leading to the destination head) to its 2-hop away nodes. Each intermediate node receiving this ant message calculates its QoS metrics value and inserts it in the appropriate field of the message. Meanwhile, the ants save each visited node in the "Nodes Visited Stack" field of the *ANT-HELLO* message to

Algorithm 4 MPR Selection Algorithm

Initialization:
$MPRSet(k) := MPRSet(d) := \emptyset$

Part I - Go Phase

procedure GoPhase
  for each source k **do**
    Set "Type" flag in *ANT-HELLO* message to 0 (forward)
    Broadcast $m(k)$ *ANT-HELLO* messages two-hop away
    for each intermediate node i **do**
      Compute $QoS(i)$
      Insert $QoS(i)$ into *ANT-HELLO*
    end for
  end for
end procedure

Part II - Back Phase

procedure BackPhase
  for each destination d **do**
    Set "Type" flag in ANT-HELLO message to 1 (backward)
    for each path i **do**
      Calculate $D(i)$
      Compute $QoS(i) := QoS(x)|x \in i$ and $QoS(x) := min\{QoS(u)|u \in i\}$
      Compute $Pheromone(i) := \ QoS(i) - D(i)$
      Compute $Prob(i) := Pheromone(i)/\sum_{j\in 1}^{P} Pheromone(j)$
    end for
    $MPRSet(d) := \{x|x \in j|prob(j) := max\{prob(u)|u \in P\}\}$
    Send back the ANT-HELLO messages 2-hop away
  end for
end procedure

Part III - Final Phase

procedure FinalPhase
  for each source k **do**
    $MPRSet(k) := \{x|x \in j|prob(j) := max\{prob(u)|u \in P\}\}$
  end for
end procedure

be used later for tacking back the route. The *ANT-HELLO* messages keep being propagated 2-hop away until reaching the intended cluster-head.

Once reached, this cluster-head sets the type of *ANT-HELLO* messages to 1 indicating that these messages will be *backwarded* to the source. It then extracts the QoS values of the intermediate nodes and sums up the QoS values for the nodes forming a single path. It calculates also the End-to-End delay for each path using the number of hops presented in the "Nodes Visited Stack". It updates hence the "route time" field accordingly. In order to compute the pheromone value for each path, it subtracts the End-to-End delay from the sum of QoS values for each single path. Now, this cluster-head node has the pheromone values of all the paths leading to it. Hence, it updates the "pheromone value" field with these values. Similarly, the pheromone value of each single node is calculated. This value is equal to the node's QoS value. Thereafter, this cluster-head calculates the probability of pheromone for each path. Afterwards, it selects the nodes belonging to the path having the higher probability of pheromone and located within the scope of its cluster as MPRs. Next, it sends back the *ANT-HELLO* messages two-hop away until reaching the source head through the chosen optimal path. This latter cluster-head, in turn, receives the messages and selects the nodes belonging to the optimal path and locating within its cluster as MPRs. Now, these two cluster-heads can communicate with each other through the selected MPR nodes. Note that the 3-hop away cluster heads may be reached through the 2-hop away nodes. The MPRs selection algorithm is presented in Algorithm 4.

3.2.2 MPR Recovery Algorithm

Link failures represent a big challenge to the stability of the vehicular network. Fig. 2 illustrates a link failure example where node 8 serving as MPR between Cluster 1 and Cluster 2 decides to leave its current cluster and join Cluster 3. Thus, the link between Cluster 1 and Cluster 2 is broken and they cannot communicate with each other until a new set of MPRs is selected. Link failures occur due to several reasons such as: mobility, interference, and congestion.

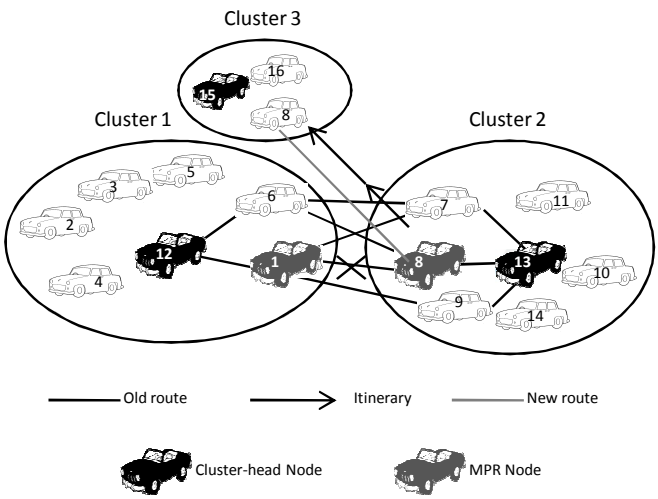

Figure 2: Link failure example

- Mobility: VANET is characterized by a high mobility resulting from the high speed of vehicles. This leads to recurrent disconnections and link failures.

- Congestion: The heavily loaded networks may produce congestions in Vehicular Ad hoc Networks, which would in turn cause link failures.

- Interference: The interference occurs mostly due to packets collisions. This collision may be intentional or unintentional. In both cases, the interference would result in

link failure.

In order to maintain the stability of the network and reduce the overhead caused by the repeated elections, we propose a MPR recovery algorithm capable to deal with link failures and keep the network connected. Our algorithm does not rely on lower level service to detect link failures. Instead, link failures are detected when an expected *TC* message from a certain MPR is not received. The algorithm works as follows. Once the cluster-head receives the *ANT-HELLO* message, it first sorts the "Nodes Visited Stack" in decreasing order according to the pheromone values. Then, if a cluster-head misses a *TC* message from a certain MPR, it first deactivates this link by removing this node from the stack. This means that a link failure by this MPR has occurred. It selects then the first element of the stack as MPR. This node leads to the same destination since it was visited by the *ANT-HELLO* message and has the higher pheromone value as a result of the sorting. This process is repeated until the stack becomes empty. When the stack becomes empty, the cluster-head launches the MPRs selection algorithm again in order to select a new set of MPRs. Thus, we are reducing the overhead by providing a simple method capable to deal with link failures and keep the network connected without the need for repeated re-elections. The MPR recovery algorithm is presented in Algorithm 5.

3.3 Illustrative Example

To illustrate how VANET QoS-OLSR works, we present a concrete example. Fig. 3 shows a network with fourteen nodes and six possible paths. Table 5 gives the pheromone value and the relevant probability of each path using the MPRs selection algorithm (refer to

Algorithm 5 MPR Recovery Algorithm

```
procedure MPRRECOVERY
    for each cluster-head k do
        Sort the "Nodes Visited Stack" s
        if(TCmsgNotRcvdTime(n) > TimeAllowedForTC()) then
            s := s − {n}
            MPRset(k) := i/i ∈ s(1)
            if(isEmpty(s)) then
                MPRSelectionAlgorithm()
            end if
        end if
    end for
end procedure
```

Table 4: QoS metrics values of nodes using the BCDV model

Nodes	1	2	3	4	5	6	7
QoS value	575.8	197	503.2	379.4	316.7	338.7	308.1
Pheromone	575.8	197	503.2	379.4	316.7	338.7	308.1
Nodes	8	9	10	11	12	13	14
QoS value	400	234.01	159.54	389.5	746.5	797.8	546.76
Pheromone	400	234.01	159.54	389.5	746.5	797.8	546.76

Algorithm 4), while Table 13 shows the QoS metrics value and the pheromone value for each node according to the BCDV model (Table 3). The pheromone value for a single node corresponds to the QoS value of this node. The QoS value of a certain path is determined by finding the minimal QoS for the path. It is computed as follows. Let's take the path p1: $QoS(p1) = min(QoS(node\ 6), QoS(node\ 7)) = min(338.7, 308.1) = 308.1$. After receiving the *HELLO* messages from its neighbors, a node votes for the neighbor having the local maximal Quality of Service metric value to be the cluster-head. This is done according to the BCDV QoS function (Table 3). Using the Cluster Head Election algorithm, nodes 12 and 13 are elected (Algorithm 3) as cluster-heads. From now on we call node 12 as CH-1 and node 13 as CH-2. To connect CH-1 with CH-2 which is 3-hop far away, CH-1 has 6 possible paths: 6-7-CH-2, 6-8-CH-2, 6-9-CH-2, 1-7-CH-2, 1-8-CH-2, 1-9-CH-2.

