VANET QoS-OLSR: QoS-based clustering protocol for Vehicular Ad hoc Networks

Article *in* Computer Communications · July 2013

DOI: 10.1016/j.comcom.2013.07.003

3 authors:

Omar Abdel Wahab
Université du Québec en Outaouais
51 PUBLICATIONS **721** CITATIONS

SEE PROFILE

Hadi Otrok
Khalifa University
160 PUBLICATIONS **2,001** CITATIONS

SEE PROFILE

Azzam Mourad
Lebanese American University
123 PUBLICATIONS **1,302** CITATIONS

SEE PROFILE

VANET QoS-OLSR: QoS-based Clustering Protocol for Vehicular Ad Hoc Networks

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

Abstract

In this paper, we address the problem of clustering in Vehicular Ad hoc Networks (VANETs) using Quality of Service Optimized Link State Routing (QoS-OLSR) protocol. 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 VANET safety, emergency, and multimedia services 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. 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 able to select alternatives and keep the network connected in case of link failures. Performance analysis and simulation results show that the proposed model can maintain the network stability, reduce the End-to-End delay, increase the packet delivery ratio, and reduce the communications overhead.

Keywords: Vehicular Ad hoc Network (VANET), Mobility, Quality of service (QoS), Stability, Ant Colony Optimization (ACO).

1. Introduction

Mobile Ad hoc Network (MANET) is a self-configuring network that connects the mobile nodes wirelessly. Vehicular Ad hoc Network (VANET) [10, 8, 17, 9] is a special kind of MANET that is characterized by a very high mobility. Hence, maintaining the stability in such kind of networks is a challenging task. In fact, the high

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

mobility of vehicles would shorten the network lifetime and cause link failures due to the frequent disconnections of clusters. Several clustering algorithm are presented for VANET such as [16, 22, 24]. However, these algorithms do not show how the routing is performed according to their clustering algorithms after the clusters formation. Hence, they do not guarantee the network topology during the routing process. Their clustering algorithms ignore as well the quality of service requirements important for safety, emergency, and multimedia services. On the other hand, QoS-based clustering algorithms take into consideration the quality of service metrics such as bandwidth, energy, and end-to-end delay to group the nodes. However, they ignore the high speed mobility metrics which makes them inefficient to deal with Vehicular Ad Hoc Networks. The *Optimized Link State Routing* (OLSR) [7] is a proactive routing protocol that 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. QoS-OLSR [18] is an enhanced version of OLSR that extends the MANET network lifetime taking into consideration the available bandwidth and the residual energy per node during cluster-heads election and MPR nodes selection. Nonetheless, this protocol does not consider the mobility of nodes while computing the QoS. Thus, nodes with high bandwidth, energy and mobility may be elected as cluster-heads which leads to recurrent disconnections. Likewise, the MPRs selected according to this protocol do not satisfy both mobility constraints and routing parameters (End-to-End delay and Packet Delivery Ratio). Moreover, the MPR selection algorithm according to QoS-OLSR is vulnerable to cheating in the sense that some nodes may claim bogus QoS values in order to ensure being selected as MPRs. Furthermore, QoS-OLSR does not advance any MPR recovery algorithm able to select quick alternatives and keep the network connected in case of link failures.

Based on this, QoS-OLSR protocol has the following limitations when used for VANET:

- QoS-OLSR neglects the mobility metrics while computing the QoS function.
- The MPR selection algorithm is unable to select the optimal set of MPRs in terms of stability, end-to-end delay, and packet delivery ratio since it ignores these parameters during the selection.
- Lack of MPR recovery mechanism in case of link failure.
- QoS-OLSR allows the nodes to cheat by claiming bogus QoS values during the MPRs selection.

To address the aforementioned shortcomings, we propose a new cluster-based protocol for VANET called *VANET QoS-OLSR*. The protocol is an extension of the QoS-OLSR that considers a tradeoff between Quality of Service requirements and mobility constraints and solves the limitations of QoS-OLSR that affect the network stability. It is composed of three components: (1) QoS-based clustering using Ant Colony Optimization, (2) MPR recovery algorithm, (3) and cheating prevention mechanism. First,

a QoS-based clustering algorithm is presented. This algorithm consists of electing cluster-heads and selecting MPRs with regard to the QoS and mobility constraints. The following metrics are considered to compute the QoS value per node: bandwidth, connectivity and mobility that includes both velocity and residual distance. 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. Based on these metrics, the cluster-head is elected according to the local maximal QoS value. Once elected, it is then responsible for selecting the set of MPR nodes responsible for transmitting the packets and connecting the clusters. This operation is done using an Ant Colony Optimization (ACO) derived algorithm that aims to reduce the end-to-end delay and increase the packet delivery ratio through a path guaranteeing the Quality of Service and mobility constraints. Nonetheless, some nodes having high mobility and low QoS metrics may claim bogus values to ensure being selected as MPRs. To guarantee the truth-telling and prevent the cheating during the selection procedure, we introduce a cheating prevention mechanism that consists of encrypting the QoS values during the selection. After being selected, some MPR nodes may cause link failures and break the stability of the network. Therefore, we introduce a MPR recovery algorithm that is able to select alternative MPR nodes with acceptable quality of service and mobility metrics able to keep the network connected and reduce the re-elections. Thus, we are able to maintain the stability of the network during the clusters formation, during the routing process, and in case of link failures while preserving the Quality of Service requirements.

