

Research Article

A Multiconstrained QoS-Compliant Routing Scheme for Highway-Based Vehicular Networks

Abderrahmane Lakas ¹, Mohamed El Amine Fekair,²
Ahmed Korichi ² and Nasreddine Lagraa³

¹CIT, United Arab Emirates University, P.O. Box 17551, Al Ain, UAE

²Computer Science Department, University of Ouargla, Ouargla, Algeria

³LIM, University of Laghouat, Laghouat, Algeria

Correspondence should be addressed to Abderrahmane Lakas; alakas@uaeu.ac.ae

Received 31 August 2018; Revised 13 November 2018; Accepted 2 January 2019; Published 4 February 2019

Academic Editor: Laurie Cuthbert

Copyright © 2019 Abderrahmane Lakas et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the deployment of multimedia services over VANETs, there is a need to develop new techniques to insure various levels of quality of services (QoS) for real time applications. However, in such environments, it is not an easy task to determine adequate routes to transmit data with specific application QoS requirements. In this paper, we propose CBQoS-Vanet, a new QoS-based routing protocol tailored towards vehicular networks in a highway scenario. This protocol is based on the use of two techniques: first a clustering technique which organizes and optimizes the exchange of routing information and, second, a bee colony inspired algorithm, which calculates the best routes from a source to a destination based on given QoS criteria. In our approach, clusters are formed around cluster heads that are themselves elected based on QoS considerations. The QoS criteria here are based on the two categories of metrics: QoS metrics and mobility metrics. The QoS metrics consists of the available bandwidth, the end-to-end delay, and the jitter. The mobility metrics consists of link expiration time and average velocity difference. We have studied the performance of CBQoS-Vanet through simulation and compared it to existing approaches. The results that we obtained show that our technique outperforms, in many aspects, the approaches that it was compared against.

1. Introduction

In recent years, vehicular ad hoc networks (VANETs) have attracted the interest of researchers and car manufacturers for their benefits to the deployment of Intelligent Transportation Systems (ITS). A VANET is a network of interconnected vehicles that are able to communicate with each other and exchange relevant information. One of the main objective of VANETs is to expand the reach to Internet by commuters and to access transportation-related services while on the road. But it is the road safety and travel comfort applications that add value to vehicular networks [1]. However, with the deployment of multimedia and real time applications over VANETs, there is a need not only to route information between vehicles in an optimal way but also to select routes that are adequate for Quality of Service (QoS) requirements for these applications. QoS requirements typically include

bandwidth, end-to-end delay, and jitter; but other criteria that are inherent to the nature of vehicular networks can be considered as well. In VANET where the network topology is highly dynamic, the connectivity between vehicles is constantly changing. Therefore, we should also consider the stability of routes and the life expectancy of links as a QoS metric. Providing support for QoS in such a highly dynamic environment can be a very challenging optimization problem similarly to Multiconstrained Optimal Path problems which have been established as NP-complete [2].

To solve this problem, several swarm intelligence and metaheuristic strategies have been considered. Among these strategies, the use of bioinspired artificial bee colony (ABC) technique introduced in [3] has been proposed. ABC technique is based on the biological paradigm of communication between bees when searching for food and has been shown to be an efficient tool for solving multiconstrained routing

problems in static network topologies [4]. For the case of VANETs, any optimization strategy needs to take into consideration the instability of the network topology. That is, a clustering is the approach that is most suitable for enhancing the stability of the network [5], especially in the scenario of highways. Cluster formation allows the vehicular network to adapt to the changing topology of the network and to the speed of the vehicles, therefore, allowing routing protocols to find stable paths while requiring little overhead of the network.

In this paper, we present CBQoS-Vanet, a QoS-based routing protocol over VANET. This research is an extension of a previous work presented in [6]. CBQoS-Vanet protocol is based on clustering and ABC techniques. The clustering algorithm is used to organize and optimize the exchange of routing information between clusters. Each cluster is formed around a cluster head, which is itself elected based on QoS considerations. That is, the routing structure itself is here tailored toward moving data based on QoS. The ABC algorithm is used here to find the best route among many routes between a source and a destination. The selection of the best route is the route that complies the most with the application's QoS requirements. ABC operation is inspired by the behavior of bees in their search for food and by their strategy for scouting routes from the hive to various food locations. CBQoS-Vanet uses the concept of *forward* and *backward* scouting, which serves as the process for discovering the final destination. Although CBQoS-Vanet is a reactive routing protocol, it is, however, equipped with a caching scheme that allows routes to be proactively discovered without completing the full scouting process. In this study, we consider the case of a highway scenario with two categories of QoS: QoS metrics and mobility metrics. The QoS metrics consists of the available bandwidth, the end-to-end delay, and the jitter. The mobility metrics consists of link expiration time and average velocity difference.

The remainder of this paper is organized as follows: in Section 2, we provide a review of related works. In Section 3, we present a description of CBQoS model and mathematical formulation for the routing problem we consider. In Section 4, we present a detailed description of CBQoS-Vanet protocol. In Section 5 we provide a detailed performance study with an analysis of the performance results. Finally, we draw our conclusions in Section 6.

2. Related Work

Many research studies have investigated the problem of QoS-based routing in general, but only fewer have been conducted on QoS-based routing in the context of vehicular networks. In this section we provide a review of the main existing work done on QoS-based routing that adopted the use of clustering for routing information between vehicles and those which used bioinspired techniques for the solving the problem of routing optimization.

2.1. Cluster-Based QoS Routing for VANET. The literature includes many research contributions proposing cluster-based techniques for QoS routing. In [7], the authors propose a routing protocol called VANET QoS-OLSR using both a

clustering approach and an optimization technique based on ant colony. In this approach, the cluster head (CH) is selected based on two parameters: the highest QoS parameter and the node mobility parameter. The CH uses a *Hello* message to discover the network and selects a multipoint relay (MPR) to form a stable cluster that could contribute to the stability of the path. VANET QoS-OLSR showed that it can reduce link failures by choosing alternative MPRs. However, as is the case for the simple OLSR, VANET QoS-OLSR may not be suitable for VANETs given their highly dynamic nature and that the formed clusters have often outdated network state information.

In [8], the authors propose a Passive Clustering Aided Routing (PassCAR) protocol, destined to enhance the routing performance in one-way multilane highway scenario. PassCAR includes route discovery, route establishment, and data transmission phases. It selects suitable nodes for the role of cluster heads by using multimetric election approach based on metrics such as node degree, estimated transmission count, and link expiration time. PassCAR use a broadcasting strategy to discover paths, therefore generating a high routing overhead.

In [9], the authors propose a distributed cluster based multichannel MAC communications (CBMMAC) scheme for QoS provisioning over VANET. The protocol organizes the vehicles in close proximity with each other and traveling in the same direction into clusters. Cluster members, whether cluster head or simple member, are equipped with two transceivers, which can operate simultaneously on two separate channels: SCHs and CCHs. The CHs use these channels to support QoS for real-time delivery of safety messages and to regulate the throughput for the nonreal-time traffic through these channels. The solution proposed produces considerable levels of collision when two clusters are in nearby proximity.

2.2. Swarm Intelligence QoS Routing for VANET. In this section we review the routing approaches, which use bioinspired techniques to provide QoS for VANET applications. Among these studies, QoSBeVanet [10] is a multipath routing protocol inspired by the communication paradigm used by bees when foraging for food. In this protocol, *scouts* are packets exchanged by vehicles to discover a destination, and *foragers* are application data packets transmitted over VANET. Each scout records its information in a routing table, and each path in the table is evaluated based on its quality using weighting factors. In QoSBeVanet when a link fails, the protocol floods the network with high control packets for recovery.

Similarly to the work above, the authors of [11] propose HyBR, a hybrid bee swarm routing protocol for VANET. HyBR is a unicast and multipath routing protocol tailored towards road safety applications, by transmitting safety packets with minimum delays and high packet delivery rate. In the case of a dense network, HyBR uses a topology-based routing procedure for network discovery using scouts and foragers. When the network density is low, the protocol uses geography-based routing procedure using a genetic algorithm to search for the shortest path to the destination.

HyBR generates a high level of routing overhead due to the use of considerable number of control packets.

In [12], the authors propose MAZACORNET, a solution to search for multiple routes for a given source and destination using a hybrid multipath ant colony based routing algorithm. MAZACORNET partitions the network into multiple zones, using a proactive approach to find a route within a zone, and reactive approach to find route between zones by using local information stored in each zone.

The authors of [13] propose MQBV (multicast QoS swarm bee) routing protocol for VANET. MQBV is a reactive, adaptive, and tree-based protocol. In MQBV, the source node uses a stochastic function to disseminate randomly its route request (forward scout) to all cluster members group members using a combination of unicast and multipath forwarding. The cluster head forwards the request to the next cluster only if the QoS conditions are checked and satisfied. When the packet reaches the destination, a multicast backward scout is launched back to the source along the reverse path. In MQBV protocol, the urban scenario with a highly dense environment is assumed, and the link lifetime metric is not considered.

In [14] an optimization algorithm called bees life algorithm (BLA) was proposed for QoS in VANET using multicast routing. The main objective is to find the optimal multicast tree from the source to multiple destinations with minimum cost, reduced delays and jitter, and a maximum bandwidth. Three QoS criteria are considered when building the multicast tree: delay, jitter, and a minimum bandwidth. However, this technique does not factor in essential metrics such as the packet loss and routing overhead.

