

CBQoS-Vanet: Cluster-based Artificial Bee Colony Algorithm for QoS Routing Protocol in VANET

Mohammed El Amine Fekair

Computer Science Department,
University of Ouargla, Algeria

Email: fekair.medamine@univ-ouargla.dz

Abderrahmane Lakas

College of Information Technology
UAE University, Al Ain, UAE

Email: alakas@uaeu.ac.ae

Ahmed Korichi

Computer Science Department,
University of Ouargla, Algeria

Email: ahmed.korichi@univ-ouargla.dz

Abstract—Recent years have seen a growing interest in Vehicular Ad hoc Networks (VANETs) and their benefits to the development of intelligent transportation systems (ITS). With the deployment of multimedia services over VANETs, there is a need to develop new approaches to insure higher level of quality of services (QoS) for real time applications, and integrate QoS into routing protocols. However, in VANET environment, it is not an easy task to search for routes which satisfy the QoS required by the applications. In this paper, we propose CBQoS-Vanet, a new QoS-based unicast routing protocol for vehicular networks. This protocol is based on the use of two techniques: a clustering algorithm which organizes and optimizes the exchange of the routing information based on QoS requirements, and an artificial bee colony algorithm, which finds the best routes from a source to a destination based on QoS criteria. In our approach clusters are formed around cluster-heads that are elected based on QoS consideration. In this paper we consider the following QoS criteria: available bandwidth, end-to-end delay, jitter, and link expiration time. Through simulation experiments, we show that our method can improve greatly the performance of routing in VANET by selecting routes based on the above mentioned QoS criteria.

Keywords—VANET, QoS, Clustering, Artificial Bee Colony, ad-hoc networks.

I. INTRODUCTION

In recent years, Vehicular Ad hoc Networks (VANETs) have attracted the interests of researchers 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. The main objective of VANETs is to expand the reach to Internet and to access available services while on the road. However road safety and travel comfort applications are seeking to benefit from vehicular networks. With the deployment of multimedia and real time application over VANETs, there is a need not only to route information between vehicles in an optimal way, but also the need to select routes that comply the the Quality of Service (QoS) that are usually required by such applications. QoS requirements typically include bandwidth availability, end-to-end delay, jitter, but other criteria that are inherent to the nature of vehicular networks can be considered. Indeed, in VANET where the network topology is highly dynamic and connectivity between vehicles is constantly changing, one may consider the stability of routes and the life expectancy of links as QoS criteria. Therefore, providing support for QoS in such a highly dynamic environment

can be a very challenging task. Indeed, the objective of a QoS routing protocol is to find a route between given source and destination with the required QoS conditions despite the high mobility of the communicating nodes and the link failure that may occur between disconnecting vehicles.

In this paper, we present CBQoS-Vanet, a new QoS-based routing protocol over VANET, based on the use of two techniques: a QoS-based clustering algorithm and a bio-inspired Artificial Bee Colony (ABC). The clustering algorithm is used here to organize and optimize the exchange of the routing information between various clusters. Each cluster is formed around a cluster head which is elected based on QoS criteria. The ABC is here to find the best route between given source and destination and which comply with given QoS metrics. ABC operation is inspired by the behavior of bees in their search for food and by their strategy for pinpointing various routes from their hive to various food locations. CBQoS uses the concept of forward and backward scouting which serves as the process for discovering a given destination. In our approach the QoS criteria considered are: available bandwidth, end-to-end delay, jitter, and link expiration time. CBQoS-Vanet can be considered as a reactive routing protocol, however, it is equipped with a caching scheme which allows routes to be proactively discovered without completing the full scouting process.

The remainder of this paper is organized as follows: in section II, we briefly review the related work. In Section III, we present a detailed description of our proposed protocol CBQoS-Vanet. In Section IV we provide an analysis of the performance results obtained from simulation experiments. Finally, we draw our conclusions in Section V.

II. RELATED WORK

There is a several research studies that have been carried out on routing for VANET in general and on QoS-based routing in particular. In this section we will limit our review of the existing studies to those adopting the clustering and bio-inspired swarm intelligence approaches.

Many routing protocols such as [1] and [2] are based route discovery while minimizing network overhead and packet collisions. However only few compute all possible routes before selecting an optimal one. In [3], the authors propose a routing protocol based on clustering approach and ant colony optimization technique. In this approach, the cluster head is selected based on highest QoS value. The cluster head uses

an ant hello message to discover the network and select[s] a MultiPoint Relay (MPR) to form a stable cluster and increase the stability during communications and link failures while satisfying the Quality of Service requirements.

In [4], the authors propose a Passive Clustering Aided Routing (PassCAR) protocol, which enhances the routing performance in one-way multi-lane highway scenario. It includes route discovery, route establishment, and data transmission phases. PassCAR selects suitable nodes to become cluster heads by using multi-metric election approach based on metrics such as node degree, estimated transmission count, and link expiration time. Where this solution construct a cluster during the route discovery phase.

In [5], the authors propose a new quality of service multi-path routing protocol called QoSBeeVanet for VANET. It is based on biological paradigm of communication between bees in the food source searching behavior. This protocol used scout and forager as packets for discover network and transport data to destination where each scout recorded its information in routing table, and for each path in this table is evaluated relevant to its quality by a weighting factor.