In summary, our contribution is a novel QoS-based clustering protocol based on Ant Colony Optimization that is able to:

- Extend the network lifetime and maintain the QoS requirements by introducing a QoS-based clustering algorithm that considers the mobility metrics.
- Enhance the *End-to-End* delay and the *Packet Delivery Ratio* by selecting the MPR nodes using Ant Colony Optimization (ACO).
- Prevent the cheating during the MPR nodes selection using an encryption algorithm.
- Reduce the overhead by introducing a MPR recovery algorithm that is able to select alternative MPRs in case of link failures.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 formulates the problem. Section 4 explains the proposed protocol and describes its three components. Section 5 describes the packet format of the messages used by our protocol. Section 6 analyzes the performance of the proposed protocol and discusses the potential problems and solutions related to the cheating risk. Section 7 explains the model used for simulations and presents empirical results. Finally, Section 8 concludes the paper.

2. Related Work

Several clustering and routing algorithms have been advanced to cope with Mobile and Vehicular Ad Hoc Networks. In this section, we present the main clustering algorithms proposed for VANET. We present as well the main QoS-based clustering algorithms dedicated for MANET as well as the routing algorithms based on Ant Colony Optimization since our proposed protocol combines these two key concepts.

2.1. Clustering Algorithms for VANET

Modified DMAC [24] 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.

Affinity Propagation for Vehicular Networks (APROVE)[22] 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.

In [21], the authors propose a clustering approach that groups vehicles of similar mobility patterns in one cluster. The mobility pattern is represented in terms of speed and direction. The objective of this approach is to increase the stability and extend the lifetime of clusters.

The authors in [25] 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 cluster-head 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 [16], 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 [23]. 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.

2.2. QoS-based Clustering Protocols

The classical Optimized Link State Routing (*OLSR*) [7] 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 the Quality of Service Optimized Link State Routing (*QoS-OLSR*) [18], 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, the above stated protocols designed for MANET have different limitations that make them insufficient for VANET. First, the absence of the mobility constraints will affect the vehicular network stability. Second, the MPRs selection algorithm is based on a simple algorithm that does not consider neither the mobility nor the routing parameters (End-to-End delay and Packet Delivery Ratio). Third, the MPRs selection procedure is vulnerable to cheating which make it unfair. Fourth, these approaches do not advance any recovery algorithm to deal with link failures.

2.3. ACO-based Routing Algorithms

Routing Algorithm Using Ant Agents For MANETs (RAAM) [19] 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) [15] 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 [11] 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.

In gross, Ant Colony Optimization (ACO) [13] is a probabilistic mechanism that imitates the real behavior of ants seeking for food to find the optimal path. The main limitation of this technique in MANETs is the overhead caused by broadcasting the ant agent to the entire network. In this paper, we present a MPRs selection algorithm that is based on the Ant Colony Optimization. To the best of our knowledge, there is no work that exploits the ACO for the MPRs selection in a QoS-OLSR based protocol. Our proposed protocol assumes that the ant agents called *ANT-HELLO* are sent exclusively by the cluster-head and two-hop away at maximum in order to reduce the overhead.

3. Problem Statement

In this paper, 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 node 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 or QoS-OLSR for heads and MPRs selection, this node 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. Second, the link failures in VANET are likely to occur. Thus, launching a MPRs selection whenever a failure happens would lead to wide overhead due to the exchange of a large set of messages. Third, some nodes may cheat during the selection of MPRs by revealing bogus QoS values to guarantee being designated. This may lead to elect unreliable MPRs. Assume a node having a QoS value of 230 claims that its QoS is 530 after observing others values. This may lead to elect this node as MPR. However, this node might not have the required share of bandwidth or the reasonable mobility (speed and residual distance) values. This would lead to a link failure. Furthermore, this node may exploit its selection to realize malicious purposes such as flooding and Denial of Service.

Based on this, it is clear that the following objectives must be achieved to ensure the stability of the network. First, the clusters formation and the MPRs selection should take into consideration a tradeoff between the Quality of Service (bandwidth, End-to-End delay, Packet Deliver Ratio) and the mobility metrics (speed and residual distance). Second, there should be a MPR recovery algorithm able to provide quick alternatives and avoid the frequent re-elections in case of link failures. Third, a cheating prevention mechanism should be applied to forbid the nodes with low QoS values and high mobility from being selected as MPRs. To achieve these goals, we propose VANET QoS-OLSR protocol that is composed of three components: (1) QoS-based clustering using Ant Colony Optimization, (2) MPR recovery algorithm, (3) and cheating prevention mechanism. The details of this protocol are discussed in the section 4.

4. 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 three components: the QoS-based Clustering, the cheating prevention, 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 according to a cheat-proof procedure. Finally, the MPR recovery algorithm deals with link failures by selecting alternative MPRs.

Table 1: Notations

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

4.1. 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.