In [15], the authors propose SAMQ, situation-aware multiconstrained QoS routing algorithm. SAMQ is based on ant colony system (ACS) and situational awareness (SA) model. The novelty of SAMQ is in its use of SA levels and ACS mechanisms by the vehicles to respond locally and immediately to link failures without interrupting the data transmission. SAMQ recuperates the link breakage by switching links or subroutes at or near the failure point. The routing algorithm in SAMQ selects reliable and optimal QoS routes and keeps evaluating the current situation through the SA levels. Due to the high number of control packets that are generated, this technique can experience higher packet loss.

The authors of [16] propose ARP-QD, an adaptive routing protocol for urban VANET environments. ARP-QD takes into account the traffic density when searching for the best path and uses an adaptive neighbor discovery algorithm to obtain information on local vehicular density so as to reduce the network overhead and improve resource usage and to recover quickly when the routes are interrupted. Two metrics are used: connectivity and distance (CDP) and segment selection weight (SSW). For the forwarding strategy, ARP-QD uses global distance and speed. Although ARP-QD has used newer metrics to search for a stable routes, this protocol does not support traditional QoS criterias like delay, jitter, and bandwidth.

Similarly to the previous study and for the context of urban VANET environment, the authors of [17] propose an intersection-based QoS routing protocol for VANETs (IRQV)

using ant colony optimization techniques. IRQV calculates the best route between two terminal intersections close to the source and the the destination nodes. The selection of the route is based on two QoS metrics: connectivity probability and transmission delay. IRQV uses a greedy carry-and-forward mechanism for the transmission of the data packet between the intersections where forwarding data to the next intersection is decided according to the maximum global pheromone. In IRQV, all the intersections integrated into the vehicular network as fixed communication nodes. Therefore, IRQV requires the use of a geographical location system and digital map.

In Table 1, we provide a summary of the protocols discussed above. The protocols are classified based on the supported QoS metrics, the adopted scheme and paradigm, the mobility model, the type of environment, and the forwarding strategy. We may note that most of proposed protocols are developed for urban environment and operate under reactive mode. The performance analysis of most of these protocols has been done through NS2 simulator.

The above review focuses on the general process of finding routes that satisfy QoS requirements only, without considering the details of the following procedure: routing overhead, stability of communication links, and recovery process of link failures, congestion and packet loss due to broadcasting (discovery), and time and memory complexity of the proposed solutions. As we can see, the proposed approaches have some drawbacks attached to the process of achieving feasible QoS compliant routes. To overcome these drawbacks, we propose a new QoS routing protocol for vehicular networks by bringing improvements on two levels: (1) improving the clustering schemes so as to reduce routing overhead and minimizing the congestion due to the discovery phase, and (2) using an artificial bee colony approach for searching for optimal routes that are compliant with the application's QoS requirements. By using these two approaches, the objective is to increase the link stability and packet delivery rate. The QoS metrics used in our solution are driven by the nature of VANET environments.

3. Problem Formulation

A vehicular network is represented by a undirected graph $G(V, E)$, where V denotes a set of vertices representing vehicles and E the set of edges representing the communication links connecting the vehicles. A link $l \in E$ is denoted by (u, v) directly connecting vehicle u to vehicle v , and, in this case, u and v are said to be neighbors to each other. We define $N(u)$ as the set of all the neighbors of vehicle u ; $N(u) = \{v \in V \text{ s.t. } (u, v) \in E\}$. We define a path $p(u, v)$ as a sequence of links connecting vehicles u to v , and P as the set of all possible paths in G . That is, $p(u, v)$ is defined recursively as $p(u, v) = (u, v_0, \dots, v_n, v)$, where $(u, v_0) \in E$ and $p(v_0, v) \in P$.

We assume that each link (u, v) is associated with a link weight vector $\bar{w}(u, v) = (w_0, \dots, w_n)$ in which w_i is an individual weight component, i.e., a single metric such as *bandwidth*, *link life-time*, and *transmission delay*. Accordingly, any path p from a given source to a given destination can be

TABLE 1: QoS-based routing protocols using swarm intelligence: a comparative table.

Protocol	QoS Metrics	Scheme	Routing type	Mobility model	Route discovery	Forwarding strategy	Path Maintenance	Simulator
QoS-OLSR [7]	Bandwidth, Connectivity, Metric Mobility	ACO	Unicast	Urban	Proactive	Greedy Forwarding	Active	MATLAB
PassCar [8]	ND LET	Passive Clustering	Unicast	Highway	Reactive	Optimum Forwarding	Passive	NS2
Multi-Channel MAC [9]	E2ED, Bandwidth	Clustering	Multicast, Unicast	Highway	Reactive, Proactive	Optimum Forwarding	Passive	MATLAB
QoSBeeVanet [10]	AB, E2ED, Hop count	BCA	Unicast	Urban	Reactive	Periodic Neighbour discovery	Passive	NS2
HyBR [11]	Bandwidth, E2ED	BCA	Unicast	Urban	Reactive	Optimum Forwarding	Passive	NS2
MAZACORNET [12]	LS, PDR	ACO	Unicast	Urban	Reactive, Proactive	Optimum Forwarding	Passive	NS2
MQBV [13]	AB, E2ED	BCA	Multicast, Unicast	Urban	Reactive	Greedy Forwarding	Passive	NS2
BLA [14]	E2ED, Jitter, Bandwidth	BCO	Multicast	Urban	—	Optimum Forwarding	Passive	NS2
SAMQ [15]	LET, PDR, Hop count	ACA	Unicast	Highway	Reactive	—	Active	OMNet++
ARP-QD [16]	Hop count, LET	OFA	Unicast	Urban	Reactive	Greedy Forwarding	Active	NS2
IRQV [17]	CP, E2ED	ACO	Unicast	Urban	Reactive	Greedy Forwarding	Active	NS2

ACO: Ant colony optimization; BCA: Bee Colony Algorithm; BCO: bee colony optimization; ACA: Ant Colony Algorithm; OFA: Optimal Forwarding Algorithm; PDR: packet delivery ratio; E2ED: end-to-end delay; LET: link expiration time; ND: node degree; CP: Connectivity probability; LS: link stability; AB: available bandwidth.

TABLE 2: Terminology.

Symbol	Definition
G	undirected graph.
V	set of vehicles
v, u, v_i	vehicle
E	set of of links $E : V \times V$
l	link
(u, v)	link connecting vehicle u to v
$N(v)$	set of v 's neighbors
$p(s, d)$	path connecting vehicles s to d
p	path
(v_0, \dots, v_n)	vehicles forming a path
$w, w(u, v)$	weight assigned to a link (u, v) , QoS metric
B	available bandwidth
ρ	overhead
v	velocity
θ	direction
δ	idle time
c	link capacity
p_{col}	probability of collision
J	jitter
D	delay
L	link expiration time
(x, y)	coordinates
QoS	QoS value
$\alpha, \beta, \sigma,$ and δ	QoS coefficients

assigned a path weight vector $\bar{w}(p) = (\bar{w}_0, \dots, \bar{w}_n)$, where \bar{w}_i is defined by an objective function $f(w_0, \dots, w_n)$ which determines the value of each one of the weight elements for a given link. The weight elements are considered here as QoS values. That is, we define $\bar{q} = \{q_1, q_2, \dots, q_m\}$ as the QoS constraint vector that a route must satisfy for it to be considered a routing solution. For example, let us consider $\bar{q} = \{64kbps, 150ms, 0.2\}$ as the vector of QoS constraints for the minimum bandwidth required, the maximum end-to-end delay, and the probability of route breakage. A route from a given source to a given destination with the following path weight vector $\bar{w} = \{75kbps, 92ms, 0.19\}$ can be considered as a satisfactory solution with regard to \bar{q} . The terminology used in this paper is defined by Table 2.

3.1. QoS Metrics. In this section, we describe the QoS metrics used in our protocol. The QoS metrics are used in two cases: in the selection of the cluster head and in the determination of the best route to the destination. Three QoS metrics considered in these models are: the available bandwidth, the end-to-end delay, and the jitter.