The authors of [6] and [7], proposed the hybrid bee swarm routing (HyBR) protocol for VANET. HyBR is unicast and multipath routing protocol which guarantees road safety services by transmitting packets with minimum delays and high packet delivery. Which in case the network is dense use topology-based routing procedure where use two types of packets for discovering network: Scout and Forager which inspired from communication between bees and in case the network density is low use geography-based routing procedure where use metaheuristic method (Genetic algorithm) to find the shortest paths between the source and the destination

In [8], authors proposed a solution (MAZCORNET) for find multiple route between nodes in the network, where developed a hybrid, multipath ant colony based routing algorithm. Where partition the network into multiple zones, where use proactive approach to find a route within a zone and reactive approach to find route between zones by using the local information stored in each zone. This algorithm is scalable and robust to link failures

III. CBQoS-VANET PROTOCOL

In this section, we describe our proposed protocol CBQoS-Vanet, a new quality of service unicast routing protocol adapted for the vehicular ad hoc networks (VANET). CBQoS-Vanet is a new QoS unicast routing protocol where combines two algorithms: Clustering algorithm and Artificial bee colony algorithm, in the following we present this algorithms and relation between them. Before that, we present QoS metrics used in our proposed protocol CBQoS-Vanet.

A. QoS Metrics

In this section, we describes the QoS metrics used in our system. The QoS metrics are used to determine the routes which best comply with given QoS conditions. To improve and optimize the selection of the QoS compliant routes, QoS

metrics are also used here as a basis in the election of cluster heads. The five metrics considered in our model are the (1) available bandwidth, (2) the end-to-End delay, (3) the jitter, (4) the link expiration time (LTE) and (5) the degree.

In order to define these metrics mathematically, let us define our model. We represent VANET by a graph $G = (V, L)$, where V is the set of vehicles in the network and E as the set of connection links between pairs of vehicles; that is, $L = \{l_{ij} | \exists v_i, v_j \in V \times V, v_i \text{ and } v_j \text{ are connected}\}$. The quality of a link $l_{i,j}$ is evaluated by the function $qos : L \rightarrow q$ where q is a value determined by one of the four QoS metrics. Let's assume that p is the path between pairs of vehicles comprising with the following links l_0, \dots, l_{n-1} . The quality of a link $qos(l_i)$, where $l_i \in L$ is evaluated by one or a combination of the four QoS metrics mentioned earlier. Therefore, the quality of a route is function of the quality of all the links, which make up the route. That is, if p is a path connecting directly or indirectly two vehicles v_0 and v_n , then the quality of the path p is stated as $qos(p) = qos(l_0, \dots, l_{n-1})$ where l_i is the link connecting v_i to v_{i+1} .

1) *Available Bandwidth*: The available bandwidth for each link is defined as the maximum throughput at which packets can be transmitted between these two nodes without disrupting any already ongoing flow in the network [9]. To improve the QoS provided with our algorithm, we use an estimator of the available bandwidth in VANETs, and which based on the ABE^+ estimator[10] for IEEE 802.11p networks illustrated by the following equation:

$$ABE^+(s, d) = (1 - K)(1 - p(m, N, s)) \times T_s \times T_r \times C \quad (1)$$

where T_s is the idle time at the sender, T_r is the idle time at the receiver, C is the maximum link capacity; K is the additional overhead due to the binary exponential backoff mechanism in 802.11 networks, and $p(m, N, s)$ is the collision probability for packets the m bits long. The value for the available bandwidth is source-specific and recorded at the neighboring vehicles since ABE^+ estimates available bandwidth between a sender and a final receiver. The calculation of the available bandwidth is obtained by the following:

$$bw(p) = \min(bw(l_0), \dots, bw(l_n))$$

where p is the communication path between two vehicles s and d and $p = \langle l_0, \dots, l_n \rangle$ where $l_0, \dots, l_{n-1} \in L$ are the links, which make up path p . The QoS condition in terms of available bandwidth that a path must satisfy to be selected is expressed as;

$$bw(p) \geq B$$

where B is a given value of the bandwidth.

2) *End-to-End Delay*: The end-to-end delay is the time taken for a data packets to be transmitted across a VANET from source vehicle s to a destination vehicle d . This metric is important for understanding the delay incurred by intermediary vehicles which relay the packets towards the destination but also by the initial route discovery [11]. This metric is defined as follows:

$$Delay(p) = \sum_{i=0}^n Delay(l_i)$$

where p is the communication path between two vehicles $p = \langle l_0, \dots, l_n \rangle$ where $l_0, \dots, l_n \in L$ are the links, which make up path p . The delay condition that a path must satisfy to be selected is expressed as;

$$\text{delay}(p) \leq D$$

where D is a given value of time delay.