4.1.1. 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 this paper, 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 2.

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 2. 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 $i \in N$ **do**
        Velocity(i):= random integer between Min and Max 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 $i \in N$ **do**
        CurrentPosition(i): the current position of i
        ResidualDistance(i):=MaximumDistance-CurrentPosition(i)
        DistanceRatio(i):=ResidualDistance(i)/MaximumDistance
    end for
end procedure

4.1.2. 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, 6] 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

Table 2: 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) = $BW(i) \times N(i) \times \frac{DistRatio(i)}{VelRatio(i)}$;

network-dedicated factors such as bandwidth and connectivity ensures to have a stable and reliable Vehicular Ad Hoc Network.

4.1.3. 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 (Fig. 3) 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 (Fig. 5) 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 Topology Control (*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 (Fig. 4) are used by the nodes to indicate the neighbors for which node this neighbor has voted for. Section 5 explains in details the format of these messages.

Algorithm 3: Cluster Head Election Algorithm

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

4.1.4. Ant Colony Optimization Basic Notations

Ant Colony Optimization [13] 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 paper, 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 [12]: $\tau_i = \lambda \times \tau_i + (1-\lambda) \times q_i$ 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.

4.1.5. 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 (Fig. 6).

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 (Fig. 6) to 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 (Fig. 6) 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.

4.2. Cheating Prevention

In order to guarantee a reliable and fair MPRs selection procedure, the cheating risk should be considered. In fact, some nodes may receive the *ANT-HELLO* message and notice that some other vehicles have QoS values that are higher than theirs. For this reason, these nodes may cheat by revealing exaggerated QoS values in a way to ensure them being selected as MPRs. Therefore, the QoS values should be somehow hidden. Consequently, we propose an encryption mechanism to be applied during the elections. The mechanism works as follows. After being elected as a cluster-head, each head node must propagate a message called *Ack* (Fig.

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 $Pheromone(i) := \quad QoS(i) - D(i)$
      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

5) containing its public key 2-hop away. This key is used during the elections by the intermediate nodes to encrypt their QoS values using the destination head's key $ENCRYPT\{QoS\ value, destination\ public\ key\}$. Thus, each node's QoS value is protected from further interception and exploitation since no other node than the destination cluster-head can decrypt these values. Furthermore, upon launching the MPRs selection procedure, the cluster-heads include their public keys in the *ANT-HELLO* (Fig. 6) messages. These messages are then propagated two-hop away until reaching the cluster-head destination. Once reached, the destination cluster-head receives the *ANT-HELLO* messages which contain the source cluster-head's public key. It decrypts then the encrypted QoS values using its private key (since they are encrypted using its public key) $DECRYPT\{QoS\ value, destination\ private\ key\}$, extracts the values and updates the pheromone flag accordingly. Finally, it encrypts back the QoS values using the source head's public key received from the *ANT-HELLO* messages and sends back these messages to the source head, which in its turn selects its set of optimal MPR nodes.

4.3. MPR Recovery Algorithm

Link failures represent a big challenge to the stability of the vehicular network. Fig. 1 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.

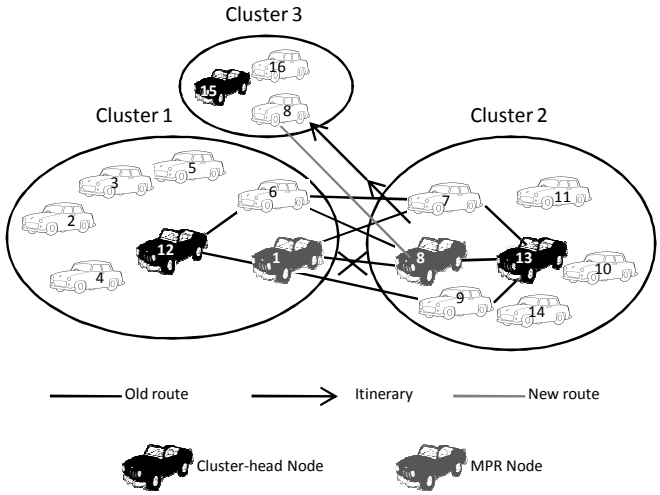

Figure 1: Link failure example: Node 8 serving as MPR between Cluster 1 and Cluster 2 decides to leave its current cluster and join Cluster 3

- 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.

Algorithm 5: MPR Recovery Algorithm

```
1:  procedure MPRRECOVERY
2:      for each cluster-head k do
3:          Sort the "Nodes Visited Stack" s
4:          if(TCmsgNotRcvdTime(n) > TimeAllowedForTC()) then
5:              s := s − {n}
6:              MPRset(k) := i/i ∈ s(1)
7:              if(isEmpty(s)) then
8:                  MPRSelectionAlgorithm()
9:              end if
10:         end if
11:     end for
12: end procedure
```

4.4. Illustrative Example

To illustrate how VANET QoS-OLSR works, we present a concrete example. Fig. 2 shows a network with fourteen nodes and six possible paths. Table 4 gives the pheromone value and the relevant probability of each path using the MPRs selection algorithm (refer to Algorithm 4), while Table 3 shows the QoS metrics value