3.1.1. Available Bandwidth. The available bandwidth over a link $(u, v) \in E$, denoted by $Bw(u, v)$, is defined as the maximum throughput at which packets can be transmitted between vehicles u and v without disrupting any already ongoing flow in the network [18]. $Bw(u, v)$ is calculated as

$$Bw(u, v) = (1 - \rho)(1 - p_{col}) \cdot \delta_u \cdot \delta_v \cdot c \quad (1)$$

where δ_u is the idle time at the sender, δ_v is the idle time at the receiver, c is the maximum link capacity, ρ is the additional overhead due to the binary exponential back-off mechanism in 802.11 [19], and p_{col} is the collision probability (calculated using 802.11p Hello packets).

$$\rho = \frac{F + \bar{K}}{T_m} \quad (2)$$

where T_m is the time separating the transmission of two consecutive frames, F representing *DIFS* (Distributed Inter Frame Space) is a fixed interval, and \bar{K} is the number of back-off slots decremented on average for a single frame. That is, the available bandwidth $B(p)$ for an entire path p is defined as

$$Bw(p) = \min_{(u,v) \in p} Bw(u, v) \quad (3)$$

We also define the available bandwidth $Bw(v)$ at node v as the available bandwidth over links connecting v to its neighbors. We use an exponential function to obtain high metric values as an estimator for the available bandwidth:

$$Bw(u) = \min_{(u,v) \in p} e^{-1/Bw(u,v)}, \quad v \in N(v) \quad (4)$$

3.1.2. End-to-End Delay. The end-to-end delay is defined as the time taken for a data packet to be transmitted across a VANET from a source node to a destination node. That is, $D(u, v)$ denotes the transmission time over the link connecting u to v . This metric is significant in understanding the delay introduced by path discovery [20]. The end-to-end delay for an entire path p is therefore defined as

$$D(p) = \sum_{(u,v) \in p} D(u, v) \quad (5)$$

Similarly, we define the delay $D(v)$ at a node v as the average delay incurred by transmitting data to the neighboring vehicles. That is,

$$D(v) = \frac{\sum_{u \in N(v)} D(v, u)}{|N(v)|} \quad (6)$$

3.1.3. Jitter. In general, a delay jitter is defined as the end-to-end delay variation between packets arriving at the destination. In real-time applications such as VoIP, video conferencing, and multiplayer games, jitter is often considered as an important QoS metric for the quality of the transmission [21].

$$J(p) = \sum_{(u,v) \in p} J(u, v) \quad (7)$$

Similarly, we define the jitter $J(v)$ at a node v as the average jitter incurred when transmitting data to the neighboring vehicles. That is,

$$J(v) = \frac{\sum_{u \in N(v)} J(v, u)}{|N(v)|} \quad (8)$$

3.2. Mobility Metrics. In this section, we describe the mobility metrics used in CBQoS-Vanet. We included the mobility metrics to increase the stability of clusters and improve the lifetime of the routes. These metrics are the link expiration time and the average velocity difference.

3.2.1. Link Expiration Time. In highly dynamic wireless networks, it is important to reduce the probability of link breakage. The formula that we use to calculate the link expiration time (LET) is based on work in [22]. LET is considered as the distance separating neighboring vehicles (u, v), and selecting the best path boils down to selecting the set of links with the minimum LET. The formula for the calculation of LET is presented in (9). Assume that two vehicles u and v are within a distance d from each other. Let (x_u, y_u) be the coordinates of u and (x_v, y_v) be that of v . Also, let v_u and v_v be their respective velocities and θ_u and θ_v ($0 \leq \theta_u$ and $\theta_v < 2\pi$) be the moving directions of u and v , respectively. Then, the amount of time, L , two vehicles will stay directly connected ($(u, v) \in E$) is calculated as

$$L(u, v) = \frac{\sqrt{d^2 \cdot (a^2 + c^2) - (ad - bc)^2 - (ab + cd)}}{a^2 + c^2} \quad (9)$$

where

$$\begin{aligned} a &= v_u \cdot \cos \theta_u - v_v \cdot \cos \theta_v \\ b &= x_u - x_v \\ c &= v_u \cdot \sin \theta_u - v_v \cdot \sin \theta_v \\ d &= y_v - y_u \end{aligned}$$

As LET is a function of the distance separating neighboring nodes and their corresponding velocities, we assume that the vehicles in our model are equipped with a GPS receiver with appropriate accuracy. Therefore, the lifetime of a link is the average of the LET of all the links connecting a vehicle to its neighbors. That is, the lifetime of a connected path is the minimum value of the LET for each link in the path, defined as

$$L(p) = \min_{(u,v) \in p} L(u, v) \quad (10)$$

We also measure the ability of a vehicle to maintaining connectivity with vehicles within its transmission range as the average lifetime of the links established with the neighboring vehicles; that is,

$$L(v) = \frac{\sum_{u \in N(v)} L(v, u)}{|N(v)|} \quad (11)$$

3.2.2. Average Velocity Difference. The average velocity difference metric is an important factor for constructing relatively stable clustering topology. Each vehicle has access to the position and the speed of the vehicles within its communication range and hence can calculate the velocity differences. The vehicular average velocity difference is computed as follows:

$$\vartheta(v) = \frac{\sum_{u \in N(v)} |\bar{v}(u) - \bar{v}(v)|}{|N(v)|} \quad (12)$$

where $N(v)$ is the total neighbors number of nodes v , $\bar{v}(u)$, and $\bar{v}(v)$ the average velocity of nodes u and v , respectively in m/s.

3.3. Multiconstrained Optimal Path (MCOP). QoS-based routing algorithm is responsible for finding feasible paths from a source to a destination, which satisfy multiple constraints. This task can be achieved by solving multiconstraint optimal path selection problem (MCOP). The MCOP routing algorithm identifies the optimal path among several feasible paths found by the routing algorithm according to specific QoS criteria.

A path p is to be optimal if and only if all the QoS values presented in the path are optimal. The routing algorithm must therefore return to a path p respecting the MCOP which is defined in the following:

$$w_i \propto q_i, \quad \text{for } w_i \in \bar{w}_p, \text{ and } \in \bar{q}_p \quad (13)$$

where \propto is equivalent to \leq for an additive or multiplicative metric, and \geq for a concave metric. We define $f(p)$ as the nonlinear function that calculates the cost of a path p in which that eliminates all infeasible paths.

$$f(p) = \max_{w_i \in \bar{w}_p, q_i \in \bar{q}_p} \frac{w_i}{q_i} \quad (14)$$

Since vehicles participate to multiple types applications which generates various types of traffic (audio, video, data, email, ftp, etc.), each type of traffic has its own QoS requirements. Therefore, the fundamental function of MCOP is to identify the optimal routes according to various requirements.

4. CBQoS-Vanet Protocol

In this section, we provide a description of CBQoS-Vanet. This protocol is a unicast routing protocol which combines two algorithms: a clustering algorithm and an artificial bee colony inspired algorithm.

4.1. Clustering Algorithm. The clustering mechanism of CBQoS-Vanet is inspired from [7] and modified to make it suitable for VANET environments in highway scenarios with two main objectives: (1) improving the stability of the clustering structure and (2) maintaining end-to-end communication even during frequent link failures.

This clustering algorithm is divided into two components: (1) cluster head (CH) election algorithm whose purpose is to partition the network into clusters and elect an optimal CH based on local QoS requirements; these CHs will play later an important role in determining the best QoS compliant paths; (2) multipoint relay (MPR) selection algorithm where each cluster selects one or few special nodes responsible for relaying data between different clusters. The benefit of MPRs is to minimize data flooding between and within clusters during the dissemination of routing packets. In a sense, MPRs are considered as gateways. Accordingly, members of a given cluster are classified as CH, MPR, or a regular cluster member (CM) as illustrated in Figure 1.

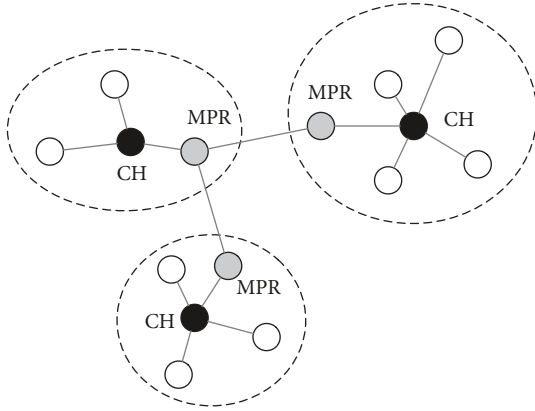


FIGURE 1: Cluster formation in VANET.

0			8			16			31		
Vehicle ID											
Message ID											
Idle Time											
Degree	Role	Average Speed									
GPS Position (X,Y)											
Creation Time											
QoS Value											
Cluster Head ID											

FIGURE 2: Format of Hello Message.

In the CBQoS-Vanet each vehicle periodically broadcasts a *Hello* message (see Figure 2) every δ time units to its neighbors to inform them of its presence. Upon hearing the message, all the vehicles in the sender's neighborhood update their neighbors list accordingly and take note of the received QoS value as show in Algorithm 1. When all the vehicles converge on a common list of neighbors, the best QoS value is determined and therefore the CH is elected. Obviously, QoS values can change dynamically due to the change in the network topology. That is why each QoS value is also associated with a expiry time (determined by the creation time). When a QoS value is changed a new value is created and communicated to all the neighbors and a new CH may be elected accordingly. This is done following the conditions shown in Algorithm 3. Each node continuously updates its neighbors list. Therefore, the *Hello* message is very important in building the local topology. That is, each node keeps track of the link expiration time of its links to its neighbors and update this information every δ time, after which the corresponding node is deleted from the list. This is shown in Algorithm 2.

The QoS model used by vehicles for electing the cluster head is crucial in complying with the QoS requirements and participates greatly in finding the best routes between communicating vehicles. Therefore, the use of QoS metrics and mobility metrics contributes to the stability of the clusters. Each node votes for its CH according to its local QoS value obtained through the *Hello* message. This QoS

value is referred to here as *Suitability value* consisting of a multimetric value derived from both QoS (Q) and mobility (M) metrics. The vehicles with greater suitability values are the most suitable to take the role of a CH. That is, a good candidate to serve as a CH is a node v with the greater suitability value V , determined by

$$V = \psi.Q + \phi.M \quad (15)$$

where Q is the QoS metrics and M is mobility metrics. Furthermore, ψ and ϕ are the corresponding weighing factors to indicate the importance, $\psi \in [0, 1]$ and $\phi \in [0, 1]$, and $\psi + \phi = 1$. In addition, the value of Q is defined by

$$Q = \frac{\alpha.Bw(v)}{\delta.D(v) + \gamma.J(v)} \quad (16)$$

where $\alpha, \gamma, \delta \in [0, 1]$ are weighting factors used here to indicate the importance we give to each one of the three QoS metric and $\alpha + \gamma + \delta = 1$. For instance, if we consider $\alpha = 0.5, \gamma = 0.3$, and $\delta = 0.2$ indicate that we are focusing on maximizing the bandwidth and minimizing the delay and the jitter. Similarly, the value of the mobility metric is calculated using the following equation:

$$M = \beta.\vartheta(v) + \sigma.L(v) \quad (17)$$

where $\beta, \sigma \in [0, 1]$, with $\beta + \sigma = 1$, are weighting factors used here to indicate the importance we give to each one of the two mobility metrics, which are the average velocity difference, and the link expiration time.

