This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

Performance Improvement of Cluster-Based Routing Protocol in VANET

Ahmad Abuashour[1], Michel Kadoch[2] ,
Department of Electrical Engineering, École de Technologie Suprieure ETS,
Montreal, Canada

***Abstract*—Vehicular Ad-Hoc NETworks (VANETs) have received considerable attention in recent years, due to its unique characteristics, which are different from Mobile Ad-Hoc NETworks (MANETs), such as rapid topology change, frequent link failure, and high vehicle mobility. The main drawback of VANETs network is the network instability, which yields to reduce the network efficiency. In this article we propose three algorithms: Cluster-Based Life-Time Routing (CBLTR) protocol, Intersection Dynamic VANET Routing (IDVR) protocol, and Control Overhead Reduction Algorithm (CORA). The CBLTR protocol aims to increase the route stability and average throughput in a bidirectional segment scenario. The Cluster Heads (CHs) are selected based on maximum Life-Time (LT) among all vehicles that are located within each cluster. The IDVR protocol aims to increase the route stability and average throughput, and to reduce end-to-end delay in a grid topology. The elected Intersection CH (ICH) receives a Set of Candidate Shortest Routes (SCSR) closed to the desired destination from the Software Defined Network (SDN). The IDVR protocol selects the optimal route based on its current location, destination location, and the maximum of the minimum average throughput of SCSR. Finally, the CORA algorithm aims to reduce the control overhead messages in the clusters, by developing a new mechanism to calculate the optimal numbers of the control overhead messages between the CMs and the CH. We used SUMO traffic generator simulators and MATLAB to evaluate the performance of our proposed protocols. These protocols significantly outperform many protocols mentioned in the literature, in terms of many parameters.**

***Index Terms*—VANET, MANET, ICH, IDVR, Grid topology, AODV, Life-Time, CBLTR, CBR, SCSR, CORA, CMHELLO message, CHADS message, control overhead, CH, CM.**

I. INTRODUCTION

THE Intelligent Transportation System (ITS) that includes all types of communications between vehicles is an important next-generation transportation system. ITS provides many facilities to the passengers, such as safety applications, assistant to the drivers, emergency warning, etc. Vehicular Ad Hoc NETwork (VANET) is a derived form of self-organized Mobile Ad Hoc NETwork (MANET). In VANET, vehicles are equipped with an On-Board Units (OBUs) that can communicate with each other (V2V communications), and/or with stationary road infrastructure units (V2I) that are installed along the roads. VANETs have several characteristics that makes it different from MANETs, such as high node mobility, predictable and restricted mobility patterns, rapid network topology change, and frequent battery charging, so energy consumption is not a big issue in VANET[1].

Dedicated Short Range Communication (DSRC) technology is an emerging technology that is developed to work in very high dynamic networks, to support fast link establishment and to minimize communication latency. DSRC is designed to ensure the reliability of safety applications, taking into consideration the time constraints for this type of applications. In the United States, Federal Communication Commission (FCC) has allocated 5.9GHz for DSRC technique to support public and commercial application in V2V and V2I. The frequency takes the range of (5.850-5.925) GHz and divides it into seven non-overlapping 10MHz channels. The DSRC is developed to support the data transfer in a rapidly changing topology networks, such as VANET, where time response and the high transmission rate is required. VANETs deal with two wireless access standards: IEEE 802.11p deals with the physical and MAC layer, and IEEE 1609 deals with higher-layer protocols. According to IEEE 802.11p, vehicles are capable to share their GPS related position together with velocity and acceleration[2].

VANET is proposed and adapts different types of routing protocols, such as proactive[3], reactive[4–6], hybrid [7] [8], and geographic-based routing protocols [9], [10]. The proactive and reactive routing protocols are classified under the topology based routing protocol category, which aims to discover the route between the source and destination before starting the data transmission. The main difference between the two is that the proactive routing protocol initiates a route discovery to all nodes located in the entire network, yielding an increase in control overhead and end-to-end delay. While in the reactive routing protocol, a source node initiates a discovery process to reach only the desired destination. This process reduces the control overhead; however, the route discovery process is required in finding a route for every new node. The hybrid routing protocol combines the features of both proactive and reactive routing protocol. The nodes in the hybrid network are grouped together in a particular area called clusters. Hybrid routing protocols, sometimes called Cluster-Based Routing (CBR) protocols, are designed to improve the network scalability by allowing the nodes within the clusters to communicate through a pre-selected Cluster Heads (CHs) using a proactive routing protocol. However, in the case of

[1] Ahmad Abuashour,
✉ ahmad.abuashour.1@ens.etsmtl.ca

[2] Michel Kadoch,
✉ michel.kadoch@etsmtl.ca

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

communication between clusters, a reactive routing protocol is triggered.