Table 3: 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

Table 4: 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

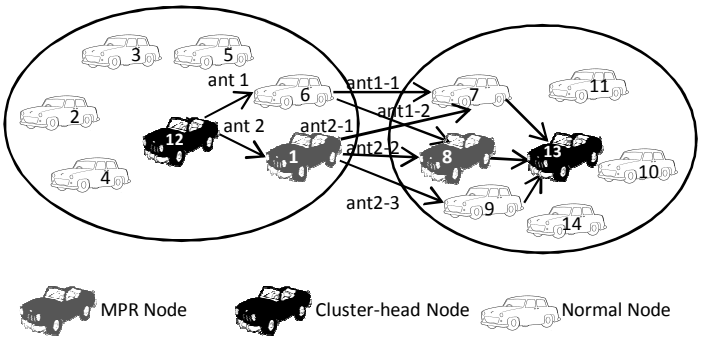

Figure 2: Vehicular Ad Hoc Network example: A network of 14 nodes needs to form clusters by electing cluster-heads and connect the clusters by selecting MPR nodes

and the pheromone value for each node according to the BCDV model (Table 2). 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 2). 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. The source head CH-1 first sends 2 (according to the number of its 1-hop away neighbors) forward *ANT-HELLO*

messages (Fig. 6) 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 4). 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.

5. Packet Format

In this section, we present the format of the messages needed by the cluster-head election and the MPRs selection algorithms.

5.1. Cluster-head Election Messages

The cluster-head election algorithm relies on modified *HELLO*, *Election*, and *Acknowledgement* messages.

5.1.1. HELLO Messages

Some modifications are done on the original *HELLO* message. The modifications are presented below:

- *Reserved field:* The first three bits are used to encode the version number of this extension (001).
- *Willingness:* This field maintains the QoS value of the issuing node.
- *Htime:* This field specifies the time interval between two emitted *HELLO* messages.
- *Link Code:* This field is split up into two subfields *Neighbor Type* (2-bit field) and *Link Type*. Another neighbor type is also added. H_NEIGH signals that a neighbor has been elected as a cluster head.

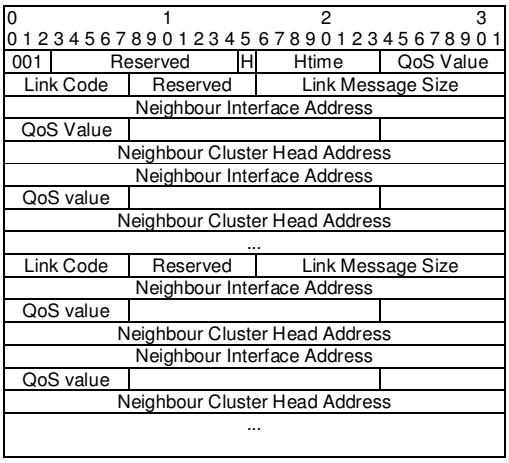

Figure 3: *HELLO* message format: This message is used by the nodes to find their 1-hop and 2-hop neighbors and to propagate their QoS values

5.1.2. Election Messages

After the exchange of *HELLO* messages, each node votes for the neighbor having the highest QoS value. It can vote for itself if it has this value. The election procedure is achieved using an *Election message*. In fact, the nodes use this message to broadcast locally their votes. This message is straightforward. It indicates the identifier (ID) of the voted cluster head by the issuing node with its QoS value. This message has the same format as the *HELLO* message. The message format is illustrated in Fig. 4.

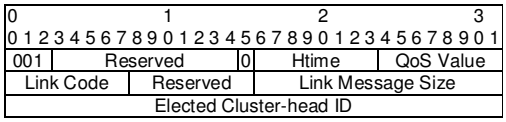

Figure 4: *Election* message format: This message is used to propagate the votes during elections

5.1.3. Acknowledgement Messages

The *Ack* message is sent by the cluster-head once elected. Each head sends it 2-hop away to signal that it acknowledges serving as a cluster-head for its electors. This message is derived from the original *HELLO* message which makes them share some common fields such as: *Htime*, *Link Code* and *Link Message Size*. The *H* flag is set to 1 in this message indicating that it is sent by a cluster-head. It contains in addition a new flag, the *Head Public Key* flag, used to propagate the public key of each cluster-head to its 2-hop away nodes. These nodes can later encrypt their QoS values during the MPRs selection process allowing only the entitled cluster-heads to decrypt and check them.This mechanism is used to prevent the cheating represented by claiming bogus

QoS values in order to guarantee being selected as MPRs. The structure of this message is illustrated in Fig. 5.

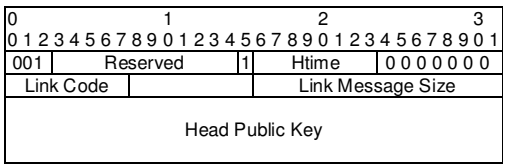

Figure 5: *Ack* message format: This message is used by the cluster-head to acknowledge serving its voters and to broadcast its public key used for cheating prevention

5.2. MPRs Selection Messages

The MPRs selection algorithm relies on one new specialized *HELLO* message called *ANT-HELLO* and one *TC* message. The next subsections explain these messages and describe their usages.