4.1.1. Cluster Head Election Algorithm. The cluster head is responsible for selecting the appropriate path between the source and the destination by routing packets according to an artificial bee colony (ABC) algorithm based on the specified QoS requirements. The cluster head election algorithm works as follows: each vehicle v_i broadcasts every δ time period a *Hello* messages to its one-hop neighbors. The *Hello* message (see Figure 2) contains the suitability QoS value. If no new neighbor has joined in within a time period of β , then v_i and all existing nodes in the neighborhood will send an *Elect* message (Figure 4) to the vehicle with the best QoS value selecting it as the cluster head. In the case a node has received an *Elect* message before the β time has expired, and the vehicle does not have the best QoS value, then it will send an *Elect* message to the node in its neighbors list with the highest QoS value. The node which has received an *Elect* message from all its neighbors will change its class to CH and assumes its role as such, and confirms it by broadcasting an *Ack* message to its neighbors. Upon receiving the *Ack* message, all the other nodes will assume their role as regular cluster member (CM) (See Figure 3). When a nonclustered or a previously clustered node with a newly changed suitability QoS value find itself between two clusters, in this case the node may be considered for an MPR position. However, only its suitability QoS value that wil prevail when considered to serve as a CH for any one or both of the clusters it belongs to. This algorithm is described further in Algorithm 3.

```

1:  $N(u_i)$ :  $u_i$ 's neighbors list;
2: Upon receiving Hello message from node  $u_j$  Do
3:   if ( $u_j \in N(u_i)$ ) then
4:     Update neighbors information with a new time of arrival;
5:     Re-calculate  $L(u_j)$ ;
6:   else
7:      $N(u_i) \leftarrow N(u_i) \cup \{u_j\}$ ;
8:     Re-calculate  $L(u_j)$ ;
9:   end if
10: endDo

```

ALGORITHM 1: Updating the neighbors list of u_i .

```

1: TimeNow: Current time;
2:  $N(u)$ : Neighbors list;
3:  $v \in N(u)$ ;
4:  $v_{time}^{hello}$ : Time of Last Hello message arrived from  $v$ ;
5: For every ( $\delta$  time) Do
6:    $TimeNow \leftarrow$  current time;
7:   if ( $L(v) \leq (TimeNow - v_{time}^{hello})$ ) then
8:     Remove node  $v$  from the  $N(u)$ ;
9:   end if
10: endDo

```

ALGORITHM 2: Maintain the neighbors list of u .

```

1: TimeNow: Current time;
2:  $\{v_i, v_j, v_{CH}\} \in V$ ;
3:  $\{v_i, v_j\} \in N(v_{CH})$ ;
4:  $QoS(v_{CH}) > QoS(v_i)$ ;
5: Foreach vehicle  $v_i \in N(v_{CH})$  Do
6:   if ((Last neighbor arrival time + WaitTime  $\beta$ )  $\leq TimeNow$ ) Then
7:     Broadcast Elect ( $v_i, v_{CH}$ );
8:     Flag vehicle  $v_{CH}$  as having best QoS value;
9:   end If
10: endDo
11: Upon receiving at  $v_j$  (Elect ( $v_i, v_{CH}$ )) where ( $v_j \neq v_{CH}$  and WaitTime  $\beta$  not expired) Do
12:   Broadcast Elect ( $v_j, v_{CH}$ );
13:   Flag vehicle  $v_{CH}$  as having best QoS value;
14: endDo
15: Upon receiving at  $v_{CH}$  (Elect ( $v_i, v_{CH}$ )) Do
16:    $v_{CH}\text{-Voted}++$ ;
17: endDo
18: If ( $v_{CH}\text{-Voted} \approx N(v_{CH})$ ) Then
19:    $v_{CH}\text{-role} \leftarrow$  Cluster Head;
20:   Broadcast Ack( $v_{CH}$ ) to  $N(v_{CH})$ ;
21: endif
22: Upon receiving at  $v_i$  Ack( $v_{CH}$ ) from  $v_{CH}$  Do
23:    $v_i\text{-role} \leftarrow$  Cluster Member;
24: endDo

```

ALGORITHM 3: Cluster Head Election Algorithm.

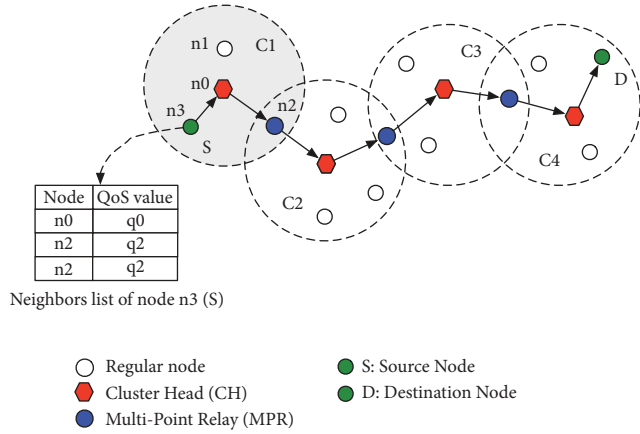


FIGURE 3: Clustering showing how CH (n_0) in cluster C1 is selected based on its suitability QoS values (q_0). QoS values in the neighbors list of n_3 are collected through *Hello* message and ordered with having n_0 having the best value ($q_0 \geq q_i$ for all $q_i \in \{1, 3\}$).

0	31
Vehicle ID	
Message ID	
QoS Value	
Elected Node ID	

FIGURE 4: Format Message Elect.

4.1.2. MPR Selection Algorithm. An MPR node play the role of communication gateway with the neighboring clusters. One cluster may have one or more MPRs. When clusters are formed and the corresponding CHs elected, the algorithm of determining the MPRs kicks off (see Algorithm 4). That is following the election procedure and upon receiving an *Ack* message from a CH, a vehicle takes note of its CH. In addition, all the nodes continue receiving *Hello* messages thus learning about their current CH, and therefore about the presence of other neighboring clusters. The nodes initiate the MPR selection process by sending an *MPRPropose* (Figure 5) to the CH offering to serve as an MPR to a given neighboring cluster. The CH replies with *MPRSelect* (Figure 6) confirming the role of the vehicle as an MPR. If the CH receives multiple *MPRPropose*, the vehicle with the best QoS value will be selected. Upon receiving an *MPRSelect* message, a vehicle changes its role to MPR and starts operating as such for its own cluster.

4.1.3. Cluster Maintenance. Due to the highly dynamic nature of vehicular networks, the vehicles high mobility can quickly and frequently modify the cluster structure, thus causing vehicles to join and leave the clusters frequently. Therefore, the cluster structure needs a maintenance mechanism which keeps its stability as much as possible. This is achieved through during the two operations of join and leaves

- (i) **Joining a cluster:** When a nonclustered vehicle starts the process of joining a cluster through the sending of

0	31
Vehicle ID	
Message ID	
Creation Time	
QoS Value	
Cluster Head ID	

FIGURE 5: Format Message MPRPropose.

0	31
Vehicle ID	
Message ID	
Creation Time	
MPR ID	
Cluster Head ID	

FIGURE 6: Format Message MPRSelect.

a *join* request to the CH, this later checks whether its relative speed is within the predetermined threshold. If so, CH accepts the join request and adds it to the list of its cluster members. This condition ensures that the vehicle requesting to join the cluster will stay for the longest period of time possible when accepted.

- (ii) **Leaving a cluster:** Leaving a cluster happens when a cluster member moves out of the CH transmission range. The CH periodically checks its neighboring vehicles, and when CH stops receiving *Hello* messages from a member in its cluster for a pre-defined period of time, it assumes it has moved out its transmission range. In this case, it removes it from its neighbors list and considers it as no longer member of its cluster. If that member is an MPR, CH selects another one from the neighbors list, one that can play the gateway role and have the best suitability value. Similarly, if the CH itself shows no sign of presence in the neighborhood of the other member, the CH election procedure starts and a new CH is elected. Here again; deciding that a member is no longer part of the cluster has be confirmed through the choice of the right value for the time period for which CH should wait until it is certain that the node has left the cluster.
- (iii) **Cluster merging:** Due to the nature of VANETs cluster heads my come close to each other and get within each other's transmission ranges, leading to relative speeds within the defined threshold velocity. This is usually the case for starting the process of merging the two clusters. In this case, the cluster head that has the lesser suitability value changes its role to cluster member forcing all the members in its cluster to change their status to unclustered. Therefore, these members enter the process of discovering their new cluster head through the election process. The vehicles which cannot join the new cluster and

```

1:  $\{v_{CH}, u_{CH}, v_i, u_i\} \in V$ ;
2:  $v_{CH}, u_{CH}$ : Cluster heads;
3:  $v_i \in N(v_{CH})$ ;
4:  $u_i \in N(u_{CH})$ ;
5: Foreach vehicle  $v_i \in N(v_{CH})$  Do
6:   Upon receiving at  $v_i$  Hello message from  $u_j$  Do
7:     If ( $v_{CH} \neq u_{CH}$ ) Then
8:        $v_i$  send MPRPropose to  $v_{CH}$ ;
9:     end if
10:   endDo
11: end For
12: Upon receiving  $v_{CH}$  MPRPropose from  $v_i$  Do
13:    $v_{CH}$  select  $v_i$  with has best QoS value
14:    $v_{CH}$  send MPRSelect to  $v_i$ ;
15: endDo
16: Upon receiving at  $v_i$  MPRSelect from  $v_{CH}$  Do
17:    $v_i$ .role  $\leftarrow$  MPR;
18: endDo

```

ALGORITHM 4: MPR Selection Algorithm.