Table 5: The pheromone probability values using MPRs selection algorithm

Path	$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	Sum
Nodes	$6-7$	$6-8$	$6-9$	$1-7$	$1-8$	$1-9$	$-$
End-to-End delay (seconds)	125	256	233	479	107	108	$-$
Pheromone	521.8	482.7	339.71	404.9	868.8	701.81	3329.72
Probability	0.16	0.14	0.11	0.12	0.26	0.21	1

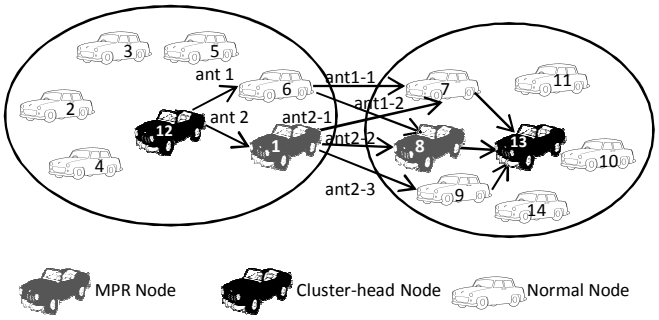

Figure 3: Vehicular Ad Hoc Network example

The source head CH-1 first sends 2 (according to the number of its 1-hop away neighbors) forward *ANT-HELLO* messages to all the 2-hops away nodes (nodes 7, 8 and 9). During the Go phase (Algorithm 4-Part I), each node receiving this message calculates its QoS metrics, encrypts this value using the destination head CH-2 public key, and inserts the encrypted value in the message. Upon receiving the messages (Algorithm 4-Part II), CH-2 decrypts the QoS values and subtracts them from the path route time to calculate the pheromone values. In our case, the path $1-8$ gives the higher pheromone probability (Table 5). Then, CH-2 chooses the node 8 as MPR, encrypts the QoS values using CH-1 public key and sends back the ANT-HELLO messages through the $1-8$ path. The source head (CH-1), in its turn, upon receiving the messages (Algorithm 4-Part III), selects node 1 as MPR. Now, the CH-1 and CH-2 can communicate through the path $1-8$. The selected cluster-heads CH-1 and CH-2 then sort the "Nodes Visited Stack" of the *ANT-HELLO* message in

decreasing order according to the pheromone values. Suppose now that node 1 serving as MPR fell out of the transmission range of the cluster-head CH-1 and causes hence a link failure. Using the MPR recovery algorithm (Algorithm 5), CH-1 deactivates the link of node 1 by removing it from the stack. Then, it selects the first element of the stack as MPR (node 6 in our case) since this node has the higher pheromone value after node 1 and leads to the same destination CH-2 given that the *ANT-HELLO* message has visited it. The path $6 - 8$ is then used to connect the two clusters. CH-1 can still handle the link failures in the same way until the stack becomes empty. If it is the case, then it has to launch the MPRs selection algorithm again.

3.4 Performance Analysis

Since the simulation results have become recently not sufficient for evaluating a proposed scheme, we analyze in this section the performance of several aspects related to our approach such as: overhead of the MPR selection algorithm, percentage of MPRs, network stability, end-to-end delay, and packet delivery ratio.

3.4.1 Computation Overhead

Each normal node *i* encrypts its Quality of Service (QoS) value. Later on, only the cluster-heads decrypt, using their private keys, the encrypted values in order to find the optimal path and select then the appropriate MPRs. They also encrypt back the QoS values using each other public keys. Hence, each normal node encrypts one message and does not decrypt anything. On the other hand, the cluster-head encrypts TNg_i and decrypts TNg_i

messages where TNg_i is the number of 2-hop away nodes leading to the desired destination. Note that each normal node must find the highest QoS value amongst its neighbors to elect it as cluster-head which requires $O(log(Ng_i))$ where Ng_i is the number of neighboring nodes. Therefore, each node approximately performs *O(1)* encryption, 0 decryption, and $O(log(Ng_i))$ to calculate the highest QoS value. The cluster-head node performs TNg_i encryptions and TNg_i decryptions. Thus, the computation overhead for each node is $O(TNg_i) + O(1) + O(log(Ng_i)) \approx O(TNg_i)$. Note that this overhead level is small in comparison with other algorithms since it is bounded by the number of 2-hop away nodes instead of being bounded by the number of all neighboring nodes. In the most of protocols that use ant colony optimization for the routing such as SACOM [19], AntHocNet [10], ARA [14], and PERA [4], the sender node has to broadcast the *ant* packet many hops away which causes a wide overhead over the network.

3.4.2 Communication Overhead

The cluster-head nodes broadcast three messages to at maximum 2-hop away nodes (*HELLO*, *ANT-HELLO*, and *Ack*). The normal nodes broadcast two messages (*HELLO*, and *Election*) also two-hop away. Later on, the MPR nodes broadcast *TC* messages over the network to indicate neighbors information. Hence, the total communication overhead of our algorithm is $Ng_i + 3TNg_i + 2TNg_i = Ng_i + 5TNg_i$, where Ng_i is the total number of nodes and TNg_i is the number of 2-hop away nodes. This level of overhead is acceptable compared with other Ant Colony Optimization based approaches where the source node has to broadcast the messages to many hops away. In this model, the cluster-head broadcasts three

messages 2-hop away only.

3.4.3 Percentage of MPRs

The number of needed MPRs is inversely proportional to the connectivity of the selected set of MPRs. This means, as the connectivity increases, the number of selected MPRs will decrease and vice versa. Consider a cluster of N nodes. Suppose that the cluster-head of this cluster selects a MPR with connectivity $N-8$. Hence, there will be $N-(N-8)$ nodes not covered by this MPR and need another set of MPRs to may communicate with other clusters. In contrary, if the connectivity of the MPR was $N-3$ there will be $N-(N-3)$ nodes not covered by this MPR and need another set of MPRs to may communicate with other clusters. Knowing the fact that $N-(N-3) < N-(N-8)$, it is clear that the number of uncovered nodes by the MPRs having higher connectivity level is less than that by the MPRs having less connectivity. Thus, as the connectivity of the selected MPRs increases, the need for selecting new MPR nodes will decrease. This shows that our proposed model, which assumes that the connectivity factor should be multiplied by the QoS function, is able to reduce the percentage of MPRs and decrease hence the jamming over the network caused by the large number of sent *TC* messages.

3.4.4 Network Stability

Consider a network composed of two clusters. The first cluster has to select a MPR in order to communicate with the other cluster. We have two axioms:

- Axiom1: the time for a MPR existing in the first cluster to reach the other cluster is

$t = d/v$, where v is the velocity at which the MPR is driving and d is the distance separating the MPR from the second cluster.

- Axiom2: $d = D$, where D is a constant.

Consider the two following cases.

- Case1: the first cluster-head elects a MPR with velocity V. So, the time for this MPR to get the other cluster is $t = D/V$.

- Case2: the first cluster-head elects a MPR with velocity $2V$. Thus, the time to get the other cluster will be $t = D/2V$.

Knowing the fact that $D/2V < D/V$, it is obvious that the MPR in the second case will move to the other cluster earlier and break down hence the communication between the two clusters. Therefore, the less the velocity, the more the stability and dividing the QoS function by the velocity will prolong the clusters' lifetime. Let's take a similar example for the residual distance. A cluster-head has to select a MPR in order to communicate with other clusters. We have the following axiom:

- Axiom1: the MPR is driving with velocity V where V is a constant.

Consider the two following cases.

- Case1: the first cluster elects a MPR having a residual distance of D.

- Case2: the first cluster elects a MPR having a residual distance of $2D$.

In the first case, the time separating the MPR from reaching the other cluster is $t = D/V$. In the second case, the time will be $t = 2D/V$. Since $2D/V > D/V$, the MPR in the

second case will be farther from reaching the other cluster, which is desirable. Thus, the link between the two clusters will last for more time. Consequently, the more the residual distance, the more the stability; Overall, we can notice that multiplying the QoS metrics function by the residual distance and dividing it by the velocity parameter increase the stability of the network.

3.4.5 End-to-End Delay

Consider a network with two clusters. The first cluster has to elect a MPR to be able to communicate with the other cluster. It has the choice between Node1 and Node2 belonging respectively to Path1 and Path2. Initially, the pheromone values of the paths are pheromone(Path1)=QoS(Path1)=α and pheromone(Path2)= QoS(Path2)=α, for example. According to the MPR selection algorithm (Algorithm 4), the cluster-head has to send some ants to detect the local optimal path in terms of pheromone value. Assume ants reported that the route times of Path1 and Path2 are t and $t + 10$ seconds respectively. According to Algorithm 4, the pheromone values are calculated in the following way: pheromone(Path1)=QoS(Path1)-time(Path1)=$\alpha - t$ and pheromone(Path2)=QoS(Path2)-time(Path2)= $\alpha-(t+10)$. Node1, which belongs to the path having the highest pheromone, will be then selected to serve as MPR. It's obvious that Node1 has to traverse less number of hops to reach the second cluster since $t < t + 10$. According to Ant Colony Algorithm, this node will still be selected as MPR until another local optimal choice arises due to the fact that it will get marched frequently by ants. Thus, the end-to-end delay represented by the number of hops is minimized in our protocol.

3.4.6 Packet Delivery Ratio

The Packet Deliver Ratio is defined as the total number of packets received by the destination over the total number of packets sent by the source within a period of simulation: $PDR = \frac{Total\ number\ of\ received\ packets}{Total\ number\ of\ sent\ packets}$. Thus, as the number of received packets increases, this ratio will also increase. The number of received packets relies on several factors including: connectivity, percentage of stability, and End-to-End delay. The connectivity and the percentage of stability ensure that the packets are transmitted along a continuous connected path without packet losses. This increases the probability of the packets to be received. The End-to-End delay is also important in this context. The increase of this factor increases the likelihood of packet losses and timeouts which reduces the total number of received packets and reduces hence the packet delivery ratio and vice versa. The above paragraphs show that VANET QoS-OLSR is able to increase the connectivity and the percentage of stability and decrease the End-to-End delay. As a result, VANET QoS-OLSR is able as well to increase the packet deliver ratio.

3.5 Simulation Model and Parameters

In order to compare the different models, we resorted to the use of Matlab [46] network simulator with the VanetMobiSim [13] traffic simulator. 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 [13]. 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. 4. 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. Each simulation round lasted 500 seconds after 30 seconds for the initialization. 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. To provide more accurate simulations, we took a confidence level of 95%. Then, we run independent simulations for each factor being evaluated (e.g, clusters stability, percentage of MPRs, packet delivery ratio...) and we calculate the mean average of this factor as well as the standard deviation. Using these calculated parameters, we calculate the lower and upper bounds for the confidence interval and we check whether the estimated mean average falls within this interval. If so, this means that we would expect 95% of the interval estimates to include the parameter average. Then, we stop the simulation runs. Otherwise, we keep increasing the simulation runs until attaining a 95% confidence interval (e.g., obtaining a mean average within the confidence interval). Experiment results show that running 100 independent simulation runs is able to provide results within the confidence interval.