5.2.1. ANT-HELLO Messages

As shown in Fig. 6, the *ANT-HELLO* messages are an extended version of *HELLO* messages. The modifications are the following:

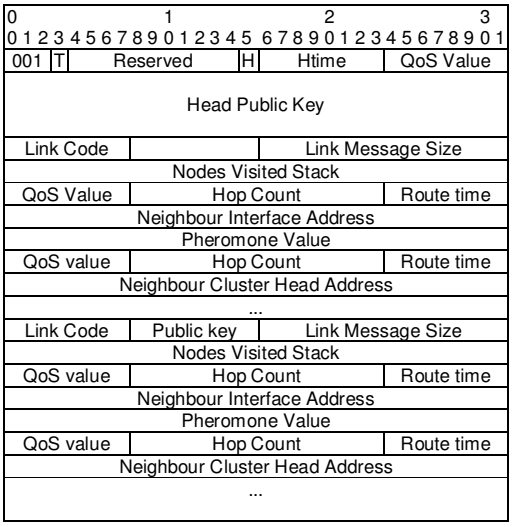

Figure 6: *ANT _HELLO* message format: This message is used to collect the paths information during MPRs selection

- *T:* indicates the message type. It is 0 for forward ants and 1 for backward ants.

- *Hop Count:* This field is incremented by the *forward ANT-HELLO* messages when they are about to move to the next node. It is used to signal the number of intermediate nodes visited by the ant messages.
- *Nodes Visited Stack:* This stack maintains the path of the *forward ANT-HELLO* messages when it gets the destination. *ANT-HELLO* uses then this stack to track back to the source.
- *Public key:* This field contains the public key of the originator cluster-head. It is represented in terms of 512 bits.

5.2.2. TC Messages

We made a slight modification on the *Topology Control (TC)* message. In fact, the first three bits of the Reserved field are used to encode the version number of this extension (001). Additionally, the message incorporates the QoS value of the issuing node. This value is used to calculate the optimal MPRs chosen to transmit the packets. The message format is illustrated in Fig.7.

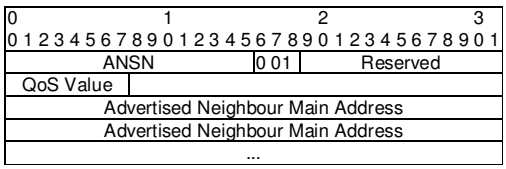

Figure 7: *TC* message format: This message is used by the MPRs to propagate neighbor information over the network

6. Performance Analysis and Discussions

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. We discuss as well the cheating risk in terms of problem, solution, and future work.

6.1. 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 [20], AntHocNet [11], ARA [15], and PERA [4], the sender node has to broadcast the *ant* packet many hops away which causes a wide overhead over the network.

6.2. 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.

6.3. 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.

6.4. 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.

6.5. 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.

6.6. 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.

6.7. Cheating Risk Discussion

In order to guarantee a reliable and fair MPRs selection procedure, the cheating risk should be considered. In fact, some nodes may receive the ANT-HELLO message, used to propagate the QoS values, during elections and notice that some other vehicles have QoS values that are higher than their own values. For this reason, these nodes may cheat by revealing exaggerated QoS values in a way to ensure them being selected as MPRs. Therefore, the QoS values should be somehow hidden. Consequently, we design a MPR selection procedure of three rounds. In the go phase, each node encrypts its QoS using the head destination public key so that only the latter can decrypt it. Then, nodes update the *ANT-HELLO* message with the encrypted value. Thus, each node is preserving its QoS value from being observed and exploited by malicious vehicles. In the back phase, upon receiving the encrypted values from the nodes, the destination cluster-head decrypts these values using its private key. It calculates then the pheromone value of each path and selects the nodes belonging to the path having the highest probability of pheromone and located within its cluster limits as MPRs. Thereafter, it re-encrypts the QoS values using the source head public key received from the *ANT-HELLO* message and sends back the messages towards the source via the chosen optimal path. The source head node can, in its turn, select the nodes belonging to the optimal path and existing within its cluster as MPRs.

To perform such a mechanism, a combination of TESLA [26] and Public Key Infrastructure (PKI) [27] can be used as a possible solution where these two techniques have proved to be lightweight when used to MANET. In fact, recent investigations showed that computationally limited mobile nodes such as vehicles in VANET can perform public key operations. Thus, since our algorithm involves more verification than signing, the vehicles can verify a signature in $0.43s$ using the PKI technique [27]. Note that the encryption/decryption mechanism is done only during MPRs selections where the mission of the cluster-heads is to verify the encrypted QoS values not to sign. This makes our process lightweight for the nodes and especially for the cluster-heads even in dense networks.

Furthermore, the use of TESLA and PKI for cheating prevention can achieve three main security properties: integrity, authentication, and freshness. In fact, the use of TESLA and PKI protocols allows the messages to be signed by the source nodes (intermediate nodes) and verified by others (cluster-heads). Thus, the integrity is ensured and the possibility of modifying the QoS values is prevented. Besides, the PKI allows the recipient of a message to verify the identity of the sender through its signature, which achieves the source authentication property. Moreover, the TESLA protocol allows the synchronization among the vehicles' clocks, which avoids the message replay attacks. Thus, the freshness property is guaranteed. Note that with TESLA, loosely synchronized clocks are available.