Geographic-based routing protocols or Location-based routing protocols combine the position information with topological knowledge of the actual road map and surroundings. In geographic-based routing protocols, the data is transmitted directly from the source to the destination without initiating any route discovery process. Therefore, each forwarding node assumes to know the following: its current location (using GPS), neighbors locations (by periodically exchanging of Hello messages), and destination location (by using location service protocol[11]).

Intersection-based Geographical Routing Protocol (IGRP) is a location-based routing protocol, which is suitable to urban environments. IGRP [12] is based on an effective selection of road intersections a packet must follow to reach the desired destination. This protocol is characterized by selecting the routes with high route stability. In addition it satisfies QoS constraints with tolerable delay, bandwidth usage, and error rate.

CBR protocols are widely used to improve the scalability of VANET environment and to reduce the control overhead message. Although the clustering techniques are minimizing the routing control overhead, frequent CH elections increase the control overhead associated with the re-election process. The control overhead messages are produced by: First, exchanging of HELLO messages between the CMs and the CH, and second, the CH ADvertiSement (CHADS) messages broadcasted periodically by the CH. When control overhead messages are increasing in a cluster topology, it reduces the available bandwidth resources.

In this article, we define three contributions as follows:

1) We combine the characteristics of geographic-based routing protocol with cluster-based routing protocol to produce a novel CBR protocol. The proposed routing protocol is called Cluster-Based Life-Time Routing (CBLTR) protocol, which objects to eliminate the route discovery process and reduces the number of re-election process for new CHs. CBLTR protocol aims to increase the route stability and average throughput in a bidirectional segment scenario.
2) We propose a novel Intersection Dynamic VANET Routing (IDVR) protocol, which aims to increase the overall network efficiency, by increasing the routes throughput, and decreasing end-to-end delay.
3) We propose a Control Overhead Reduction Algorithm (CORA). The proposed protocol aims to minimize the number of the control overhead messages generated by CMs in a clustered segment scenario.

This article is outlined as follows; in section II, we present state of the arts that related to our works. Section III presents CBLTR protocol in segments scenario. Section IV explains IDVR protocol in a grid scenario. Section V explains CORA algorithm in a segment scenario. Section VI explains the mathematical model. Section VII shows the simulation results and analysis. Finally, section VIII concludes this article.

II. STATE OF ARTS

In general, Cluster-Based Routing (CBR) protocol is a hybrid routing protocol, that divides the large network into small areas called clusters, and inside the cluster, there are a specific routing protocols called intra-cluster routing protocol. The communication between clusters is performed via pre-selected nodes called Cluster Heads (CHs). The CHs are responsible for coordinating the members of the cluster, and communicating betwen clusters using inter-cluster routing protocol[13]. By clustering, only the CH requires to find the destination route. Therefore, the routing overhead is proportional to the number of clusters and not the number of nodes. The objectives of using clusters are to minimize the control overhead, and increase the scalability of the network.

A large number of algorithms have been proposed for the CH election process in VANET. There are many parameters considered to improve the CH election process, such as location, direction, and velocity. In [5], the proposed protocol elects the CHs by considering vehicle movement parameter and link quality between vehicles, forming CHs relatively more stable. The proposed protocol reduces the message overhead and MAC layer contention time at each vehicle while maintaining a high Packet Delivery Ratio (PDR).

In [13], Tao et al. proposed a Cluster-Based Directional Routing Protocol (CBDRP) for highway scenario, in which the CH selects another CH according to the moving direction of the vehicle. The vehicles which are closest to the center coordination of the clusters are elected as cluster heads. This protocol shows significant improvement compared with AODV and is equivalent to GPSR in terms of transmission performance.

Ahmed et al. [14] proposed Cluster-Based algorithm for connectivity maintenance in VANET (AODV-CV), the CH is elected based on the closest actual velocity to the average velocity of all nodes located inside the cluster zone. The proposed protocol outperform AODV in terms of throughput by increasing the velocity.

B. Ramakishnan et al. in[15] presented a new CBR protocol called CBVANET. This model focused on the development of clustering framework for communication among the VANET vehicles. This model decreased the latency in VANET by reducing the cluster creation time, CH election time, and cluster switching time. The vehicle with minimum velocity was chosen as the CH. The proposed protocol outperformed other protocols in terms of the creation time and switching time.