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 4.1. For 0% of selfish nodes, there

Table 6: Simulation Parameters

Parameter	Value
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
Number of simulation runs	100 (95% confidence interval shown)

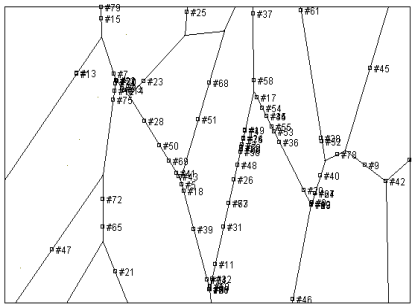

Figure 4: Screenshot of the vehicular movement simulation using VanetMobisim

is no need for detection. Similarly, above 50% the misbehaving nodes form the majority and their negative impact begin to diminish gradually since they can form new clusters and resume the networking functions again. The simulation parameters are summarized in Table 6.

3.6 Simulation Results

This section is divided into two parts. The first part presents the results after comparing our five proposed models (available in Table 3) with each others, while the second part is devoted to compare the preferred model among them with the QoS-OLSR and the classical QOLSR approaches. The factors to evaluate during the simulations are the: percentage of MPRs, percentage of stability, End-to-End, packet delivery ratio, and bandwidth average difference.

3.6.1 Comparison Between Our Proposed Models

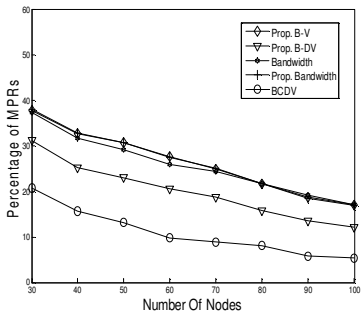

Figure 5: Percentage of MPRs

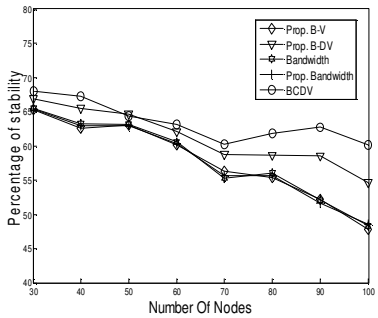

Figure 6: Percentage of stability

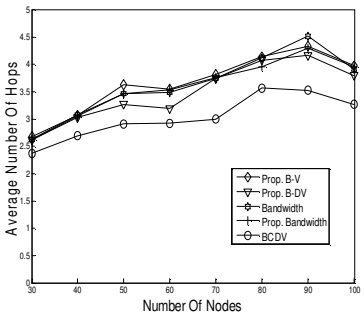

Figure 7: Path Length

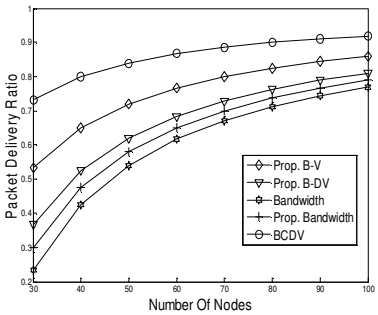

Figure 8: Packet Delivery Ratio

In this part, we present a comparison between our proposed models presented in Table 3 in order to find the best model that will be compared with the other approaches. In terms of MPRs, Fig. 5 reveals that the Bandwidth-Connectivity & Proportional Distance (BCDV) model gives the least percentage. This result is obtained by multiplying the connectivity by the other metrics instead of dividing it by the QoS metrics in the most of other functions (refer to Table 3). Concerning the clusters stability, which depends mainly on the distance and velocity factors, Fig. 6 shows that BCDV gives an improved percentage of stability compared to the other models. The average number of hops between the source and destination is also reduced with this model according to Fig. 7 which reduces the end-to-end delay. Similarly, the packet delivery ratio is increased using BCDV model as depicted in Fig. 8.

Table 7: Bandwidth Average Difference between our models

Models	Transmission Ranges		
	150	200	300
Bandwidth	0%	0%	0%
Proportional bandwidth	6.75%	3.85%	3.15%
Proportional B-V	7.57%	4.21%	4.55%
Proportional B-DV	6.63%	5.11%	3.76%
BCDV	7.08%	4.3%	3.92%

Moving to the percentage of bandwidth average difference, this factor can be defined as the bandwidth difference between the path having the maximal bandwidth value and the path currently selected. Table 7 reveals that the model adopting the bandwidth alone should annul this percentage and give hence the optimal solution in this context. For the remaining models, the BCDV and Proportional Bandwidth models compete to give the least average difference.

In gross, the BCDV model should be selected to be compared with other approaches. From now on, we call the BCDV model as VANET-QoS-OLSR when comparing it with the other approaches.

3.6.2 Comparison With Other Approaches

In this part, we present a detailed comparison between our proposed protocol, the cluster-based QoS-OLSR, and the classical without clustering QOLSR. The latter approach adopts only the bandwidth factor for calculating the QoS function, while the QoS-OLSR uses the proportional bandwidth combined with the residual energy of each node to build the Quality of Service function. In contrary to QOLSR, VANET QoS-OLSR and QoS-OLSR adopt the clustering concept so that each set of nodes elects their cluster-head which is, in turn, responsible for electing the appropriate set of MPRs entitled to communicate with other clusters.

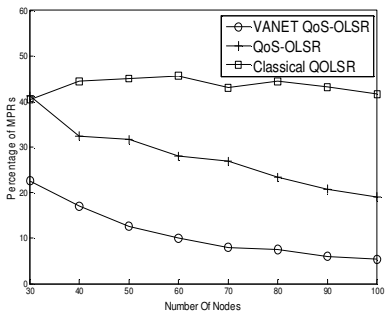

Figure 9: Percentage of MPRs

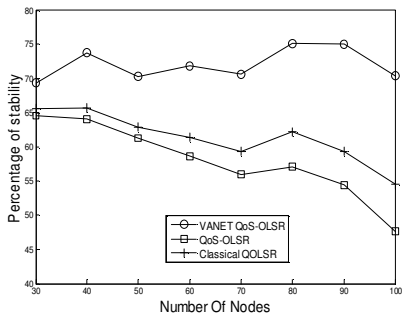

Figure 10: Percentage of stability

Percentage of MPR Nodes. The MPR is a node selected by the cluster-head to serve as a relaying point during the communications among clusters. It also includes the cluster-head

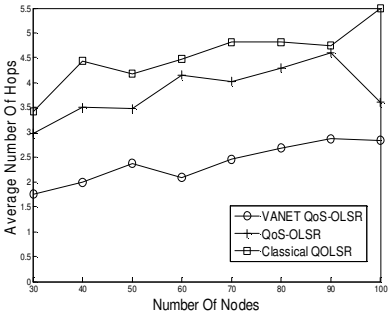

Figure 11: Path Length

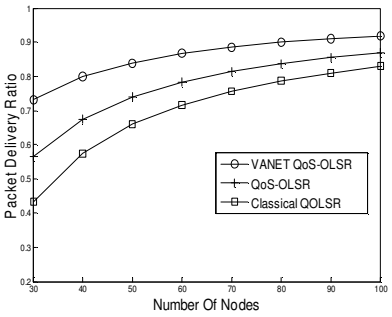

Figure 12: Packet Delivery Ratio

itself. Fig. 9 shows that the cluster-based models (VANET QoS-OLSR and QoS-OLSR) give a reduced percentage of MPR nodes since these multi-points relay are selected by a limited number of nodes namely the cluster-heads. Similarly, the VANET QoS-OLSR outperforms the QoS-OLSR by reducing the percentage of MPRs around 20%. This result can be justified by the fact that VANET QoS-OLSR multiplies the QoS function by the connectivity factor. This would lead to elect the MPRs having higher connectivity which reduces the need for electing wide set MPR nodes. In contrary, the QOLSR model divides the bandwidth by the number of neighbor nodes which will affect the protocol performance and raise the need for a larger set of MPRs. By reducing the number of MPRs, the VANET QoS-OLSR is decreasing the jamming over the network produced by the large number of exchanged *TC* messages. Therefore, this model seems to be efficient for dense networks.

Percentage of Stability. The percentage of stability is obtained by dividing the number of current nodes in each cluster by the previous number of nodes in the same cluster before a slot of time. If 60% or above of the nodes are still in the cluster, then the cluster is considered stable. Otherwise, it is considered unstable. Fig. 10 reveals that VANET

QoS-OLSR increases the percentage of clusters stability as the number of nodes increase. This result can be justified by the fact that our model takes into consideration the distance factor proportionally to the adopted velocity while calculating the QoS function. Hence increasing the distance and decreasing the velocity leads to a better QoS value. Multiplying by the distance factor guarantees that the clusters are formed by vehicles having convergent distance to traverse before reaching the destination. It guarantees as well that cluster-heads and MPRs have a considerable remaining distance to traverse in order to avoid the frequent disconnections. Dividing by the velocity ensures that vehicles violating speed limits have less chance to be cluster-heads or MPRs and that nodes belonging to the same cluster must have a convergent scale of speed.