3) *Jitter*: The jitter – also referred to as delay variation, is typically used to measure the transmission quality of real-time applications such as VoIP, video conferencing and multiplayer games[12]. It represents the end-to-end delay variation between packets arriving at the destination in comparison to the delay variation between packets leaving the source. This metric is defined as follows:

$$\text{jitter}(p) = \sum_{i=0}^n \frac{\text{jitter}(l_i)}{n}$$

where $p = \langle l_0, \dots, l_n \rangle$ is the communication path between two vehicles and $l_0, \dots, l_{n-1} \in L$ are the links, which make up path p . The QoS condition in term of jitter that a path must satisfy to be selected is expressed as;

$$\text{jitter}(p) \leq J$$

where J is a given value of delay jitter.

4) *Link Expiration Time*: In highly dynamic wireless networks like VANET, it is important to reduce the probability of link breakage. Therefore a measure for a link expiration time (*LET*) can be an important QoS criteria for the quality of a network. The calculation for this measure is based on the work of[13], considering the locations of the vehicles, their velocities, and the direction towards which vehicles are heading. We assume each vehicle is equipped with GPS receiver allowing therefore calculation of various distances based on vehicles' coordinates at different times[11]. The lifetime of a path is the minimum value of the expected lifetime for each link in the path.

$$\text{LET}(p) = \min(\text{LET}(l_0), \dots, \text{LET}(l_n))$$

The QoS condition in term of link expiration time that a path must satisfy to be selected is expressed as:

$$\text{LET}(p) \geq E$$

where E is a given value of the expiration time.

To find a path which satisfies multiple QoS conditions in terms of a given bandwidth, delay, jitter and LET, we use the following formula:

$$\begin{cases} \text{qos}(p) = \frac{\alpha \cdot \text{bw}(p) + \beta \cdot \text{let}(p)}{\delta \cdot \text{delay}(p) + \gamma \cdot \text{jitter}(p)} \\ \alpha + \beta + \delta + \gamma = 1 \end{cases} \quad (2)$$

where α , β , δ and γ are weighting factors which indicate the level of importance we give to the any one of the four QoS components. In general, these factors are decided based on the application. For instance, when the vehicles are transferring real-time data such as audio or video, we may focus on minimizing the jitter and the delay, and maximizing the bandwidth and the LET. For example $\alpha = 0.1$, $\beta = 0.1$, $\delta = 0.4$ and $\gamma = 0.4$.

B. QoS Model for Cluster Head Election

In this section we present the QoS model used by nodes (vehicles) for electing the cluster head. To improve the quality of service and the stability of of the clusters, CBQoS-Vanet use several QoS metrics as we explained above. Each node votes for its cluster head according its local QoS value obtained through the Hello message. Nodes pay extra attention to the link expiration time of the node, as we would like to maximize the stability of the cluster head and thus the cluster. The calculation of the QoS value is based on the formula 2 explained above.

Finally, for the sake of electing the cluster head, we use the formula in (2) with the following parameters $\alpha = 0.4$, $\beta = 0.4$, $\delta = 0.1$ and $\gamma = 0.1$ with a focus on a maximization of the link expiry time and the bandwidth, and a minimization.

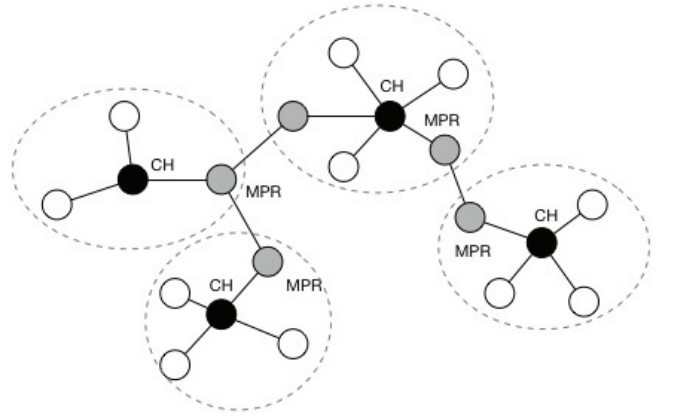


Fig. 1: Clusters

C. Clustering Algorithm

In this section we present a clustering mechanism inspired from [3], and suitable for VANETs in a highway environment, with the objective of improving the stability of the network topology and maintaining an end-to-end communication between nodes even during link failures.

This clustering algorithm can be divided in two components: (1) a *cluster-heads election algorithm* which purpose is to partition the network into clusters and elect an optimal cluster-head (CH) based on QoS requirements. The cluster heads will play later an important role in determining QoS-based routing paths. (2) *Multi-point relays selection algorithm* where each cluster selects one or few special nodes that are responsible for relaying information between different clusters. Considered as gateways, the multi-point relays (MPRs) are used with the objective to minimize the information flooding between and within clusters during the dissemination of routing packets. Accordingly, members of a given cluster are classified as cluster heads (CH), multipoint relays (MPR), or regular cluster member (CM).