TABLE 3: Bee colony in VANET.

Bee Technique	VANET Alternative
Bees environment	VANET environment
Bee Hive	Cluster (contain source node)
Scout Bees	Forward and Backward scout
Food Source	Cluster (contain destination node)
Workers	Intermediate clusters
Recruited Employed Bees	Forager

start clustering mechanism to form a new cluster accordingly.

4.2. Artificial Bee Colony Algorithm

4.2.1. Bees Communication Principle. Routing can be considered as an optimization problem when the objective is to find the best path between a source node and a destination node among many. Various computational problems such as this one can be solved using algorithms inspired by the honey bees behavior. Artificial bee colony (ABC) algorithm is an optimization technique that simulates the mating and foraging behavior of honey bees [3].

In CBQoS-Vanet, VANET is viewed as a bee environment (Table 3) where bees are organized as clusters to accomplish specific tasks. ABC algorithms are modeled on how bees optimize their search for food sources and on the way they communicate to each other. That is, a beehive cluster is seen as the source cluster, and a cluster destination is the cluster of bees at the food locations. Members of intermediate clusters represent beeworkers and are used to transmit data between the source and the destination. Each cluster is represented by the cluster head responsible for routing information [13].

Food foraging behavior is observed when bees search for new nest sites or during the food source foraging. To do this,

some bees, called scouts, navigate and explore the region to find a food source. When found, they come at the dance floor in the beehive to share this discovery with their nest mates via the language of dance, which can be round or waggle related to the distance of discovery. Some bees (called foragers) are recruited to exploit this discovery. Thousands of bees find efficient paths from their beehive to a food source using shared information including location, direction, quantity, and quality of food found [23]. At the end of the search process, the nectar information collected from all the scouts is evaluated and the food source (a path here) that has a good probability to have a big amount of nectar is selected. This probabilistic selection is based on a roulette wheel selection mechanism described by

$$P_i = \frac{f_i}{\sum_j f_j} \quad (18)$$

where f_i is the fitness value of the i^{th} solution presented for evaluation. As expected, here, the higher the probability P_i is the better the solution is. The fitness value is determined by a fitness function represented in CBQoS-Vanet by the QoS value computed by (19). Here, CBQoS-Vanet uses two types of packets. The first type called scout is used to discover routes between the cluster heads to a given destination, while satisfying certain QoS constraints. The second type of packets, called forager, is used to transmit the data between vehicles.

4.2.2. Routing Table. Each cluster head maintains a routing table to route packets between clusters, as it is the responsibility of the cluster head for discovering the network and exploring it before data is exchanged between members of the clusters. Each entry of the table represents a route to a destination and is evaluated according to the QoS condition. Multiple routes to the same destination can be found in the routing table, but each route evaluates to a different QoS

value. The route with a larger PQV is selected to transmit the data packets.

When *backward scout* arrives at a source cluster head, it creates a new entry in the table, which contains the following information: a unique *path id*, a *scout id* which discovered this path, the next cluster head (that is the next hop), and the *hop count*, which indicate the number of cluster heads travelled through before the backward scout reaches the destination. In addition the entry includes a weighted factor field reflecting the QoS value for this route. A timeout value is used to retransmit the *forward scout* if it does not return within a maximum waiting time.

The entries in the routing table are ordered based on the corresponding QoS value. This allows CBQoS-Vanet to acquire the best route with respect to the QoS requirement according to the class of the traffic. Figure 7 illustrates the routing table used by CBQoS-Vanet protocol.

4.2.3. Route Discovery. The first phase in ABC algorithm is the route discovery in which if a node source s wants to establish a path with destination node (d) possible another cluster, it sends a request to CH. In this case, two scenarios are possible: (1) a route to d is available in the routing table of the CH, or (2) no route to d exists in CH's routing table. In the first case, a route to d existing in CH's routing table responds to QoS requirement; CH launches a backward scout which carries information about the past itinerary (or the backward path to s). In the second case, CH launches a forwards scout responsible for exploring the network (Figure 8) and discovering a route to d . Forward scouts use MPRs to move from one cluster to the next one. At each cluster, the cluster head tries to respond to the QoS requirement by verifying the QoS information carried by the the forward scout message (Figure 8) (Bw, D, J , and L). If there is enough resource to accommodate the message, CH updates the message and forwards it on to its neighbors until the message reaches the destination.

To limit the effect of the flooding and the generation of duplicate forward scouts, CH and MPR everytime check whether the forward scout visit is the first one; otherwise its message is dropped.

When the destination node is reached CH propagates the backward scout (Figure 9) back to the CH in the source node along the reverse path taken by the forward scout. Each CH in the reverse path records the backward scout as an entry in its routing table to be used later on by foragers and also as a caching mechanism to ensure that the discovered paths are loop free. When a backward scout arrives at the source s , the cluster head computes the PQV based on the QoS information carried in the backward scout message. CH selects the route with the larger for transmitting network traffic. CH usually allows a predefined period of time for all the backward scout message to arrive before it makes its final selection. Another strategy is to start transmitting data as soon as the first backward scout arrives and the path with the highest PQD is selected, under the provision that CH can change the route when a new backward scout arrives with a better PQD.

The QoS value is recorded along with the route in the routing table. The routing table is updated as *BackwardScouts* arrive at the cluster head (see Algorithm 5). Finally data packets are forwarded through the appropriate path according to the QoS requirements and the *path quality values* PQV associated with the paths in the routing table (see Algorithm 5).

CBQoS-Vanet use (19) to calculate PQV. This equation can be defined as follows:

$$QoS_{max}(s, d) = \max \left[\frac{\alpha.B(v) + \beta.L(v)}{\delta.D(v) + \gamma.J(v)} \right] \quad (19)$$

where $\alpha, \beta, \gamma, \delta \in [0, 1]$ and $\alpha + \beta + \gamma + \delta = 1$. α, β, γ , and δ are coefficients which reflect the importance assigned to the specific QoS metrics.

CBQoS-Vanet uses the MCOP selection problem where it returns the route that maximizes the objective function $QoS(s, d)$ as follows:

$$\begin{aligned} \max [QoS(s, d) = p_{opt} \mid p_{opt} \in p(s, d)] \\ \text{and } p_{opt} = QoS_{max}(s, d) \end{aligned} \quad (20)$$

4.2.4. Route Maintenance. The route maintenance procedure in ABC starts following cases: (1) when a cluster head detects a link failure then any path passing though the broken link is considered as disconnected; and (2) when the cluster head detects an invalidity in the computed path quality value stored in the routing table such that it does not match the QoS requirements, for instance, there is a limited available bandwidth or an increase in the time delay. In both of these cases, an *error scout* is sent by the cluster head to the cluster head source and all cluster heads along the path. All cluster heads in the reverse route are therefore informed about the link failure, thus allowing them to remove it from their routes tables and to start a new route discovery.

5. Performance Evaluation

We used a simulation approach to evaluate the performance of our protocol. The simulation was implemented using OMNeT++ [24] with the VEINS as framework [25] for VANETs. The communication between vehicles is based on IEEE 802.11p standard [26] and IEEE 1609.4 DSRC/WAVE, including multichannel operation [27], QoS channel access, noise, and interference effects to make the simulation of VANET as realistic as possible [28]. We used VEINS as the basis for developing CBQoS-Vanet protocol. For vehicle mobility we use the *car following model* which is implemented in SUMO [29] coupled with a OMNeT++ using a TCP socket developed to permit bidirectional coupling of both the road traffic simulation and the communication simulation.

The evaluation of CBQoS-Vanet is based on two series of experiments: (1) traffic class comparison, where we compare CBQoS-Vanet performance in the context of different classes of traffic such as best effort, audio, video, bulk transfer, and general application (email, browsing, etc.); this simulation is performed to study the various parameters and select

ABE+	Delay	Jitter	LET	$\alpha, \beta, \gamma, \delta$	Val

Path ID	Scout ID	Dest.	CH Dest.	CH Next.	QoS	Hop #	TTL

FIGURE 7: Routing table at the cluster head.

0	16	31
Vehicle ID		
Message ID		
Scout ID		
Food ID		
Beehive ID		
Creation Time		
B_{req} (Kbps)	D_{req} (msec)	
J_{req} (msec)	L_{req} (msec)	
Validity Time		
Cluster Head ID		
...		
Cluster Head ID		

FIGURE 8: Forward scout message format.