Path Length. The path length is the average number of hops needed to transfer data between the source and destination. This factor reflects the End-to-End delay. In our protocol, the optimal path between a given source and destination is chosen according to the highest QoS value and the least expected route time. Fig. 11 describes the average number of hops yielded by the three protocols (VANET QoS-OLSR, QoS-OLSR and QOLSR) after sending messages from ten random sources to ten random destinations. The shown results prove that the VANET QoS-OLSR model gives less number of hops compared to other models. This improvement is earned by considering the route time while calculating the pheromone value used to select the MPRs. Moreover, using Ant Colony Optimization guarantees that the shortest path will still be chosen until a link failure occurs due to the fact that this path will get marched by ants over and over again and reinforced hence by more pheromone values.

Packet Deliver Ratio. In order to evaluate the efficiency of any routing algorithm, two major metrics should be considered: the End-to-End delay and the packet delivery ratio. We evaluate in this part the efficiency of the MPRs selection algorithm by measuring the packet delivery ratio yielded by this algorithm. The packet delivery ratio is obtained by dividing the total number of received packets by the total number of sent packets. Fig. 12 reveals that VANET QoS-OLSR is able to increase this ratio. This is due to the fact that it is able to increase the connectivity, maintain the stability, and decrease the End-to-End delay compared to the other approaches.

Table 8: Bandwidth Average Difference

Models	Transmission Ranges		
	150	200	300
Classical QOLSR	0%	0%	0%
QoS-OLSR	7.03%	4.53%	3.90%
VANET QoS-OLSR	7.08%	4.3%	3.92%

The Bandwidth Average Difference. The bandwidth average difference can be defined as the bandwidth difference between the path having the maximal bandwidth value and the path currently selected. Thus, the decrease of this aspect improves the Quality of Service over the network. Table 8 presents the percentage average difference for a 100 nodes network using the three scenarios: VANET QoS-OLSR, clustered QoS-OLSR, and without clustering QOLSR. According to this table, the classical QOLSR model shows a zero percentage average difference since the best path is selected according to the optimal

bandwidth path. The two remaining models show almost similar percentage of average difference with a slight advantage for the QoS-OLSR over VANET QoS-OLSR with a transmission range of 150 and 300 meters. For 200 meters of transmission range, the VANET QoS-OLSR model shows a better average difference around 0.23%. In the light of these results, we can notice that the average difference given by VANET QoS-OLSR is not such big. Moreover, this value is tolerable since in this model we need to combine the bandwidth with a bunch of other important metrics (speed, connectivity and distance) to ensure other important factors namely the stability, congestion and delay.

3.7 Conclusion

In this chapter, we proposed VANET QoS-OLSR protocol that aims at maintaining the stability of the vehicular network while achieving the Quality of Service requirements. The protocol is composed of two components: (1) QoS-based clustering using Ant Colony Optimization, and (2) MPR recovery algorithm. To ensure the stability of clusters, we add the velocity and distance that represent the mobility metrics to the QoS function. Thereafter, the protocol elects the cluster-heads according to the local maximal QoS value. The cluster-heads select then a set of optimal MPRs satisfying both mobility and routing constraints according to an Ant Colony Optimization algorithm. Finally, a MPR recovery algorithm is introduced to select alternatives and keep the network connected in case of link failures. Simulation results prove that our protocol is able to extend the network lifetime up to 12%, reduce the percentage of selected MPRs by 20%, increase the packet delivery ratio by 10%, and decrease the path length up to 2-hops. However, the faithful application of

this clustering protocol is not guaranteed. In fact, some vehicles (cluster-heads and MPRs) may behave selfishly by refusing to cooperate with the other vehicles in the networking functions. This misbehavior would hamper the clustering objectives and entrain negative implications on the network lifetime, stability, overhead, and delay. Therefore, we propose in chapter 4 the VANET-DSD model that (1) detects the misbehaving vehicles after clusters formation, and (2) regulates the cooperation in the network.

Chapter 4

A Dempster-Shafer based Tit-for-Tat Strategy to Regulate the Cooperation

4.1 Introduction

This chapter addresses the problem of selfish nodes in VANET using our proposed clustering protocol VANET QoS-OLSR. According to this protocol, nodes might behave selfishly either during the clusters formation by claiming bogus information or after clusters' formation by speeding up or down the road average speed limit. Several contributions have been proposed to tackle this problem during clusters formation such as [47] and [48]. In this thesis, we address the problem of selfish nodes after the clusters are formed. In this case, the nodes may refuse to cooperate with each other and serve as cluster-heads or MPRs in order to save their time and resources. Such behavior will entrain negative implications on the percentage of MPRs, percentage of stability, percentage of clusters' disconnections, and end-to-end delay as depicted in figures 13, 14, 15, and 16, respectively.

Fig. 13 depicts the impact of selfish nodes on the percentage of elected MPRs. The plot indicates that the clusters need to elect more number of MPRs as long as the number of misbehaving vehicles increases in the network. This is because the clusters do not stand up a long time due to the high mobility of the selfish nodes. Fig. 14 reveals that the increase in the number of selfish nodes will deteriorate the stability of the network gradually. This is due to the fact that the over-speeding nodes move quickly to other clusters while the under-speeding nodes remain for a long time in the same cluster.

Fig. 15 shows that the percentage of disconnected cluster-heads increases in conjunction with the increase in the percentage of selfish vehicles due to the high mobility of the selfish MPRs connecting the clusters. Finally, Fig. 16 describes the impact of selfish vehicles on the End-to-End delay represented in terms of average number of hops. It is obvious that this delay increases considerably as the percentage of selfish nodes increases. Suppose, for instance, a path between two clusters composed of three MPR nodes $MPR1$, $MP2$, and $MPR3$. Suppose that the $MPR2$ decided to over speed. The path connecting the two clusters will then break down and the intended packets between the two clusters will not be received on time.

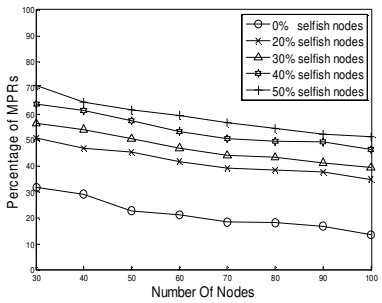

Figure 13: Impact on MPRs

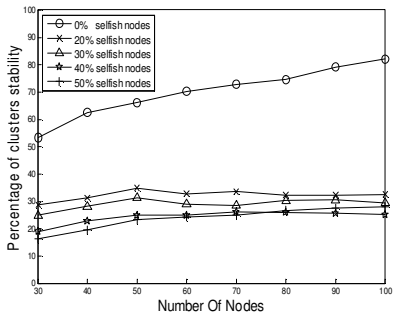

Figure 14: Impact on stability

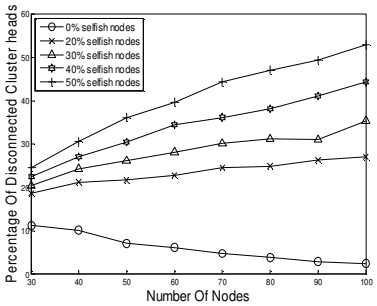

Figure 15: Disconnections

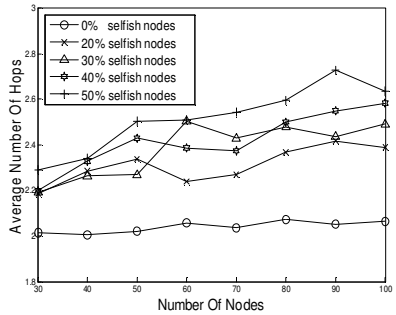

Figure 16: Impact on path length

As a solution, we propose a model that is able to detect misbehaving nodes after clusters are formed, and regulate the cooperation in the network. To detect misbehaving vehicles after clusters' formation, cooperative watchdog model based on Dempster-Shafer is modeled where evidences are aggregated and cooperative decision is made. After detecting the misbehaving vehicles, there should be a mechanism able to regulate the cooperation by punishing the misbehaving vehicles and rewarding the cooperative vehicles. 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.

The remainder of the chapter is organized as follows. Section 4.2 explains the proposed

approach in details, explains the model used for simulation, and presents empirical results. Section 4.3 gives an illustrative example to show our model works. Finally, Section 4.4 concludes the chapter.

4.2 VANET-DSD Model

In this section, we describe the VANET Dempster-Shafer Detection (*VANET-DSD*) model. The model starts with modeling the packet forwarding problem in VANET as a game theory without and with collisions. Thereafter, we present the Tit-for-Tat strategies proposed to detect the misbehaving vehicles after clusters formation as well as to regulate the cooperation.

4.2.1 The Game Model Without Packet Collisions

Game theory [35] is a formal study of conflict and cooperation that applies whenever the actions of several peers are interdependent. In Vehicular Ad hoc Networks, 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 head and 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 head and 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 participant 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 9) which is the worst case since:

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

Thus, the strategy (*Drop*,*Drop*) represents the Nash Equilibrium [36] 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.

This leads us to the classical Prisoner's Dilemma [37] identified by the payoff matrix presented in Table 10, 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 [37]. 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 [38], 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 = u + i
Punishment P = o + p
Temptation T = o + i
Sucker S = u + p

where u is a negative payoff representing the forwarding cost, i is a positive payoff representing the reward of being cooperated (served), o is null payoff representing the cost of dropping, and p is a negative payoff representing the cost of being defected (not served).