In CBQoS-Vanet clustering algorithm works as follows: each vehicle periodically broadcasts a *Hello* message every δ time to its neighbors to inform them of its presence. The Hello message, illustrated in Figure 2, includes a field *QoS value* which is used for the election of the CH. Every time a vehicle

joins a cluster and send a Hello message, all the other vehicles in the neighborhood update their *neighbors list* and take note of the received QoS value. When all the vehicles converge on a common list of neighbors and the corresponding QoS value, 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 *Creation Time*. When a QoS value is changed a new value is created and communicated to all the neighbors and a new CH is elected accordingly. This is done following the conditions shown in Algorithm 1. Each node continuously updates its neighbors list. Therefore the Hello message is very important in building the local topology. For that, each node counts value the link expiration time between itself and the sender. A node is removed from the list of neighbors when no Hello message is received within a δ time. This is illustrated further in Algorithm 2.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MAC Address						ID Vehicle						ID Message																			
IDLE Time						Degree						Role																			
GPS Position						Average Speed																									
QoS Value						Creation Time						Cluster Head ID																			

Fig. 2: Hello Message

Algorithm 1 Adding and Updating the neighbors list.

```

1: NeighList: Neighbors list;
2: Upon receiving (Hello message) from node  $n_i$  Do
3:   if ( $n_i \in \text{NeighList}$ ) then
4:     Update neighbors information with a new time of arrival;
5:     Re-calculate link expiration time;
6:   else
7:      $\text{NeighList} \leftarrow \text{NeighList} \cup \{n_i\}$ ;
8:     Re-calculate link expiration time;
9:   end if
10: endDo

```

Algorithm 2 Maintain the neighbors list.

```

1: TimeNow: Current time;
2: LET: Link Expiration Time;
3: NeighList: Neighbors list;
4: For every ( $\delta$  time) Do
5:    $\text{TimeNow} \leftarrow$  current time;
6:   if ( $n_i.\text{LET} \leq \text{TimeNow}$ ) then
7:     Remove node  $n_i$  from the NeighList;
8:   else
9:      $n_i.\text{LET} \leftarrow n_i.\text{LET} - \delta$ ;
10:  end if
11: endDo

```

1) *Cluster Head Election Algorithm*: The cluster head is responsible for selecting the appropriate path between the source and destination by routing packets according to an Artificial Bee Colony (ABC) algorithm based on specific QoS requirements. The cluster head election algorithm works as follows: each node n_i broadcast Hello messages to its one-hop neighbors, containing its QoS value every δ time period. Within a *Wait Time* β , if no new neighbor has joined the cluster, then all existing nodes in the neighborhood will send

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MAC Address						ID Vehicle						ID Message																			
QoS Value						Elected Node ID																									

Fig. 3: Elect Message

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MAC Address						ID Vehicle						ID Message																			
QoS Value						ID MPR						Cluster Head ID																			

Fig. 4: Format MPRSelection Message

an *Elect* message to the node with the best QoS value selecting it as the cluster head. In the case a node has received an elect message before the *Wait Time* has expired, and the node does not have the best QoS value, then it will send an *Elect* message to the nodes 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 cluster head and assume this role by broadcasting an *Ack* message to all its neighbors. Upon receiving the *Ack* message, all the other nodes will assume their role of Cluster member. This algorithm is described further in Algorithm 3.

Algorithm 3 Cluster Head Election Algorithm

```

1: N: Number of nodes
2: ID: ID of the node
3: CHID: Cluster head ID
4: ForEach node  $n_i$  ( $1 \leq i \leq N$ ) Do
5:   if (Last neighbor arrival time + WaitTime ( $\beta \geq \text{TimeNow}$ )) Then
6:     Broadcast Elect Msg(ID, CHID);
7:     Flag node as having best QoS value;
8:   endif
9:   Upon receiving (Elect-Msg) and  $n_i.\text{ID} \neq \text{Elect-Msg.CHID}$  Do
10:    Broadcast Elect-Msg (ID, CHID);
11:    Flag node as having best QoS value;
12:  endDo
13: endForEach
14: Upon receiving (Elect Msg) and  $n_i.\text{ID} == \text{Elect msg.CHID}$  Do
15:    $n_i.\text{Voted}++$ ;
16: endDo
17: if ( $n_i.\text{Voted} == N$ ) Then
18:   Node  $n_i.\text{role} \leftarrow$  CH;
19:   Broadcast(Ack msg);
20: endif
21: Upon receiving (Ack msg) and  $n_i.\text{role} \neq \text{CH Do}$ 
22:    $n_i.\text{role} \leftarrow$  CM;
23: endDo

```

2) *MPR Selection Algorithm*: MPR nodes play the role of communication gateways between different clusters. All communication between clusters must transit through corresponding MPRs. One cluster may have one or many MPRs. When clusters are formed and cluster heads are selected for each one of them, the algorithm of determining MPRs kicks off (Algorithm 4). Upon receiving an *Ack* message from a cluster head, nodes take note of their cluster head. When a node receives *Hello* messages from its neighbors, it also learns about the presence of other clusters. The node sends, then, an *MPRPropose* to the cluster head offering to serve as an MPR to a given cluster. The cluster head replies with *MPRSelect* (Figure 4) confirming the role of the node as an MPR. If the cluster head receives multiple *MPRPropose*, the node with the best QoS value will be selected. Upon receiving

an *MPRSelect* message, a node change its role to MPR and starts assuming its role as a gateway of its own cluster to another cluster.