7. Simulation Results

This section is divided into two parts. The first part presents the results after comparing our five proposed models (available in Table 2) 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.

7.1. Simulation Scenario and Parameters

In order to compare the different models, we resorted to the use of MATLAB [28] network simulator with the VanetMobiSim [14] traffic simulator. VanetMobiSim is a traffic simulator that employs XML code to represent the 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 refers to road topology namely the number of lanes, the traffic light constraints, speed limits, etc. Micro mobility is concerned more by driving behavior [14]. We parse then the TIGER file to take the information related to the road topology. A simulation area of 3000×1000m is used to simulate a set of nodes varying from 30 to 100. A screenshot of this area is presented in Fig. 8. The highway topology is exploited to simulate the traffic **since the most of works dedicated to VANET use this topology to evaluate the performance of their models [30, 29, 22]**. The velocity bounds on this highway range from 60 km/h to 120 km/h. The transmission ranges used for the simulations vary from 150 to 300. Each simulation round lasted 420 seconds after 30 seconds of the initial, excluding the movement of the nodes. **The log-normal shadowing model has been used as a propagation model. In this model, the signal strength perceived by a certain node is affected not only by the distance between the transmitter and the receiver, but also by some other random factors. In fact, the log-normal shadowing radio propagation takes into account that the antennas are not perfectly isotropic, and, even more importantly, the environment might be obstructed by, e.g., buildings or trees. The parameters of this model are set as follows:**

- **Path loss exponent: 5 (this parameter describes the environment decay rate).**
- **Shadowing Deviation: 6 (this parameter describes the variation due to obstacles).**

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...) and we calculate the confidence interval using the mean and standard deviation to know the number of simulation runs that are able to yield results within this interval. Experiment results show that running 100 independent simulation runs is able to provide results within the confidence interval. The simulation parameters are summarized in Table 5.

Table 5: Simulation Parameters

Parameter	Value
Clustering Protocols	VANET QoS-OLSR, QoS-OLSR, and Classical QOLSR
Number of nodes	$30, 40, 50, 60, 70, 80$, and 100
Transmission range	$150, 200$, and 300 m
Topology	Highway
Packet Size	1 kb
Idle Time	Random value in $[0..1]$
Link Bandwidth	2Mbps
Available Bandwidth	*Idle Time* $\times$ *Link Bandwidth*
Hello messages	18 messages are sent per minute
Radio Propagation Model	**Log-normal Shadowing Model**
Minimum Speed	60 km/h
Maximum Speed	120 km/h
Number of simulation runs	100 **(95% of confidence level)**

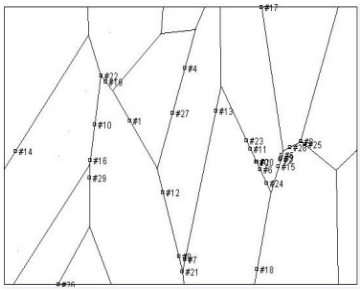

Figure 8: Graph of streets used in our vehicular movement simulations.

7.2. Comparison Between Our Proposed Models

In this part, we present a comparison between our proposed models presented in Table 2 in order to find the best model that will be compared with the other approaches. In terms of MPRs, Fig. 9 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

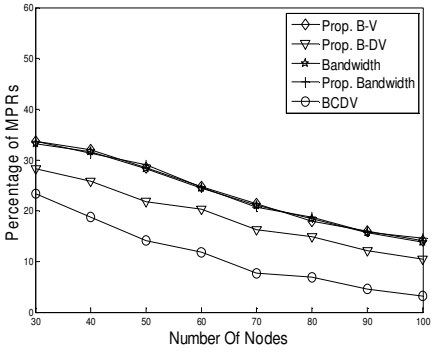

Figure 9: **Percentage of MPR nodes: The BCDV model is able to decrease the percentage of MPRs by considering the connectivity factor that is able to increase the coverage of the MPRs**

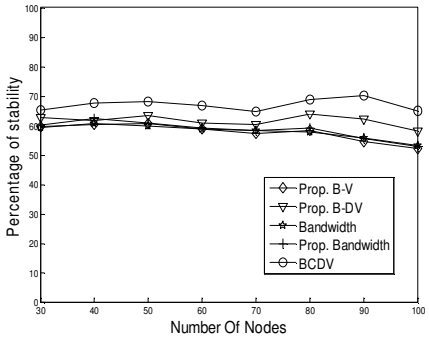

Figure 10: **Percentage of stability: The BCDV model is able to increase the percentage of stability by considering the residual distance and velocity that can decrease the clusters' disconnections**

most of other functions (refer to Table 2). Concerning the clusters stability, which depends mainly on the distance and velocity factors, Fig. 10 shows that BCDV gives an improved percentage of stability compared to the other models. **Note that the percentage of stability increases when the number of nodes reaches 90. This is due to the fact that such number of vehicles is able to form stable clusters as a result of their connectivity level, which depends on the initial positions generated randomly by the simulator**. The average number of hops between the source and destination is also reduced with this model according to Fig. 11 which reduces the end-to-end delay. Similarly, the packet delivery ratio is increased using BCDV model as depicted in Fig.