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

Table 9: Payoff matrix of the Packet Relaying Game

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

Table 10: Payoff matrix of the Prisoner's Dilemma

4.2.2 The Game Model With Packet Collisions

A major problem may face the implementation of the reputation-based strategies which is the packets collisions [39]. 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. 17, node V_3 is located within the transmission range of node V_2 and

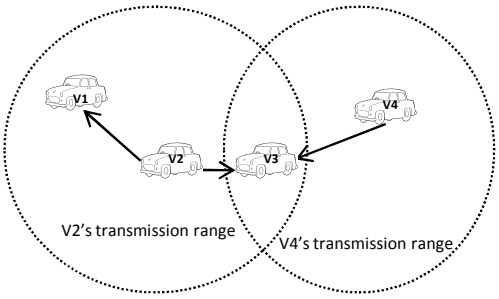

Figure 17: Collision Scenario

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 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 [40]. 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* [41] 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 [42] where each node separates the packets originating from itself 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 equation 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) [35]. 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.

4.2.3 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.

Set-up and simulation scenarios

Consider that we have a group of N participants (cluster-heads and MPRs) in a packet relaying game, each participant is a member of one cluster at a time and the routing is done according to a clustered-based QoS-OLSR protocol [17]. Each participant 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 section 4.1.

The game will run for 24 hours (86400 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 participant i, makes a request r. The relay nodes (heads or MPRs) can either decline the request or cooperate by forwarding the packets. In the former case, a participant j can be selfish or, simply, does not have sufficient resources (bandwidth, storage space, time). In the latter case, a participant 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. To do so, we take every 10 minutes (600 iterations) an average of the accumulated gain to obtain in total 144 plot points ($\frac{84600}{600}$) along the 24 hours.

We follow the following asynchronous prisoner's dilemma game while evaluating the different strategies:

- We have a total of 100 heads and 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 $0 < P_r < \frac{1}{5}$. Each participant has his own P_r value ranging from 0 to $\frac{1}{5}$. The values of P_r are distributed normally among the participants to simulate the variability among them. This variability expresses the probability of a participant's ability to relay a packet according to several constraints such as enough bandwidth, storage space, or even time (e.g., a participant (driver) may has an urgent case that forbids him from wasting his time by forwarding packets).

- 86400 requests are made sequentially.

- In each iteration, a particular source node is chosen randomly to make a request. Thus, the probability of requesting is $P_s = \frac{1}{100}$, for any given node.

- This participant 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 (Section 4.2.2).

In the following, we define the game parameters that can satisfy the conditions of the asynchronous repeated prisoners dilemma game:

- Forwarding Cost: u = -1

- Drop Cost: o = 0

- Gain from a fulfilled request: i = β = number of packets supposed to be transferred/request

- Loss from a non-fulfilled request: p = -β = -(number of packets supposed to be transferred/request)

The number of packets supposed to be transferred per request may be obtained by dividing the available bandwidth at node x by the mean packet size which we suppose it to be 1kb; that is, if the available bandwidth at node x is 1Mb and the mean packet size is 1kb then the number of packets supposed to be transferred by this node is $\frac{1Mb}{1kb}$=$\frac{1000kb}{1kb} = 1000$ packets. Note that the parameter u 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 o is hence greater than u which means that dropping the packet would be more beneficial for the rational vehicle. Furthermore, $o - u$ is less than $i - p$ 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 i 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 $\beta > \beta - 1 > -\beta > -\beta - 1$).

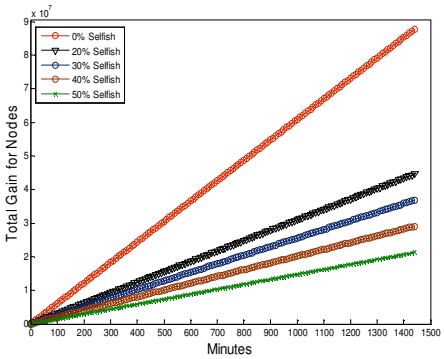

Figure 18: The optimal upper bounds

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.
- 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 section 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.

Concerning the simulation scenario, we use the scenario described in Chapter 3 to simulate the Tit-for-Tat strategies.

Fig. 18 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.

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 participant j maintains the historic records $H_{ji}(_k)$ of the last K actions with another participant i. If the accumulated value from $H_{ji}(1)$ to $H_{ji}(k)$ exceeds $\frac{k}{2}$, player i is deemed cooperative and player j will decide to cooperate with player i; otherwise, player j will refuse to cooperate. However, the cooperation decision depends as 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. $\max\limits_{1 \leq h \leq k} H_{ji}(h)$. Let $D_{ij}(t)$ describes the decision taken by player i to cooperate with player j or not, we can write the following equation:

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

Fig. 19 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 minutes). Starting from this time, the payoff of the cooperative nodes reaches a deadlock and begins to decrease as proven in the section 4.2.2. In fact, at this time each vehicle will

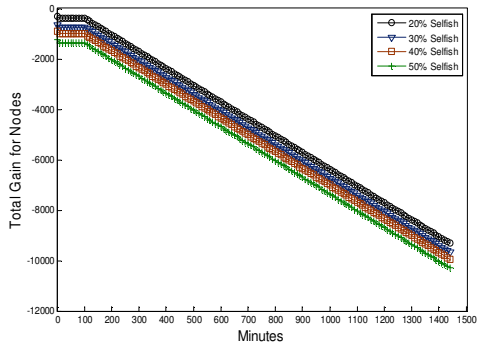

Figure 19: Classical Tit-for-Tat

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 coefficient which is equal to -(number of packets supposed to be transferred/request).

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) getting 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) [43], is a variation of the traditional Tit-for-Tat. This strategy forgives periodic defections with a certain probability. If two nodes playing

Tit for Tat strategy against one another and one of them defects by mistake or due to collisions, this would result in a long series of mutual punishment where both players fail to profit. In contrast, if the same situation happens for two players of the Generous Tit-for-Tat strategy or one Generous Tit-for-Tat player and one Tit-for-Tat player, the reciprocal punishment series will end as soon as the Generous Tit-for-Tat player responds to the defection by cooperation in the next round. Thus, the cooperative player i of GTFT cooperates at a regular basis of k movements with the other player j regardless of the prior history from $H_{ji}(1)$ to $H_{ji}(k)$. In addition, instead of waiting $\frac{k}{2}$ cooperations to consider a player cooperative in the classical Tit-for-Tat, one cooperation is enough in GTFT to assume the other player cooperative. Let $f_{ji}(t)$ be a fulfilled request by vehicle j to vehicle i at time t. The GTFT corresponds to the following equation:

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

Fig. 20 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

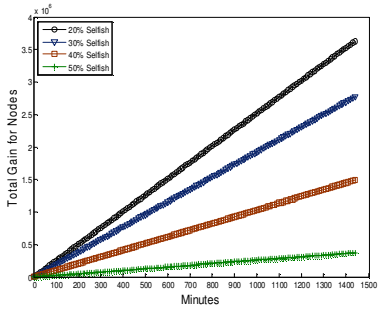

Figure 20: Generous Tit-for-Tat

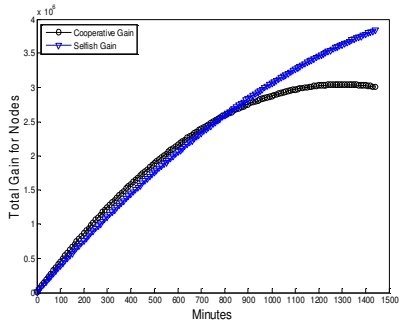

Figure 21: Selfish vs. cooperative

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. 21. Consequently, the rational vehicles will find that their interest lies in the defection. Thus, the game goes on vicious circle.

Tit-for-k-Tats

Tit-for-Two-Tats [35] 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 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.

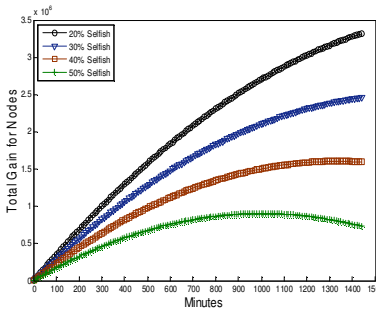

Figure 22: Tit-for-two-Tats

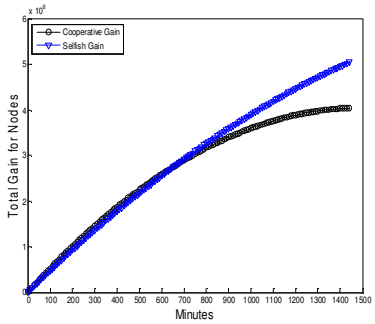

Figure 23: Selfish vs. cooperative

Fig. 22 depicts that the gain yielded by the Tit-for-Two-Tats strategy is close to the gain generated by the Generous Tit-for-Tat. Fig. 23 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.

Table 11: Average gain of cooperative nodes with different percentages of selfish nodes

K	Percentage of selfish nodes			
	20%	30%	40%	50%
2	1969000	1537100	1105100	673100
4	2098700	1666700	1234700	802700
6	2185100	1753100	1321100	889100
8	1969000	1537100	1105100	673100
10	2746700	2314700	1882700	1450700

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

Table 12: Comparison between the average gain of cooperative nodes and selfish nodes

K	Gain of Cooperative Nodes	Gain of Selfish Nodes
2	1776700	1969000
4	1976800	2098700
6	2068800	2185100
8	2309300	2401100
10	2676500	2746700

each strategy and we group them into tables because of space constraints. Table 11 reveals that the average gain of cooperative nodes in the different strategies is close somehow to the optimal 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 12, 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.

Dempster-Shafer based Tit-for-Tat

After being elected as heads and 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 node. 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 model. This model is made up of five phases: reputation calculation, watchdogs monitoring, votes aggregation, Tit-for-Tat cooperation regulation, and information dissemination.

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). The payment of cluster-heads is explained in Algorithm 6.