0	16	31
Vehicle ID		
Message ID		
Scout ID		
Food ID		
Backward ID		
Beehive ID		
Creation Time		
$Min B$ (Kbps)	$\sum D$ (msec)	
$\sum J$ (msec)	$Min L$ (msec)	
Cluster Head ID		
...		
Cluster Head ID		

FIGURE 9: Backward scout message format.

the best coefficients to be used in CBQoS-Vanet in the second simulation part; (2) protocols comparison, where we compare CBQoS-Vanet performance against two other existing protocols: SAMQ [15] and OLSR [30] QoS-aware routing protocols. These two protocols were chosen among the ones discussed in the related work section because they are representative of the reviewed approaches and use similar techniques of clustering and evolutionary algorithms. The simulation experiments use the scenario of highway with

3 lanes in each direction where each vehicle moves in one direction with the same average speed per lane. The first lane is a fast lane, the central lane is for medium-speed traffic, and the last lane is for slower-speed traffic. We assume vehicles are equipped with global positioning system (GPS) devices, and for the purpose of data transmission, the source and destination are randomly selected for each simulation run.

5.1. Performance Metrics. The selected performance metrics used in the evaluation of CBQoS-Vanet are the same ones used by the protocols used for comparison. These metrics are packet delivery ratio, network overhead, end-to-end delay, average throughput, jitter, and packet loss ratio.

5.1.1. Packet Deliver Ratio (PDR). It is defined as the ratio between numbers of packets received by the destination over the total number of packets transmitted by the source vehicle within a time period.

5.1.2. Normalized Overhead Load (NOL). It represents the ratio between the total numbers of routing packets and the total number of successfully delivered data packets. This metric provides an indication of the extra bandwidth consumed due to routing packets (forward and backward scouts), to deliver data traffic. The lesser overhead, the better the performance.

5.1.3. The End-to-End Delay (E2E). It is defined as the average time that packets take to traverse the network, calculated as the summation of the transmission delay (at MAC layer), the propagation delay, and the queuing time. E2E depends on number of hops and the traffic congestion on the network.

5.1.4. Throughput. It is defined as the average rate at which the data packet is successfully delivered from source to destination in a certain time period over a communication network.

5.1.5. Jitter. It refers to the variation in delay that occurs between two successive packets. The delays between the different packets need to be low for better performance in vehicular ad-hoc networks.

```

1:  $\{v_i, u_i, v_{CH}, u_{CH}, v_{mpr}, u_{mpr}, x_i, x_{CH}, x_{mpr}\} \in V$ ;
2:  $v_{CH}, u_{CH}, x_{CH}$ : Cluster head;
3:  $v_i, u_i, x_i$ : Cluster members;
4:  $v_{mpr}, u_{mpr}, x_{mpr}$ : MPRs;
5:  $v_i, v_{mpr} \in N(v_{CH})$ : Destination;
6:  $u_i, u_{mpr} \in N(u_{CH})$ : Source;
7:  $x_i, x_{mpr} \in N(x_{CH})$ : Intermediate;
8:  $u_{rt}, v_{rt}, x_{rt}$ : routing table;
9:  $u_i$  would sent Data to  $v_i$ ;
10:  $u_i$  sent message SentData to  $u_{CH}$ ;
11: if ( $u_{CH}$  receive message SentData from  $u_i$ ) Then
12:   If ( $u_{rt}$  Contain a path to  $v_i$  & & This path can respond to QoS requirement of  $u_i$ ) Then
13:      $u_{CH}$  Select suitable Forager and send SentData to  $v_i$ ;
14:   else
15:      $u_{CH}$  Create ForwardScout ( $v_i$ ) and disseminate to  $u_{mpr}$ ;
16:   end if
17: end if
18: Upon receiving at  $x_{CH}$  ForwardScout ( $v_i$ )
19: If ( $v_i \in N(x_{CH}) \parallel x_{rt}$  Contain path to  $v_i$ ) Then
20:    $v_{CH} \leftarrow x_{CH}$ ;
21:    $v_{CH}$  Create and launch BackwardScout ( $u_i$ ) to reverse path  $v_{mpr}$ ;
22: else
23:    $x_{CH}$  disseminate ForwardScout ( $v_i$ ) to  $x_{mpr}$ ;
24: end if
25: Upon receiving at  $x_{CH}$  BackwardScout ( $u_i$ )
26: if ( $x_{CH} == u_{CH}$ ) Then
27:    $u_{CH} \leftarrow x_{CH}$ ;
28:    $u_{CH}$  recorded BackwardScout ( $u_i$ ) as entry in  $u_{rt}$ ;
29:    $u_{CH}$  Select suitable Forager and send SentData to  $v_i$ ;
30: else
31:    $x_i$  disseminate BackwardScout ( $u_i$ ) to  $x_{mpr}$ ;
32:    $x_i$  recorded BackwardScout ( $u_i$ ) as entry in  $x_{rt}$ ;
33: end if

```

ALGORITHM 5: Route discovery.

5.1.6. *Packet Loss Ratio (PLR)*. It refers to the ratio of the network traffic that fails to reach the final destination over the total number of successfully received data packets.

5.2. *Traffic Classes Comparison*. In this section, we compare the routing performance of CBQoS-Vanet in various classes of traffic in terms of end-to-end delays, delivery ratio, throughput, and jitter. We chose to evaluate these criteria against a varying vehicular density. We assume that CBQoS-Vanet is configured with the QoS weights reflecting the importance attributed to specific QoS metrics. The values assigned to these weights decide the class of traffic that CBQoS-Vanet is supposed to accommodate to. These weights are here experimental values and can be modified by the application during data transmission (see Table 5).

The proposed simulation time is 1000 seconds in a straight road of 4 km. We use a transmission range of 250m because this is most efficient transmission range for DSRC [31]. *Hello* messages are generated every 1 second and are transmitted only on the Control Channel (CCH). Other parameters are summarized in Table 4. We conducted 20 simulation runs. In each run two nodes are selected randomly to exchange

TABLE 4: Simulation parameters (traffic classes comparison).

Parametre	Value
Length of highway	4 Km
Topology	Highway
Number of lanes	3 each direction
Number of Vehicles	50 to 100
Maximum node speed	30 m/s
Transmission range	250m
MAC specification	IEEE 802.11p
Maximum transmission power	10 mW
Minimum signal attenuation	-85dBm
Slot time	13 μ s
DIFS	32 μ s
Data packet size	1000 bytes
Data rate	4 packets/s
Simulation time	1000 s
Number of runs	20 times
MAC bit rate	18Mbps

TABLE 5: Traffic classes parameters.

Traffic class	QoS weights			
	α	β	γ	δ
Bulk transfer	0.4	0.3	0.25	0.5
Audio	0.1	0.3	0.3	0.3
Video	0.3	0.2	0.25	0.25
General applications	0.2	0.2	0.3	0.3

$\alpha + \beta + \gamma + \delta = 1$

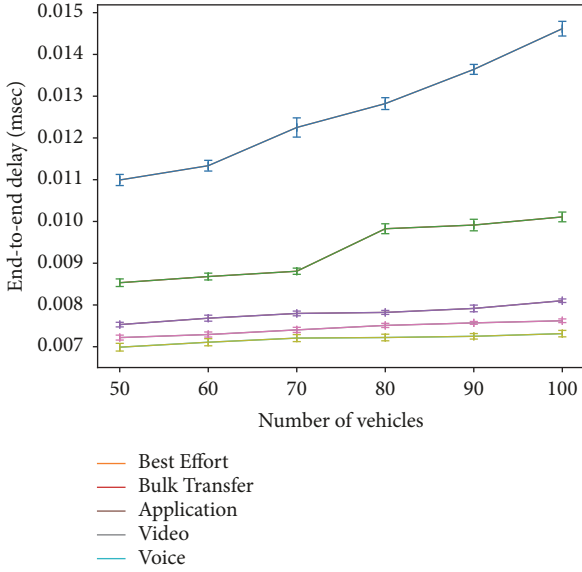


FIGURE 10: End-to-end delay vs. vehicular density.

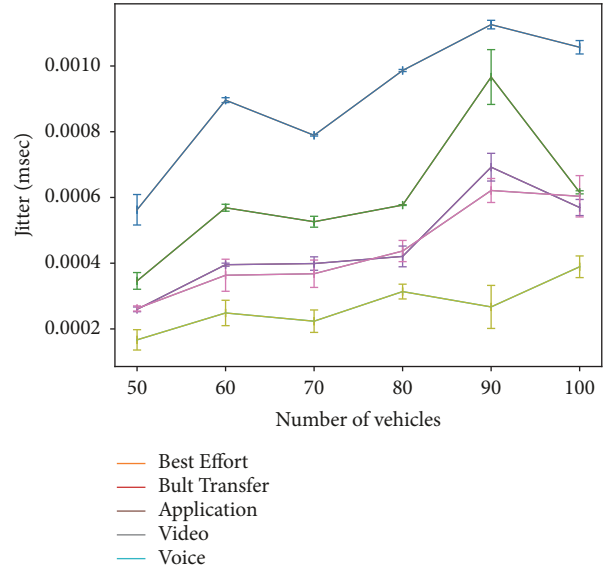


FIGURE 11: Packet deliver ratio vs. vehicular density.

data packets where packet rate is 4 packets per second. All the average results are obtained with a 96% confidence interval.