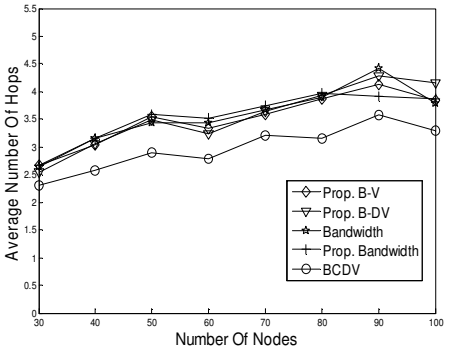

Figure 11: **Average number of hops: The BCDV model is able to decrease the average number of hops by using Ant Colony Optimization for MPRs selection and considering the route time when calculating the pheromone**

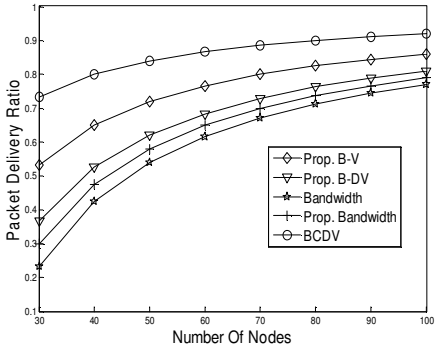

Figure 12: **Packet Delivery Ratio: The BCDV model is able to increase the packet delivery ratio by increasing the stability and using Ant Colony Optimization for MPRs selection**

12.

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 6 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.

Table 6: Bandwidth Average Difference between our models

Models	Transmission Ranges		
	150	200	300
Bandwidth	0%	0%	0%
Proportional bandwidth	6.77%	3.85%	3.15%
Proportional B-V	7.59%	4.21%	4.56%
Proportional B-DV	6.63%	5.11%	3.76%
BCDV	7.08%	4.3%	3.94%

7.3. 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

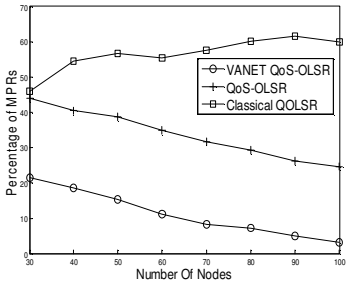

Figure 13: Percentage of MPR nodes

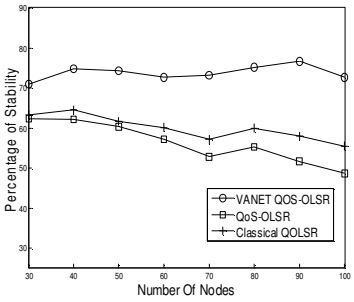

Figure 14: Percentage of stability

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 itself. Fig. 13 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

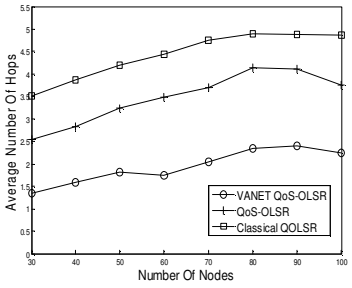

Figure 15: Average number of hops

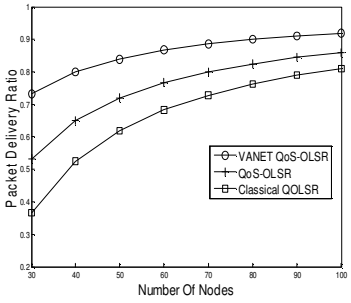

Figure 16: Packet Delivery Ratio

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. 14 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. 15 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. 16 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 7: Bandwidth Average Difference

Models	Transmission Ranges		
	150	200	300
Classical QOLSR	0%	0%	0%
QoS-OLSR	7.04%	4.58%	3.90%
VANET QoS-OLSR	7.08%	4.3%	3.94%

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 7 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.

8. Conclusion

In this paper, we proposed VANET QoS-OLSR protocol that aims at maintaining the stability of the vehicular network while achieving the Quality of Service require-

ments. The protocol is composed of three components: (1) QoS-based clustering using Ant Colony Optimization, (2) MPR recovery algorithm, and (3) cheating prevention mechanism. 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. In order to guarantee a fair and reliable selection procedure, a cheating prevention mechanism is presented. Finally, a MPR recovery algorithm is introduced to select alternative MPRs and keep the network connected in case of link failures. Performance analysis and 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.

References

[1] M. Artimy, J. Phillips and W. Robertson. Connectivity with Static Transmission Range in Vehicular Ad Hoc Networks. In *Proceedings of the 3rd Annual Communication Networks and Services Research Conference*, pages 237-242. IEEE Computer Society, 2005.

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

[3] F. Bai. QOLSR, IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols for Adhoc Networks. In *Proceedings of INFOCOM*, pages 825-835. IEEE INFOCOM, 2003.