On the other hand, the MPR node that connects the 2-hop away cluster heads should be paid by each of the two head node according to Algorithm 7.

Algorithm 6 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 7 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

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 8.

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 1000Mb/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 $\frac{109}{355} \times 1000$. The bandwidth share of the second node will be $\frac{130}{355} \times 1000$ while the share of the third node will be $\frac{116}{355} \times 1000$ with $\frac{109}{355} \times 1000 + \frac{130}{355} \times 1000 + \frac{116}{355} \times 1000 = 1000$Mb/s.

Watchdogs Monitoring : In the watchdogs monitoring phase, each 1-hop away neighbor node is designated as watchdog to monitor the behavior of all head and MPR nodes situating within its transmission range. These watchdog nodes can overhear the communications between nodes locating in their transmission ranges. Hence, each watchdog node specifies an expected time for the packet to be sent.

After the expiry of this time, the watchdog that maintains a buffer of recently sent

Algorithm 8 Payment Algorithm for MPRs Connecting 3-hop Clusters

1: **Initialization:**
2: Let $CH_3(k)$ be the 3-hop away nodes from k.
3: Let x and y be elected MPR nodes for the nodes in $CH_3(k)$.
4: Let k be an elected cluster head.
5: Let l be an elected cluster head.
6: Let $R(i)$ be the reputation of node i.
7: Let $P(k)$ be the payment offered by the head node k.

8: **procedure** THREEHOPMPRPAYMENT
9: The path (k, x_1, y_1, l) maximizes $min(QoS(x_1), QoS(y_1))$ among all paths connecting k to l.
10: The path (k, x_2, y_2, l) maximizes $min(QoS(x_2), QoS(y_2))$ among all paths connecting k to l and $min(QoS(x_2), QoS(y_2)) < min(QoS(x_1), QoS(y_1))$.
11: $R_{t+1}(x) = R_t(x) + P(k) + P(l)$.
12: $R_{t+1}(y) = R_t(y) + P(k) + P(l)$.
13: **end procedure**

packets will compare each overheard packet with the packet in the buffer to see if there is a match. If so, this means that the packet was delivered correctly and the watchdog will mark the sender head or MPR as "cooperative". Otherwise, it will not mark this head or MPR as "selfish" directly but it will accuse it to be "suspicious" awaiting the observations from the other watchdogs to build the 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 section 4.2.2. 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 heads and 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 unable to regulate the cooperating inside the network but only to punish some nodes temporarily. Here lies the importance of aggregating the observations from different watchdogs in order to build a final collective decision. We present hence the votes aggregation phase.

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 head or MPR is cooperative while it is not if a plot between these two nodes took place. Similarly, some other voters may accuse cooperative heads or 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.

To do so, we have chosen the Dempster-Shafer theory [44] of evidences to be used while aggregating the watchdog votes. Dempster-Shafer is a mathematical model that is characterized by considering the uncertain evidences and by its ability to aggregate the evidences from independent sources. The motivation behind using Dempster-Shafer in the cooperation regulation in VANET can be summarized as follows: (1) the usefulness of Dempster-Shafer to represent and combine differen types of evidences coming from indpendent sources, (2) the fact that Dempster-Shafer represents uncertain evidences makes it appealing to model the ambiguity in the detection caused by the high mobility of vehicles and the collisions in VANET, and (3) The efficiency of Dempster-Shafer in the evidences

aggregation which made it widely used 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.

$$\alpha(L) = \frac{Reputation(L)}{\sum_{j=1}^{n} Reputation(L)} \tag{3}$$

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 among the nodes. Let's define a power set Ω composed of three main elements: hypothesis $H = C$ stating that L is cooperative; hypothesis $H = S$ that it is selfish; and hypothesis $U = \Omega$ that L is either cooperative or selfish. This latter hypothesis is important to express the uncertainty in the decisions when some watchdogs are not sure if a node 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 uncooperative, 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:

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

where H represents a hypothesis. The above function may be resolved by combining each pair of beliefs. This can be done as follows [45]:

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

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

Where:

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

Dempster-Shafer results in a judgment value between 0 and 1 expressing the degree of belief in that judgment. The motivation behind using Dempster-Shafer is to discount the unreliable evidences upon building the final decision. In fact, the majority rule would produce inaccurate judgement if some nodes build malicious alliances. The averaging model may be a substitute for the majority rule. However, unreliable watchdogs could produce errors in the final judgment since this model is strongly vulnerable to outliers. Dempster-ShaferŠs

advantage here lies in the fact that it excludes evidence from untrustworthy or uncertain observers and prevents them from beating the trustworthy evidences even they constitute the majority. 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 Fig. 24 and Fig. 25 In the following, simulations are conducted to compare two models: the Dempster-Shafer model and the averaging model. We call the Dempster-Shafer model 'With Dempster-Shafer" and the averaging model 'Without Dempster-Shafer".

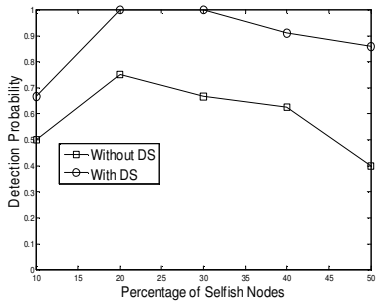

Figure 24: Probability of detection

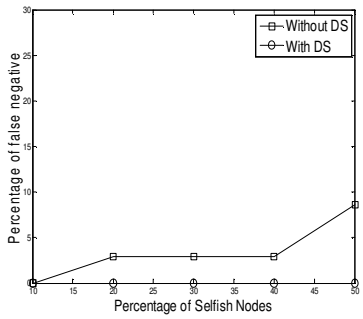

Figure 25: Percentage of false negatives

The probability of detection is obtained by dividing the number of detected selfish nodes by the real number of selfish nodes. It gives a measure of the efficiency of the proposed model. As depicted in Fig. 24, we can clearly notice that the use of Dempster-Shafer increases the detection probability around 20%. This result is expected since the Dempster-Shafer discounts the untrustworthy votes upon forming the final judgement which augments the accuracy of the decisions. By discarding the uncertain votes, the proposed model is increasing hence the number of detected selfish nodes.

False negative occurs when Intrusion Detection System fails to detect an actual attack. This value is increased whenever an existing attack is not detected. As shown in Fig. 25, the

"Without Dempster-Shafer" model allows some breaches to occur in this context. In fact, this scheme allows the selfish nodes to build alliances with some other nodes to gain the majority of votes. In contrary, the Dempster-Shafer model gives a zero percentage of false negatives. This is due to the fact that the trustworthiness value built through the payment mechanism affects the weight of each vote. This trust value gives an accurate assessment of the nodes since it is a result an accumulated incentive mechanism. 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 decision. This assures that all the selfish vehicles will be detected and hence the false negative percentage will be null.

Tit − for − Tat Cooperation Regulation : In this phase, the cooperation among the nodes is decided according to the aggregated decision; that is, a head or MPR node i will cooperate with another head or MPR node j if the belief in trustworthiness of node j is greater than 0.5. Otherwise, it will defect.

Algorithm 9 Cooperation Regulation Phase

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

procedure COOPERATIONREGULATION
    if belief(j) > 0.5 then
        i fulfills R
    else
        i drops R
    end if
end procedure
```

Information Dissemination : It's obvious that the implementation time and the overhead of the proposed model is somehow high since for each head and MPR there should

be a monitoring phase, voting phase, and aggregation phase before taking the cooperation or defection decision. To address this problem, information dissemination principle is employed to make neighbors share the belief in trustworthiness information of other nodes. Thus, the selfish nodes will be punished by all of their neighbors (who share the belief of this node) without having the monitoring, voting, and aggregation phases repeatedly. The information dissemination phase works as follows: after building the aggregated decision, each cluster-head propagates the results of the voting to its cluster members. Moreover, it has to broadcast these results 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 and voting procedures. This process allows decreasing the overhead and reducing the implementation time of the model.

Algorithm 10 Detection Algorithm - Information 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 INFORMATIONDISSEMINATION
    SelfishSet(H_1) := S
    if new contact between H_1 and H_2 occurs then
        SelfishSet(H_2) := SelfishSet(H_2) ∪ SelfishSet(H_1)
        SelfishSet(H_1) := SelfishSet(H_1) ∪ SelfishSet(H_2)
    end if
end procedure
```

Fig. 26, 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 seconds.

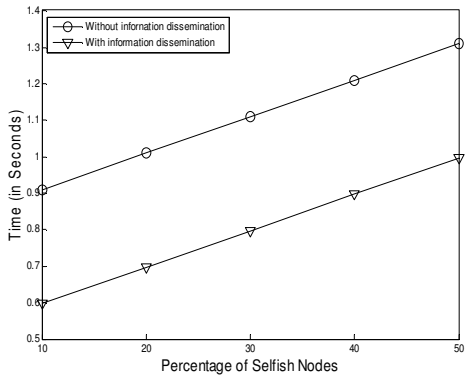

Figure 26: Model Implementation Time

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 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 $2N_i < 4N_i$.