Figure 10 shows that the voice applications have better end-to-end delays than the other classes of traffic. This is easily explained by the fact that the end-to-end delay coefficient for voice traffic is given a higher value so as to favor routes with better end-to-end delays for voice packets.

Similarly, Figure 11 shows that bulk transfer traffic has better performance with regard to delivery ratio given that the various QoS coefficients configuration favor the selection of routes with the least packet drops where it improves the packet delivery rate. It also shows that compared to the best effort traffic, voice and video traffic performance are close to that of the bulk transfer traffic.

As for the throughput Figure 12 and similarly to the PDR, the bulk transfer traffic exhibits better performance along with the voice and video compared to the best effort traffic. Here again, we show that when configured with the right QoS coefficients, the applications that require higher throughput will be favored and the routes with the highest bandwidth will be selected. Finally, in terms of jitter, Figure 13 shows that voice and video have rightfully the best performance with best effort traffic with the least performance. Here again the results show that the chosen QoS coefficients allow CBQoS-Vanet to select the best routes in term of jitter for

the applications such as voice and video, which require the minimum jitter for their packets.

5.3. Comparing CBQoS-Vanet to Other Protocols. In this section, we present the performance results when comparing CBQoS-Vanet against other two known protocols: SAMQ and OLSR. As explained above, SAMQ is based on OLSR with a similar evolutionary algorithm that is ant colony algorithm and is designed for highway.

The simulation time is 1000 seconds for each scenarios, and the scenarios are run on a 10Km-long highway segment, with a transmission range of 350 meters. Likewise, Hello messages are generated every 1 second and are transmitted only on the Control Channel. Other parameters are summarized in Table 6. It is to note here that, for the sake of preserving the same MAC bit rate used by SAMQ and OLSR, we have selected a 6Mbps bit rate as opposed to the default 18Mbps used in the previous set of experiments (see Table 4). In this set of experiments, we vary the number of vehicles within the range: 25, 50, and 75 to 200 from low density to high density, with an interarrival time that is exponentially distributed. The speeds of the vehicles vary from 1 to 30 m/s. Data packets exchanged between vehicles are 1000 bytes long and packet rate is 4 packets per second. All the average

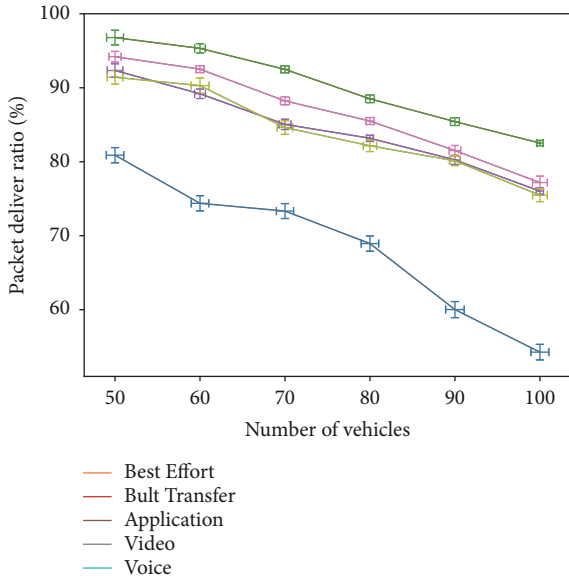


FIGURE 12: Throughput vs. vehicular density.

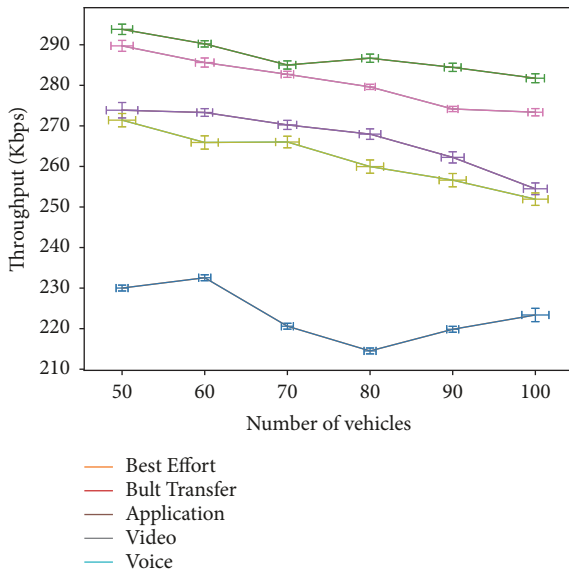


FIGURE 13: Jitter vs. vehicular density.

results for CBQoS-Vanet are obtained with a 96% confidence interval.

Figure 14 shows that in terms of end-to-end delay our proposed protocol outperforms both SAMQ and OLSR. However, SAMQ performance remains very close to that of CBQoS-Vanet for all the vehicles density. There are two elements that explain these results. First, our protocol, through tuning the various QoS coefficients, can obtain better performance with regard to the delays. Second, the reactive nature of CBQoS-Vanet allows it to use the latest reading of the networks in terms of delay to make a decision about the route to take. That also explains why SAMQ has a performance that is close to CBQoS-Vanet, as it is focuses on

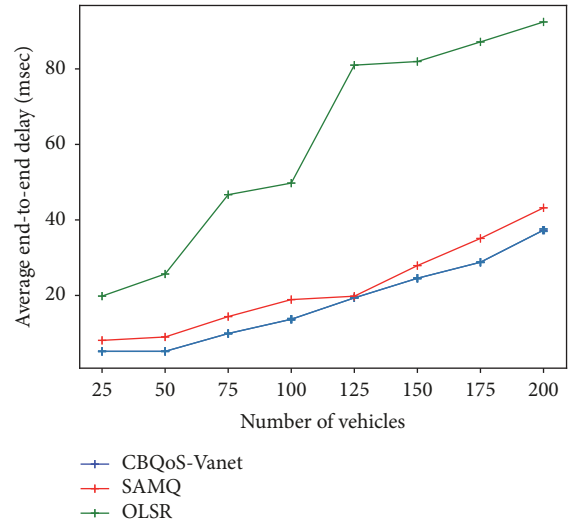


FIGURE 14: Average end-to-end delay vs. vehicular density.

TABLE 6: Simulation parameters (protocols comparison).

Parameter	Value
Length of highway	10 Km
Topology	Highway
Number of lanes	3 each direction
Number of Vehicles	25 to 200
Maximum node speed	30 m/s
Transmission range	350m
MAC specification	IEEE 802.11p
Slot time	13 μ s
DIFS	32 μ s
Data packet size	1000 bytes
Data rate	4 packets/s
Simulation time	1000 s
Number of runs	20 times
MAC bit rate	6Mbps

the use of the concept of life-time characteristic of the selected paths.

In terms of packet delivery ratio, Figure 15 shows that on average CBQoS-Vanet outperforms the other two protocols in providing routes with the best PDR. The results also show that, for CBQoS-Vanet and OLSR, as the vehicles density increases, the delivery ratio decreases almost linearly. However we can observe that, for SAMQ, not only the delivery ration is on average way lower than that of CBQoS-Vanet and OLSR, but this ratio increases with the increase of the vehicles density. The delivery ratios for CBQoS-Vanet and SAMQ are equated when the vehicles density reaches 200 vehicles. This may be explained by the fact that SAMQ provides better PDR performance when the lifetime of the links is higher, and that is achieved with more vehicles in the system which ultimately leads to better connectivity and better link lifetimes. The results in Figure 16 show that, in terms of network overhead, both CBQoS-Vanet and SAMQ

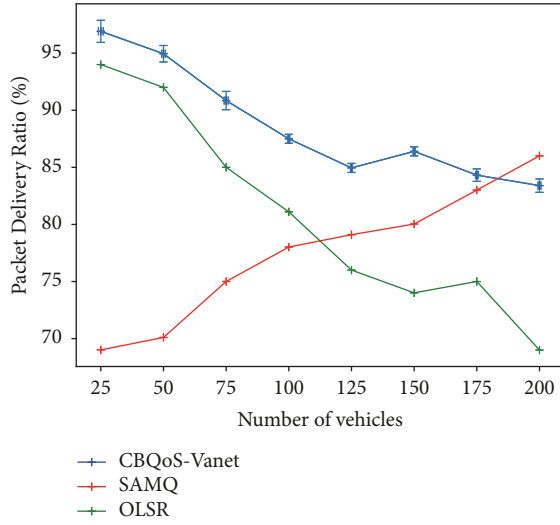


FIGURE 15: Packet delivery ratio vs. vehicular density.

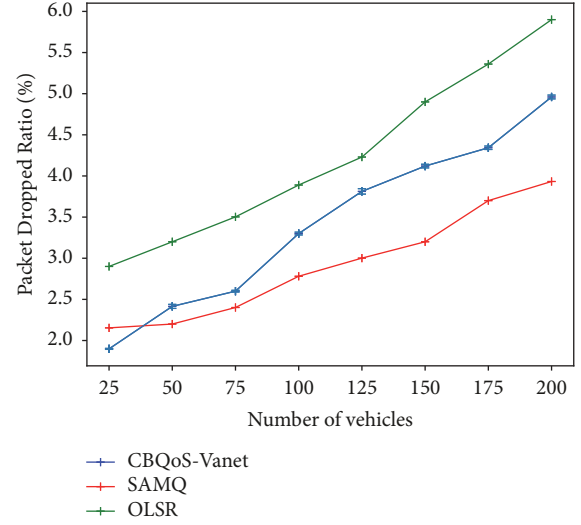


FIGURE 17: Packet dropped ratio vs. vehicular density.