Algorithm 4 MPR Selection Algorithm

```

1: C1 : Cluster 1
2: C2: Cluster 2
3: CM1: Cluster member in C1
4: CM2: Cluster member in C2
5: CH: Cluster head
6: Foreach node  $n_i \in$  CM1 Do
7:   Upon receiving(Hello msg) from  $n_j \in$ CM2 Do
8:     If (CM1.CH  $\neq$  CM2.CH) Then
9:        $n_i$ .send(MPRProposed) to  $n_i$ .CH1;
10:    end if
11:  endDo
12:  Upon receiving (MPRPropose msg) from  $n_j$  and  $n_i$  is CH1 Do
13:     $n_i$ .send(MPRSelection) to  $n_j$ ;
14:  endDo
15:  Upon receiving (MPRSelection msg) Do
16:     $n_i$ .role $\leftarrow$ MPR;
17:  endDo
18: endForeach

```

D. Artificial Bee Colony Algorithm

1) *Bees Communication Principle*: Routing can be considered as an optimization problem when we want to find the best path among many paths between a source node and a destination node. Various computational problems can be solved using algorithms inspired by bees behaviors. Two major classes of such algorithms are derived from the behavior of bees *mating* and *food foraging* [5].

Food foraging behavior is observed when bees search for new nest sites or scouting for new sources of food. To do this, some bees (called *scouts*) navigate and explore the region in search for food. When found, they come at the dance floor in the beehive to share this discovery with their nest mates via a language of dance, which can be round or waggle expressing the distance to the discovery site. Some bees (called *foragers*) are recruited to exploit this discovery by finding efficient paths from their beehive to the food sources using shared information including location, direction, quantity, and quality of food found [14].

CBQoS-Vanet protocol is inspired by the way a swarm of bees communicate and their forage for food. Here, the protocol uses two types of packets. The first type of packets, called *scout* is used to discover route between the cluster head to a given destination, a route that satisfies a certain QoS condition. Thus, the discovered route should satisfy a combination of conditions based on the available bandwidth, the end-to-end delay, jitter, and link stability. The second type of packets, called *forager* is used to transmit the data between nodes.

2) *CBQoS-Vanet RoutingTable*: The routing table in CBQoS-Vanet is situated at the level of each cluster head, which is responsible for discovering the network and exploring it before data is exchanged between members of the clusters. Each entry of the routing table, representing a route to a destination 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 the best QoS value is used to transmit the data packets.

When backward scout arrives at the CH of a source, it creates a new entry in the table, which contains the following

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
MAC Address						ID Vehicle						ID Message																			
Scout ID						Backward ID						Creation Time																			
Beehive ID						QoS Value						Hop Count																			
Food ID						Cluster Head Path																									

Fig. 5: Format Backward Scout Message

information: a unique path id, the id of the scout, 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. The entry includes 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.

3) *Route Discovery*: When a node wants to send data to another node in a different cluster, the source nodes sends a request to its CH. If no route is available in the routing table, the cluster head releases a scout (*forward scout*) to discover a route to the destination. The forward scouts moves from one cluster head to another through MPRs until it reaches the destination. As many copies of forward scouts are forwarded as there are neighboring clusters. As forward scouts are relayed, the routing table in the cluster head is updated with route back to the source. Once the destination cluster head is reached, a backward scout is created and sent back to the source along the reverse path (Figure 5). When the source CH receives all the backward scouts, it selects the path with the best QoS (see Algorithm 5). When a backward scout arrives at the source CH, a weight factor is calculated based on the QoS information carried in the backward scouts. The QoS value is recorded along with the route in the routing table. The routing table is updated as backward scouts arrive at the cluster head. Finally data packets are forwarded through the appropriate path according to the QoS requirements and the QoS values associated with paths in the routing table.

IV. PERFORMANCE EVALUATION

To evaluate the performance of our protocol we used a simulation approach. We considered the case of a 3-lane highway in each direction. Each vehicle is assigned a direction and a average speed selected randomly. However, vehicles are put in the lanes according to their speeds, given that one lane is for the faster vehicles, the middle lane if for vehicles with medium speed and the slower lane is for the slower vehicles. We assume that vehicles are equipped with global positioning system (GPS) devices. The proposed simulation time is 1000 seconds in a straight road of 4 km. We use 250m transmission range because this is most efficient transmission range for DSRC [15]. Hello messages are produced every 1 second and are transmitted only on the Control Channel. Other parameters are summarizes in Table I. We conducted 5 scenario of simulation and varying the number of vehicles simultaneously present on the highway between 50 to 100 vehicles. In each scenario two nodes are selected randomly to exchange data packets. We implemented our simulation using OMNET++ [16] with