Fig. 27 shows that the DS-based Tit-for-Tat strategy is as good as the generous strategy for the cooperative participants 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

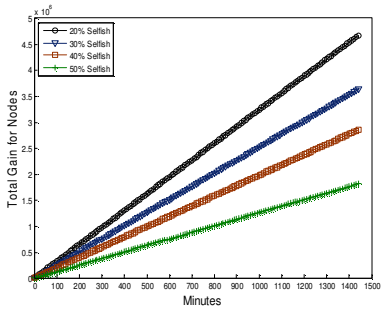

Figure 27: DS Tit-for-Tat

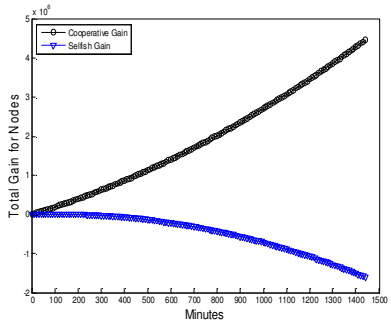

Figure 28: Selfish vs. Cooperative

high detection probability of the real selfish nodes and the null percentage of false alarms resulting from the use of Dempster-Shafer. Furthermore, Fig. 28 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 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.

4.3 Illustrative Example

Table 13: 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

According to the Bandwidth-Connectivity & Proportional Distance Model (BCDV)

Table 14: 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

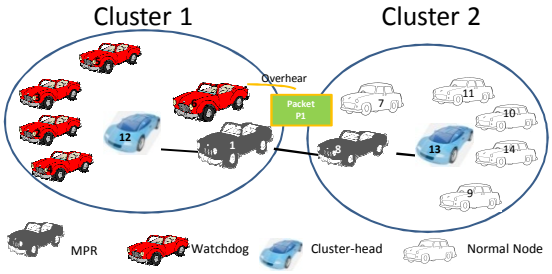

Figure 29: Vehicular ad-hoc Network example

(Table 3) which gives the best results in terms of number of MPRs, network stability, end-to-end delay, and packet delivery ratio as shown in Chapter 3, the Table 13 shows the QoS values calculated for each node. The initial reputation values of all the nodes are set to 100 as shown in Table 14. 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 payed. 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 a 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 finds 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 15.

Table 15: 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	$m_1(S)m_2(U) = 0$	$m_1(C)m_2(U) = 0$	$m_1(U)m_2(U) = 0$

Using Equations 4 and 5:

- 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.

- 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 5, K = (0.99)(0.99) + (0.01)(0.01) + (0.01)(0.99) = 0.9901.

- Using Equation 4, $m_1(S)m_2(S) = (0.99)(0.01)/[1 - 0.9901] = 1$.

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.

4.4 Conclusion

In this chapter, we addressed the problem of misbehaving nodes in Vehicular Ad Hoc Networks. We showed that the presence of these nodes have a negative impact on the network stability, lifetime, overhead, and delay. Therefore, we proposed a Dempster-Shafer base model that is able to (1) detect the misbehaving vehicles after the clusters are formed, and (2) regulate the cooperation. The detection is done in a cooperative manner where evidences from different watchdogs are gathered and aggregated using Dempster-Shafer. Thereafter, we proposed three strategies based on Tit-for-Tat to regulate the cooperation.

The strategies are: generous Tit-for-Tat, Tit-for-k-Tats, and Dempster-Shafer based Tit-for-Tat. Simulation results reveal that the Dempster-Shafer based strategy is the best to enforce the cooperation in the presence of high mobility and channel collisions. 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 regulate the cooperation inside the vehicular network by rewarding the cooperative nodes and punishing the selfish nodes. They show also that the model increases the probability of detection up to 40%, and annuls the false negatives while maintaining a minimized implementation time and overhead.

Chapter 5

Conclusion

Vehicular ad hoc network is an emerging wireless technology that is supposed to provide a wide set of safety and commercial applications and services. The main issue in such kind of networks is providing efficient and secure communications. Thus, we proposed in Chapter 3 a clustering model that organizes the network architecture followed by a routing model responsible for optimizing the communications. The clustering was done by proposing a QoS function that considers the mobility metrics such as velocity and distance and the QoS requirements such as bandwidth and connectivity. The routing was done by using an Ant Colony Optimization algorithm responsible for performing a proactive discovery to select the best paths in terms of stability, QoS, and delay. Performance analysis and simulation results prove that the proposed clustering and routing models are able to extend the network lifetime up to 12%, reduce the percentage of selected MPRs by 20%, increase the packet delivery ratio by 10%, and decrease the path length up to 2-hops. They show also that these models present an acceptable percentage of bandwidth average difference. However, some vehicles might misbehave after clusters formation by refusing to follow the models' rules.

Therefore, we developed in Chapter 4 the VANET-DSD model that detects the misbehaving vehicles, and regulates the cooperation. The detection is done in a cooperative manner where the evidences are collected by the different clusters nodes and the aggregation is accomplished using Dempster-Shafer. In the light of the detection results, the cooperation regulation phase employs the Tit-for-Tat strategy to reward the cooperative vehicles and punish the misbehaving ones. Simulations results show that the VANET-DSD model is able to regulate the cooperation by accurately rewarding the cooperative nodes and punishing the selfish nodes. They show also that the model increases the probability of detection up to 40%, and annuls the false negatives while maintaining a minimized implementation time and overhead.

In summary, the main contributions of our thesis are:

1. Optimizing the network architecture by forming homogeneous and long-living clusters.

2. Extending the vehicular network lifetime while maintaining the Quality of Service.

3. Improving the communications among clusters and reducing the overhead.

4. Detecting the misbehaving nodes after clusters formation.

5. Regulating the cooperation within the network.

The research presented in this thesis tackled the problem of clustering in Vehicular Ad Hoc Networks in the presence of passive malicious nodes. There are several research topics emerging from this work which should be continued.

Actually, the presence of active malicious nodes would hinder the application of the clustering models and would degrade the network performance considerably. These nodes that have the intention to harm the network may launch several attacks such as denial of service, and sybil attack.

Our future work will be detecting and punishing the active malicious nodes knowing that the detection of the active malicious nodes is more challenging than the passive ones since the former are more professional and have the intention to harm the network.

The following is the list of journal articles submissions derived from the thesis work:

- Omar Abdel Wahab, Hadi Otrok, Azzam Mourad. "A Dempster-Shafer based Tit-for-Tat Strategy to Regulate the Cooperation in VANET using QoS-OLSR Protocol", to the journal of Wireless Personal Communications, Springer, 2012 (Under Second Round of Minor Reviews).

- Omar Abdel Wahab, Hadi Otrok, Azzam Mourad. "VANET QoS-OLSR: QoS-based Clustering Protocol for Vehicular Ad Hoc Networks", to the journal of Computer Communications , Elsevier, 2012 (Under Second Round of Minor Reviews).

- Omar Abdel Wahab, Hadi Otrok, Azzam Mourad ." A Cooperative Watchdog Model Based on Dempster-Shafer for Detecting Misbehaving Vehicles", to the journal of Computer Communications , Elsevier, 2012 (Under Review).

Bibliography

[1] M. Artimy, J. Phillips, and W. Robertson, "Connectivity with Static Transmission Range in Vehicular Ad Hoc Networks," in *Proc. 3rd Annu. Comm. Networks and Services Research Conf.*, Washington, USA, 2005, pp. 237-242.

[2] H. Badis and K. A. Agha. (2005). QOLSR, QoS routing for ad hoc wireless networks using OLSR. *European Trans. on Telecommun.* [Online]. 16(5), pp. 427-442. Available: `http://dx.doi.org/10.1002/ett.1067`

[3] F. Bai, "QOLSR, IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols for Adhoc Networks," in *Proc. 22nd Annu. Joint Conf. of the IEEE Comput. and Comm. Societies*, Washington, USA, 2003, pp. 825-835.

[4] J. S. Baras and H. Mehta, "A probabilistic emergent routing algorithm for mobile ad Hoc networks," in *Proc. WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Sophia-Antipolis, France, March 3-5, 2003, pp. 1-10.

[5] M. Boban, G. Misek, and O. K. Tonguz, "What is the best achievable QoS for unicast routing in VANETs?," in *Proc. 2008 IEEE GLOBECOM Workshops*, New Orleans,

LO, USA, 2008, pp. 1-10.

[6] T. Clausen, P. Jacquet, P. Muhlethaler, A. Laouiti, A. Qayyum and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proc. IEEE Int. Multi Topic Conf. 2001* , Lahore, Pakistan 2001, pp. 62-68.

[7] Y. Hongseok and K. Dongkyun Kim. (2011). Repetition-based cooperative broadcasting for vehicular ad-hoc networks, *Comput. Commun.* [Online]. 34(15), pp. 1870-1882. Available: `http://dx.doi.org/10.1016/j.comcom.2011.05.007`

[8] J. T. Isaac, J. S. Camara, S. Zeadally, and J. T. Marquez. (2012). A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks. *Comput. Commun.* [Online]. 31(10), pp. 2478-2484. Available: `http://dx.doi.org/10.1016/j.comcom.2008.03.012`

[9] S. Yousefi, E. Altman, R. El-Azouzi and M. Fathy. (2008). Improving connectivity in vehicular ad hoc networks: An analytical study. *Comput. Commun.* [Online]. 31(9), pp. 1653-1659. Available: `http://dx.doi.org/10.1016/j.comcom.2007.10.045`

[10] G. Di Caro and M. Dorigo. (1998). AntNet: distributed stigmergetic control for communications networks. *J. of Artificial Intell. Research*. [Online]. 9(1), pp. 317-365. Available: `http://dl.acm.org/citation.cfm?id=1622797.1622806`

[11] G. A. Di Caro, F. Ducatelle and L. M. Gambardella, "Ant colony optimization for routing in mobile ad hoc networks in urban environments," Dalle Molle Inst. for Artificial Intell., Manno, Switzerland, Rep. May, 2008.