[4] J.-S. Baras and H. Mehta. A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks. In *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[5] R. Barr, Z.-J. Haas and R. van Renesse. JiST: An efficient approach to simulation using virtual machines. *Software: Practice and Experience*, 35(6):539-576, 2005.

[6] M. Boban, G. Misek, O.-K. Tonguz. What is the Best Achievable QoS for Unicast Routing in VANET?. In *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1-10, 2003

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

[8] Y. Hongseok and K. Dongkyun Kim. Repetition-based Cooperative Broadcasting for Vehicular Ad-Hoc Networks. *Computer Communications*, 34(15):1870-1882, 2011.

[9] J.-T. Isaac, J.-S. Camara, S. Zeadally, J.-T. Marquez. A Secure Vehicle-to-Roadside Communication Payment Protocol in Vehicular Ad Hoc Networks. *Computer Communications*, 31(10):2478-2484, 2012.

[10] S. Yousefi, E. Altman, R. El-Azouzi and M. Fathy. Improving Connectivity in vehicular ad hoc networks: An analytical study. *Computer Communications*, 31(9):1653-1659, 2008.

[11] G. Di Caro and M. Dorigo. AntNet: distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9(1):317-365, 1998.

[12] G. A. Di Caro, F. Ducatelle and L. M. Gambardella (2008). *Ant Colony Optimization for Routing in Mobile Ad Hoc Networks in Urban Environments*. IDSIA: Lugano (Switzerland). 2008:05-08.

[13] M. Dorigo, G. Di Caro and L.-M. Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2):137-172, 1999.

[14] M. Fiore, J. Harri, F. Filali and C. Bonnet. Vehicular Mobility Simulation for VANETs. In *Proceedings of the 40th Annual Simulation Symposium*, pages 301-309. IEEE Computer Society, 2007.

[15] M. Gunes and U. Sorges and I. Bouazizi. ARA - The Ant-Colony Based Routing Algorithm for MANETs. In *Proceedings of the 2002 International Conference on Parallel Processing Workshops*, pages 79-85. IEEE Computer Society, 2002.

[16] S. Kuklinski and G. Wolny. Density Based Clustering algorithm for Vehicular Ad Hoc Networks. *International Journal of Internet Protocol Technology*, 4(3):149-157, 2009.

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

[18] H. Otrok, A. Mourad, J.-M. Robert, N. Moati and H. Sanadiki. A Cluster-Based Model for QoS-OLSR Protocol. *IWCMC*, pages 1099-1104. IEEE, 2011.

[19] K.-R. Ramkumar, M. Ravichandran, N. Hemachandar, D. ManojPrasadh and M. GaneshKumar. RAAM: Routing Algorithm Using Ant Agents for MANETS. In *Proceedings of the* 11^{th} *international conference on Advanced Communication Technology*, pages 1386-1391. IEEE Press, 2009.

[20] K.-R. Ramkumar, M. Ravichandran, N. Hemachandar, D. ManojPrasadh and M. GaneshKumar. SACOM: Secure Ant Colony Optimization for MANETs. *International Journal of Computer and Electrical Engineering*, 1(2):165-170, 2009.

[21] Z.-Y. Rawashdeh and S.-M. Mahmud. Toward Strongly Connected Clustering Structure in Vehicular Ad hoc Networks. In *Proceedings of the 2009 IEEE 70th Vehicular Technology Conference: VTC2009-Fall*, pages 1-5. IEEE Press, 2009.

[22] C. Shea, B. Hassanabad and S. Valaee. Mobility-based Clustering in VANETs Using Affinity Propagation. In *Proceedings of the 28th IEEE conference on Global telecommunications*, pages 268-273. IEEE Press, 2009.

[23] S. Vodopivec, J. Bester and A. Kos. A survey on clustering algorithms for vehicular ad-hoc networks. In *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*, pages 52-56. IEEE Press, 2012.

[24] G. Wolny. Modified DMAC Clustering Algorithm for VANETs. In *Proceedings of the 2008 Third International Conference on Systems and Networks Communications*, pages 4500-4505. IEEE Computer Society, 2008.

[25] Z. Zhang, A. Boukerche and R. Pazzi. A novel multi-hop clustering scheme for vehicular ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Mobility management and wireless access*, pages 19-26. ACM, 2011.

[26] A. Perrig, R. Canetti, D. Tygar, and D. Song. The TESLA broadcast authentication protocol. *RSA Cryptobytes*, 5(2):2-13, 2002.

[27] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Proceedings of the Cryptographic Hardware and Embedded Systems (CHES)*, 2004.

[28] J. A. Weideman, and S. C. Reddy. A MATLAB Differentiation Matrix Suite. *ACM Trans. on Math. Software*, 26(4):465-519, 2000.

[29] H. Su. Clustering-Based Multichannel MAC Protocols for QoS Provisionings Over Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, 56(6):3309-3323, 2007.

[30] F. Kaisser, C. Johnen, and V. Vque. Quantitative Model for Evaluate Routing Protocols in a Vehicular Ad Hoc Networks on Highway. In *Proceedings of the IEEE Vehicular Networking Conference*, 2010.