Algorithm 5 Route Discovery (IN Cluster Head CHi).

```
1: Require: Ci: cluster, CHi: cluster head of Ci, MPRI : is MPR the cluster Ci, CMI
   : cluster member of Ci
2: if (CHi receive message Send-Packet from its members) Then
3:   If (Routing Table of CHi Contain CMj as destination) Then
4:     If (This path responds a QoS Condition) Then
5:       Select Suitable Forager of this Path for transmits Send-Packet to CMj
6:     end if
7:   else
8:     CHi Create and launch Forward-Scout (CMj) by send to MPRI
9:   end if
10: end if
11: If (CHi receive message Forward-Scout (CMj) from its MPRI ) Then
12:   If (CMj contain in neighbors list or in routing table) Then
13:     CHi Create and launch Backward -Scout (CHi) by send to MPRI
14:   else
15:     CHi forward this Forward-Scout (CMj) by send to MPRI
16:   end if
17: end if
18: if (CHi receive message Backward-Scout (CHi)from its MPRI ) Then
19:   If (CHi is destination the Backward-Scout (CHi) ) Then
20:     CHi select suitable forager for transmit Send-Packet to its destination
    CMj
21:   else
22:     CHi forward this Backward-Scout (CHi) by send to MPRI
23:   end if
24:   CHi recorded information of Backward-Scout (CHi) as entry in own routing
    table
25: end if
```

the VEINS as framework [17] for VANETs. Communication between vehicles is based on IEEE 802.11p and IEEE 1609.4 DSRC/WAVE, including multi-channel operation, QoS channel access, noise and interference effects to make simulation of VANET as realistic as possible. 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[18] which coupled with a OMNET++ using a TCP socket developed by [19] to permit bidirectional coupling of road traffic and communication simulators.

TABLE I: Simulation Parameters

Parametre	Value
Length of highway	4 Km
Topology	Highway
Number of lanes	3 each direction
Number of Vehicles	50, 60, 70, 80,90, and 100 Vehicles
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 s
DIFS	32 s
Simulation time	1000 s
MAC bit rate	18Mbps

A. Performance Metrics

Different performance metrics are used in the evaluation of our proposed solution CBQoS-Vanet: packet delivery ratio, network overhead, and end-to-end delay. These metrics are defined as follows:

1) *Packet Deliver Ratio (PDR)*: : defined as the ratio between numbers of packets received by the destination over the total number of packets transmitted by the source node within a period time.

2) *Normalized Overhead Load (NOL)*: : represents the ratio between total numbers of routing packets and the total number of successfully delivered data packets. This metric provides an indication of the extra bandwidth consumed to deliver data traffic [5]. Typically, the overhead is due to routing packets (forward and backward scouts). The less overhead load, the better performance.

3) *The End-to-End Delay (E2ED)*: : is the average time that packets take to traverse the network. End-to-End Delay is the summation of transmission delay (at MAC layer), propagation delay and queuing time of a packet. E2ED depends on number of hops and the traffic congestion on the network.

B. Simulation Results

In this section we present and explain the performance results obtained from the simulation. Figure 6 illustrates the performance of CBQoS-VANET under the criteria of PDR. The general trend shows that the PDR decreases when the density of the vehicles increases. Looking further into the details we can note that the best performance is obtained when the routes with the highest bandwidth are selected. Routes with the highest bandwidth scenario is indicated by the highest value of the factor α . For instance, we can see that with $\alpha = 0.8$ we can obtained 100% PDR with a density of 50 vehicles and a bit less than 85% PDR with a density of 100 vehicles. The scenario of best effort here is obtained without using QoS-based routing.

Figure 7 illustrates the performance of CBQoS-VANET from the end-to-end delay perspective and considering the bandwidth as a QoS parameter. Here again, the results show that a slight increase of the end-to-end delay when the density of the vehicles increases. However, there is a clear advantage of using CBQoS-VANET compared to a plain Best-Effort algorithm (a difference of 10 mili-second delay). Within CBQoS-VANET, we can notice that there is a slight advantage in end-to-end delay when selecting routes with the highest bandwidth (around 2 mili-second difference between $\alpha = 0.8$ and $\alpha = 0.6$.)

Figure 8 illustrates the performance of CBQoS-VANET from the network overhead perspective, and considering the bandwidth as a QoS parameter. The figure shows a difference between Best-Effort and the CBQoS-VANET, but there is no difference between the different α factors. As we may expect, the results in terms of network overhead are not conclusive except that the CBQoS-VANET has in general less overhead than the Best-Effort method. this can be explained by the fact that in general Best-Effort routing methods provide routes that are for the best optimal in term of number of hops and not necessarily in delays or bandwidth. In the other hand, CBQoS-VANET always looks for the best path in terms of bandwidth and delay even among the shortest-paths. In addition the results do not show a clear trend when we vary the density of the vehicles as if the density does not really influence the network overhead generated. However, we should not rule out the possibility that this may be due to the simulation parameters

used in all our scenarios such as the simulation time and the density limited between 50 and 100 vehicles.