[12] M. Dorigo, G. Di Caro and L. M. Gambardella. (1998). Ant algorithms for discrete optimization. *Artificial Life* [Online]. 5(2), pp. 137-172. Available: http://dx.doi.org/10.1162/106454699568728

[13] M. Fiore, J. Harri, F. Filali and C. Bonnet, "Vehicular mobility simulation for VANETs," in *Proc. 40th Annu. Simulation Symp.*, Washington, DC, USA, 2007, pp. 301-309.

[14] M. Gunes and U. Sorges and I. Bouazizi, "ARA - The ant-colony based routing algorithm for MANETs," in *Proc. 2002 Int. Conf. on Parallel Process. Workshops*, Washington, DC, USA, 2002, pp. 79-85.

[15] S. Kuklinski and G. Wolny. (2009). Density based clustering algorithm for vehicular ad hoc networks. *Int. J. of Internet Protocol Technology* [Online]. 4(3), pp. 149-157. Available: http://dx.doi.org/10.1504/IJIPT.2009.028654

[16] S. Lim, C. Yu, and C. R. Das. (2012). Cache invalidation strategies for internet-based vehicular ad hoc networks. *Comput. Commun.* [Online]. 35(3), pp. 380-391. Available: http://dx.doi.org/10.1016/j.comcom.2007.02.020

[17] H. Otrok, A. Mourad, J. M. Robert, N. Moati and H. Sanadiki, "A cluster-based model for QoS-OLSR protocol," in *Proc. 7th Int. Wireless Commun. and Mobile Computing Conf.*, Istanbul, Turkey, 2011, pp. 1099-1104.

[18] K. R. Ramkumar, M. Ravichandran, N. Hemachandar, D. ManojPrasadh, and M. GaneshKumar, "RAAM: Routing algorithm using ant agents for MANETS," in *Proc.*

11th Int. Conf. on Advanced Commun. Technology, Gangwon-Do, Korea (South), 2009, pp. 1386-1391.

[19] K. R. Ramkumar, M. Ravichandran, N. Hemachandar, D. ManojPrasadh and M. GaneshKumar. (2009). SACOM: Secure ant colony optimization for MANETs. *Int. J. of Comput. and Elect. Eng.* [Online]. 1(2), pp. 165-170. Available: `http://www.docstoc.com/docs/47612327/SACOM-Secure-Ant-Colony-Optimization-for-MANETs`

[20] Z. Y. Rawashdeh and S. M. Mahmud, "Toward strongly connected clustering structure in vehicular ad hoc networks," in *Proc. 2009 IEEE 70th Veh. Technology Conf.*, Anchorage, USA, September 20-23, 2009, pp.16-21.

[21] C. Shea, B. Hassanabad and S. Valaee, "Mobility-based clustering in VANETs using affinity propagation," in *Proc. 28th IEEE Conf. on Global Telecommun.*, Piscataway, NJ, USA, 2009, pp. 4500-4505.

[22] S. Vodopivec, J. Bester and A. Kos, "A survey on clustering algorithms for vehicular ad-hoc networks," in *Proc. 35th Int. Conf. on Telecommun. and Signal Process.*, Prague, Czech Republic, July, 2012, pp. 52 - 56.

[23] G. Wolny, "Modified DMAC clustering algorithm for VANETs," in *Proc. 2008 Third Int. Conf. on Syst. and Networks Commun.*, Washington, DC, USA, 2008, pp. 268-273.

[24] Z. Zhang, A. Boukerche and R. Pazzi, "A novel multi-hop clustering scheme for vehicular ad-hoc networks," in *Proc. 9th ACM Int. Symp. on Mobility Manage. and Wireless Access*, Miami, Florida, USA, 2011, pp. 19-26.

[25] J. R. Douceur and T. Moscibroda, "Lottery trees: Motivational deployment of networked systems," in *Proc. 2007 Conf. on Applicat., Technologies, Architectures, and Protocols for Comput. Commun.*, Kyoto, Japan, 2007, pp. 121-132.

[26] S. B. Lee, G. Pan, J.-S. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. 8th ACM Int. Symp. on Mobile Ad Hoc Networking and Computing*, Montreal, Quebec, Canada, 2007, pp. 150-159.

[27] F. Li and J. Wu, "FRAME: An innovative incentive scheme in vehicular networks," in *Proc. 2009 IEEE Int. Conf. on Commun.*, Dresden, Germany, 2009, pp. 4638-4643.

[28] S. Zhong, Y. Yang, and J. Chen, "Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks," in *Proc. of IEEE INFOCOM*, San Francisco, USA, 2003, pp. 1987-1997.

[29] S. Buchegger and J. Y. Le Boudec, "Performance analysis of the CONFIDANT protocol," in *Proc. 3rd ACM Int. Symp. on Mobile Ad Hoc Networking & Computing*, Lausanne, Switzerland, 2002, pp. 226-236.

[30] Q. Lian, Y. Peng, M. Yang, Z. Zhang, Y. Dai, and X. Li. (2008). Robust incentives via multi-level Tit-for-Tat: Research articles. *Concurr. Comput. : Pract. Exper.* [Online]. 20(2), pp. 167-178. Available: `http://dx.doi.org/10.1002/cpe.v20:2`

[31] S. Marti, T.-J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. 6th Annu. Int. Conf. on Mobile Computing and Networking*, Boston, Massachusetts, USA, 2000, pp. 255-265.

[32] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proc. IFIP TC6/TC11 Sixth Joint Working Conf. on Commun. and Multimedia Security: Advanced Commun. and Multimedia Security*, Deventer, The Netherlands, 2002, pp. 107-121.

[33] L. Anderegg and S. Eidenbenz, "Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents," in *Proc. 9th Annu. Int. Conf. on Mobile Computing and Networking*, San Diego, CA, USA, 2003, pp. 245-259.

[34] D. B. Johnson and D. A. Maltz. (1996). Dynamic source routing in ad hoc wireless networks. *Mobile Computing*. [Online]. 353, pp. 153-181. Available: `http://citeseer.ist.psu.edu/johnson96dynamic.html`

[35] J. M. Robert, H. Otrok, A. N. Quttoum, and R. Boukhris. (2012). A distributed resource management model for virtual private networks: Tit-for-Tat strategies. *Computer Networks*. [Online]. 56(2), pp. 927-939. Available: `http://dx.doi.org/10.1016/j.comnet.2011.11.013`

[36] M. Felegyhazi, J. P. Hubaux, and Levente Buttyan. (2006). Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *IEEE Trans. Mobile Computing*. [Online]. 5(5), pp. 463-476. Available: `http://dx.doi.org/10.1109/TMC.2006.68`

[37] J. Andreoni, and H. Varian. (1999). Preplay contracting in the prisoners' dilemma. *PNAS*. [Online]. 96, pp. 10933-10938. Available: `http://www.sims.berkeley.edu`

[38] M. A. Nowak and K. Sigmund. (1994). The alternating prisoners dilemma. *J. of Theoretical Biology*. [Online]. 168(2), pp. 219-226. Available: http://www.sciencedirect.com/science/article/pii/S0022519384711015

[39] F. Anjum and P. Mouchtaris, "Intrusion detection systems," in *Security for Wireless Ad Hoc Networks*, J. S. Brake, Ed. New York: Wiley, 2007, pp. 149-156.

[40] J. Wu and R. Axelrod. (1995). How to cope with noise in the iterated prisoner's dilemma. *J. of Conflict Resolution*. [Online]. 39(1), pp. 183-189. Available: http://www.jstor.org/stable/174327

[41] F. Milan and J.Jaramillo and R. Srikant, "Performance analysis of reputation-based mechanisms for multi-hop wireless networks," in *Proc. 40th Annu. Conf. on Inform. Sci. and Syst.*, Princeton, NJ, USA, March, 2006, pp. 12-17.

[42] F. Milan, "Achieving cooperation in multihop wireless networks of selfish nodes," in *Proc. 2006 Workshop on Game theory for Commun. and Networks*, Pisa, Italy, 2006, pp. 131-141.

[43] M. A. Nowak and Karl Sigmund. (1992). Tit for tat in heterogeneous populations. *Nature*. [Online]. 355(6357), pp. 250-253. Available: http://dx.doi.org/10.1038/355250a0

[44] J. Schubert. (2011). Conflict management in Dempster-Shafer theory using the degree of falsity. *Int. J. Approx. Reasoning*. [Online]. 52(3), pp. 449-460. Available: http://dx.doi.org/10.1016/j.ijar.2010.10.004

[45] T. M. Chen and V. Venkataramanan. (2005). Dempster-Shafer theory for intrusion detection in ad hoc networks. *IEEE Internet Computing*. [Online]. 9(6), pp. 35-41. Available: http://dx.doi.org/10.1109/MIC.2005.123

[46] J. A. Weideman, and S. C. Reddy. (2000). A MATLAB differentiation matrix suite. *ACM Trans. on Math. Software*. [Online]. 26(4), pp. 465-519. Available: http://doi.acm.org/10.1145/365723.365727

[47] N. Mohammed, H. Otrok, L. Wang, M. Debbabi, and P. Bhattacharya. (2011). Mechanism design-based secure leader election model for intrusion detection in MANET. *IEEE Trans. Dependable and Secure Computing*. [Online]. 8(1), pp. 89-103. Available: http://dx.doi.org/10.1109/TDSC.2009.22

[48] J. M. Robert, H. Otrok, and A. Chriqi. (2012). RBC-OLSR: Reputation-based clustering OLSR protocol for wireless ad hoc networks. *Comput. Commun.* [Online]. 35(4), pp. 487-499. Available: http://dx.doi.org/10.1016/j.comcom.2011.11.013