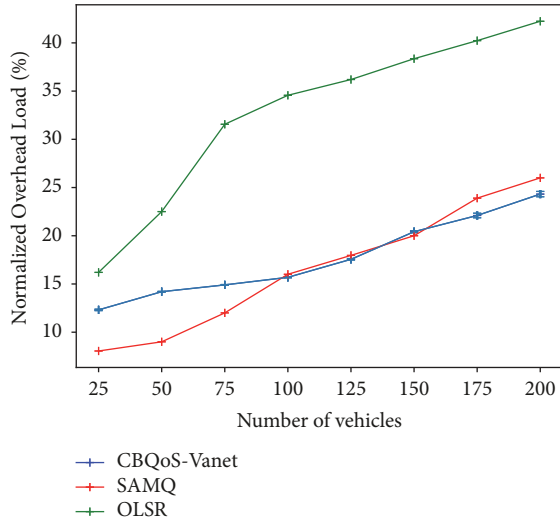


FIGURE 16: Normalized overhead load vs. vehicular density.

have the best performance over OLSR, with a slight advantage for SAMQ at a lower vehicles density. Note that, with higher density, it is expected that more overheads are generated due to the multiplications of possible paths explored and, consequently, the number of routing packets generated. The increase in the overhead is however for all the evaluated protocols linear and with almost the same slope on average with 20% overhead at a density of 200 vehicles for both CBQoS-Vanet and SAMQ and around 42% for OLSR. We may note here that CBQoS-Vanet and SAMQ have similar behavior due the use similar routing packets (forward and backward) included in swarm intelligence algorithms.

Finally, Figure 17 shows the comparison of packet loss ratio for CBQoS-Vanet vs. the SAMQ and OLSR. The results indicate that although the three protocols are within the same range of packet loss, i.e., 2% to 6%, CBQoS-Vanet is in between with the best performance for SAMQ with the lowest

performance for OLSR. The use of ACS routing mechanism and the situational awareness model is likely what gives the slight advantage that SAMQ has over our protocol.

6. Conclusion

In this paper, we presented CBQoS-Vanet, a new routing protocol which enforces QoS in VANET. This protocol is based on the use of two techniques: a cluster-based routing tailored for the case of highways and an artificial bee colony technique for multiconstrained routing optimization. The clustering technique is designed such as to expedite the forwarding of the data through the cluster heads and the multipoint relays only. That is, the formation of clusters and their maintenance are made in such a way to optimize routing according to the required QoS. Therefore, part of the work is already done during the formation of the clusters and the selection of their respective cluster heads. The artificial bee colony approach is used here to find the most QoS-compliant routes using scouts for discovering the network. The best route is determined by computing of a suitability function that takes into account both QoS and mobility metrics. The main improvement brought by CBQoS-Vanet is many folds: first, its capacity to partly enforce QoS requirements during the process of forming clustering based on general QoS requirements and not only the mobility and connectivity requirements; second, supporting a tuning mechanism by the the ant colony algorithm by combining QoS metrics in a formula weighting the different application QoS needs; lastly, optimizing the network discovery algorithm by using a caching mechanism that reduces the network overhead by temporarily caching optimal solutions that were already discovered. The performance shows that CBQoS-Vanet is more suitable to highway scenarios given the fact that it performs well in the case of normal density, and that it shows similar performance to SAMQ when the density is very high. That is, as a future work, we plan to extend this

research to the case of urban VANET and study the behaviour of CBQoS-Vanet under a higher vehicular density and a different mobility model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Roadway, Transportation, and Traffic Safety Research Center (RTT SRC) at the United Arab Emirates University (Grant no. 31R012).

References

- [1] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.
- [2] G. Xue, "Minimum-cost QoS multicast and unicast routing in communication networks," *IEEE Transactions on Communications*, vol. 51, no. 5, pp. 817–824, 2003.
- [3] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [4] W. Y. Szeto, Y. Wu, and S. C. Ho, "An artificial bee colony algorithm for the capacitated vehicle routing problem," *European Journal of Operational Research*, vol. 215, no. 1, pp. 126–135, 2011.
- [5] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A Comparative Survey of VANET Clustering Techniques," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 657–681, 2017.
- [6] M. El Amine Fekair, A. Lakas, and A. Korichi, "Cbqos-vanet: cluster-based artificial bee colony algorithm for qos routing protocol in vanet," in *Proceedings of the 2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*, pp. 1–8, IEEE, Egypt, April 2016.
- [7] O. A. Wahab, H. Otrouk, and A. Mourad, "VANET QoS-OLSR: QoS-based clustering protocol for Vehicular Ad hoc Networks," *Computer Communications*, vol. 36, no. 13, pp. 1422–1435, 2013.
- [8] S.-S. Wang and Y.-S. Lin, "PassCAR: A passive clustering aided routing protocol for vehicular ad hoc networks," *Computer Communications*, vol. 36, no. 2, pp. 170–179, 2013.
- [9] H. Su and X. Zhang, "Clustering-based multichannel MAC protocols for QoS provisionings over vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3309–3323, 2007.
- [10] S. Bitam and A. Mellouk, "QoS swarm bee routing protocol for vehicular ad hoc networks," in *Proceedings of the 2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, Kyoto, Japan, June 2011.
- [11] S. Bitam, A. Mellouk, and S. Zeadally, "HyBR: a hybrid bio-inspired bee swarm routing protocol for safety applications in Vehicular Ad hoc NETWORKS (VANETs)," *Journal of Systems Architecture*, vol. 59, no. 10, pp. 953–957, 2013.
- [12] H. Rana, P. Thulasiraman, and R. K. Thulasiram, "MAZACORNET: Mobility aware zone based ant colony optimization routing for VANET," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation, CEC 2013*, pp. 2948–2955, Mexico, June 2013.
- [13] S. Bitam, A. Mellouk, and S. Fowler, "MQBV: Multicast quality of service swarm bee routing for vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 9, pp. 1391–1404, 2015.
- [14] S. Bitam and A. Mellouk, "Bee life-based multi constraints multicast routing optimization for vehicular ad hoc networks," *Journal of Network and Computer Applications*, vol. 36, no. 3, pp. 981–991, 2013.
- [15] M. H. Eiza, T. Owens, Q. Ni, and Q. Shi, "Situation-aware QoS routing algorithm for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5520–5535, 2015.
- [16] Y. Sun, S. Luo, Q. Dai, and Y. Ji, "An adaptive routing protocol based on QoS and vehicular density in urban VANETs," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 631092, 13 pages, 2015.
- [17] G. Li, L. Boukhatem, and S. Martin, "An intersection-based QoS routing in vehicular ad hoc networks," *Mobile Networks and Applications*, vol. 20, no. 2, pp. 268–284, 2015.
- [18] C. Tripp-Barba, M. Aguilar Igartua, L. Urquiza Aguiar, A. M. Mezher, A. Zaldívar-Colado, and I. Guérin-Lassous, "Available bandwidth estimation in GPSR for VANETs," in *Proceedings of the the third ACM international symposium*, pp. 1–8, Barcelona, Spain, November 2013.
- [19] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous, "Bandwidth estimation for IEEE 802.11-based ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 10, pp. 1228–1241, 2008.
- [20] V. Patil, "Effect of traffic type on the performance of table driven and on demand routing protocols of manet," *International Journal Of Computational Engineering Research (ijceronline.com)*, vol. 2, 2012.
- [21] M. Boban, G. Misek, and O. K. Tonguz, "What is the best achievable qos for unicast routing in vanets?" in *Proceedings of the IEEE GLOBECOM Workshops*, pp. 1–10, November 2008.
- [22] L. Zhang and H. El-Sayed, "A novel cluster-based protocol for topology discovery in vehicular ad hoc network," *Procedia Computer Science*, vol. 10, pp. 525–534, 2012.
- [23] S. Bitam, M. Batouche, and E.-g. Talbi, "A survey on bee colony algorithms," in *Proceedings of the Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pp. 1–8, IEEE, 2010.
- [24] <http://omnetpp.org/>.
- [25] <http://veins.car2x.org/>.
- [26] Q. Wang, S. Leng, H. Fu, and Y. Zhang, "An IEEE 802.11p-based multichannel MAC scheme with channel coordination for vehicular Ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 449–458, 2012.
- [27] D. Eckhoff and C. Sommer, "A multi-channel IEEE 1609.4 and 802.11p edca model for the veins framework," in *Proceedings of 5th ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems: 5th ACM/ICST International Workshop on OMNeT++, OMNeT*, 2012.
- [28] C. Sommer and F. Dressler, "Progressing toward realistic mobility models in VANET simulations," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 132–137, 2008.

- [29] <http://sumo.sourceforge.net>.
- [30] J. Haerri, F. Filali, and C. Bonnet, "Performance comparison of aodv and olsr in vanets urban environments under realistic mobility patterns," in *Proceedings of the 5th IFIP Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net-2006)*, vol. 1045, Lipari, Italy, 2006.
- [31] G. Yan, D. B. Rawat, and B. B. Bista, "Provisioning vehicular ad hoc networks with quality of service," *International Journal of Space-Based and Situated Computing*, vol. 2, no. 2, pp. 104–111, 2012.



Hindawi

Submit your manuscripts at
www.hindawi.com