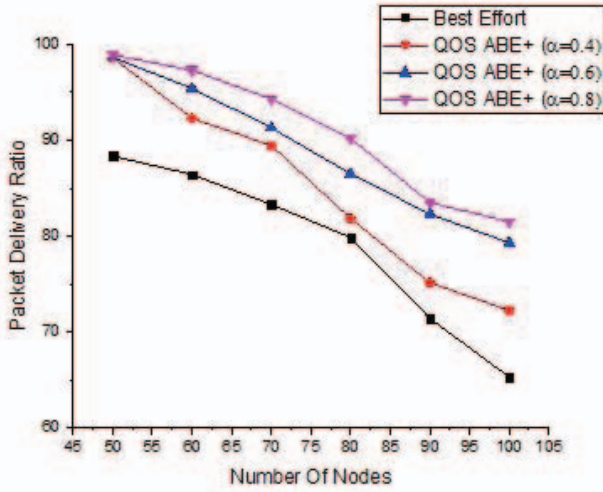


Fig. 6: Packet Delivery Ratio: CBQOS-Vanet vs. Best Effort (with different α values)

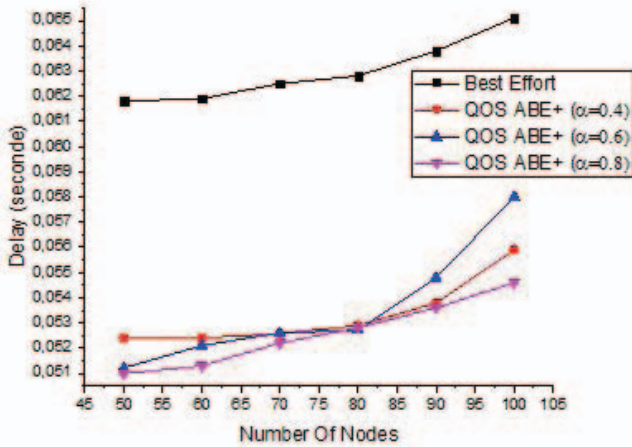


Fig. 7: End-to-end delay: CBQOS-Vanet vs. Best Effort (with different α values)

The second set of simulations concerns the scenarios where we optimize the paths based on the delay (factor β .) Similarly to Figure 6, Figure 9 demonstrates the advantage of using our method over the plain Best-Effort algorithm. Here again, the results show clearly that by selecting the path with the lowest delays ($\beta = 0.8$), better PDR is obtained. Equally for the delay, Figure 10 shows that the performance of CBQoS-VANET is way better than that of the Best-Effort algorithm, with the best performance of CBQoS-VANET obtained with $\beta = 0.8$. As for the network overhead with a variation of the β factor illustrated in Figure 11, we obtained similar results in Figure 8.

In summary, the simulation results show that our protocol is able to be select an optimal path in terms of bandwidth and

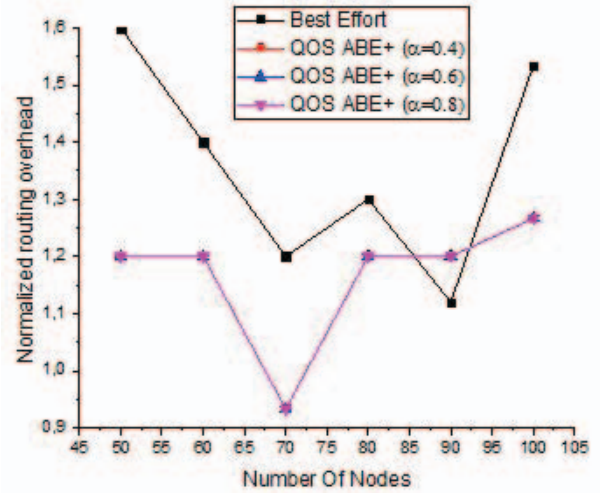


Fig. 8: Normalized Overhead: CBQOS-Vanet vs. Best Effort (with different α values)

delay and allow for the best PDR, delay, and overhead, and that the selected paths satisfy the QoS requirements of various applications.

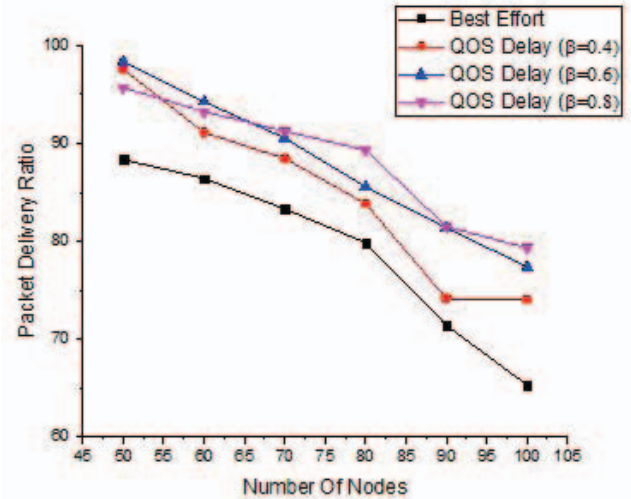


Fig. 9: Packet Delivery Ratio: CBQOS-Vanet vs. Best Effort (with different β values)

V. CONCLUSION

In this paper, we presented a new unicast routing protocol called CBQOS-Vanet which allows vehicles in a VANET environment to find optimal routes based on QoS requirements. Our method is applied in a highway environment includes two components: clustering algorithm and artificial bee colony algorithm. In this paper, we explained these two components and presented their algorithms. We also presented the QoS models for various situations using a aggregated combination of four types of QoS: bandwidth, delay, jitter, and link stability. We have presented and explained the simulation results of

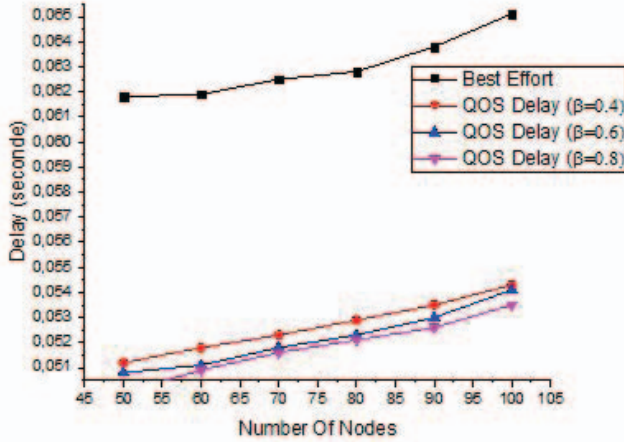


Fig. 10: End-to-end delay: CBQOS-Vanet vs. Best Effort (with different β values)

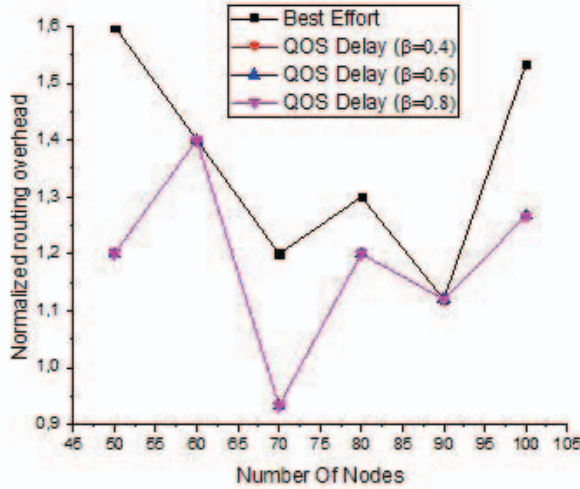


Fig. 11: Normalized Overhead: CBQOS-Vanet vs. Best Effort (with different β values)

our technique in various scenarios, and showed its advantage over the Best-Effort routing method. Although we have implemented all the four types of QoS, we only presented simulation results for the bandwidth and the delay. The results show that selecting a route based on these two QoS requirements does improve the PDR, the end-to-end delays and the network overhead. As a future work, we plan to complete the simulation of CBQoS-VANET by providing results for both jitter and link stability. In addition, we plan to study the right values for the QoS factors for a specific applications such as real-time applications, bulk transfer applications, and multimedia applications.

ACKNOWLEDGMENT

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

REFERENCES

- [1] A. M. Vegni and E. Natalizio, "Forwarder smart selection protocol for limitation of broadcast storm problem," *Journal of Network and Computer Applications*, vol. 47, pp. 61–71, January 2015.
- [2] A. Mostafa, A. M. Vegni, and D. P. Agrawal, "A probabilistic routing by using multi-hop retransmission forecast with packet collision-aware constraints in vehicular networks," *Ad Hoc Networks*, vol. 14, pp. 118–129, March 2014.
- [3] O. A. Wahab, H. Otrok, 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.
- [4] 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.
- [5] S. Bitam and A. Mellouk, "Qos swarm bee routing protocol for vehicular ad hoc networks," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.
- [6] S. Bitam, A. Mellouk, and S. Zeadally, "Bio-inspired routing algorithms survey for vehicular ad hoc networks," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 2, pp. 843–867, 2015.
- [7] —, "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. 953–967, 2013.
- [8] H. Rana, P. Thulasiraman, and R. K. Thulasiram, "Mazacornet: Mobility aware zone based ant colony optimization routing for vanet," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 2948–2955.
- [9] H. Kaaniche, F. Louati, M. Frikha, and F. Kamoun, "A qos routing protocol based on available bandwidth estimation for wireless ad hoc networks," *arXiv preprint arXiv:1101.4034*, 2011.
- [10] C. Tripp-Barba, M. Aguilar Igartua, L. Urquiza Aguiar, A. M. Mezher, A. Zaldívar-Colado, and I. Guérin-Lassous, "Available bandwidth estimation in gprs for vanets," in *Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications*. ACM, 2013, pp. 1–8.
- [11] 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.*, vol. 2.
- [12] M. Boban, G. Misek, and O. K. Tonguz, "What is the best achievable qos for unicast routing in vanets?" in *GLOBECOM Workshops, 2008 IEEE*. IEEE, 2008, pp. 1–10.
- [13] 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.
- [14] B. Salim, M. Batouche, and E.-G. Talbi, "A survey on bee colony algorithms," in *NIDISC'2010 Nature Inspired Distributed Computing Workshop in conjunction with IEEE Int. Symposium on Parallel and Distributed Processing IPDPS'2010*, 2010.
- [15] 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.
- [16] "http://omnetpp.org/."
- [17] "http://veins.car2x.org/."
- [18] "http://sumo.sourceforge.net."
- [19] C. Sommer and F. Dressler, "Progressing toward realistic mobility models in vanet simulations," *Communications Magazine, IEEE*, vol. 46, no. 11, pp. 132–137, 2008.