In IGRP [12], a source node needs to know the route it should use to forward data packet to the gateway, which has an up-to-date view of the local network topology. This gateway acts as an allocation service where it stores current location information about all vehicles in its vicinity. By using location management service, each vehicle reports its location information to the gateway as it moves within the transmission range of the gateway. Based on this information, the gateway constructs a set of routes between itself and the vehicles. To increase the stability, IGRP builds routes based only on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

the intermediate and adjacent road intersections toward the gateway.

Christian et al. [16] proposed an intersection routing protocols called Greedy Perimeter Coordinator Routing (GPCR) protocol. When the packet is delivered to the node located at the intersection, GPCR selects the next street that has a node with the shortest route to the destination. Every time the packet is delivered to the intersection, the gateway continues forwarding to the selected path. If a local maximum problem occurred, then an alternative route should be used based on the right-hand rule.

In [16], Moez jerbi et al. proposed an improved greedy traffic-aware routing protocol (GyTAR), which is an intersection-based geographical routing protocol. It uses clusters concept between adjacent intersection to forward the data packet, and it also considers the distance to cluster center to select the cluster head. ChunChun et al. [17] proposed a Vehicle Density and Load Aware (VDLA) routing protocol for VANETs. VDLA selects a series of intersection to construct the route to the destination. The selection is based on the real-time vehicle density, the traffic load of the corresponding road segment and the distance to the destination. VDLA outperforms GPCR in terms of average end-to-end delay and PDR.

IRTIV [18] is a position-based routing protocols that aims to find the shortest connected route to the destination in a city scenario, by taking into consideration the real-time segment density, estimated in a completely distributed manner based on the periodic exchange of Hello messages. IRTIV periodically calculates a real-time cost value by considering traffic density. As a result, IRTIV protocol improves the PDR and reduces the end-to-end delay compared with AODV, and GyTAR.

VANETs are autonomous systems formed by connected vehicles without the need of any infrastructure. Routing in VANET is a significant challenge due to the nature of fast topology changes. The high mobility in VANET forces the vehicles to periodically exchanging control overhead messages. Therefore, the excessive amount of control overhead messages yield to consume high amount of available bandwidth resources.

Control overhead reduction techniques are an important and interesting subject in many recent researches. The main objective of minimizing the control overhead messages is improving the network efficiency by producing more bandwidth resources for data transmission.

The main solution to reduce the control overhead messages is to use the clustering technique, the concept of clustering means to transform the big network into small grouped networks called clusters. In each cluster, one of cluster members (CMs) should be elected to be responsible for all local cluster communication, and its called Cluster Head (CH). This process will significantly reduce the control overhead because it restricts the communication between each CM and CH instead of exchanging the control overhead messages between all the CMs in the cluster. Many researches proposed several algorithms of selecting the CH in each cluster based on specific parameters, such as: vehicle ID, vehicle location, vehicle speed, vehicle direction, and vehicle LT.

In the cluster, CMs and CH should periodically exchange the control overhead messages, the HELLO message is one of important control overhead messages used to define the vehicle identity and location in VANET network. The number of control overhead messages in the cluster is in proportion to the number of CMs. Many techniques are proposed in the literature to reduce the number of HELLO messages as follows:

In [19], the authors proposed a new clustering algorithm that takes into consideration the vehicle position and speed for selecting the CH. The proposed algorithm is intended to increase the clusters stability by reducing the number of CH changes, which yields the reduction of the control overhead produced from frequently re-election process. In [19], the authors do not mention the impact for the size of CHADS messages, and they do not consider the impact of the HELLO messages in terms of its size and its updating time period. In [20], the authors proposed a lane-based clustering algorithm to improve the network stability by reducing the CH election times. The proposed algorithm elects the CH based on the traffic flow of vehicles in the cluster. In [21],the authors enhanced a new parameter to improve the CH election. This parameter is the speed difference. By using this parameter, the cluster becomes categorized based on different speeds.

The CBDRP [13] concentrates on the reduction of the routing overhead packet from source to destination, without considering the control overhead packets produced by the CMs in each cluster. Pedro et al. [22] proposed a Beaconless Routing Algorithm for Vehicular Environment (BRAVE). The proposed protocol objective is to reduce the control overhead messages in broadcast approaches. In BRAVE, the next forwarder vehicle is reactively selected among those neighbors that have successfully received the messages. The drawback of BRAVE protocol is that each vehicle participating in the routing protocol still requires to exchange a beacon messages among them. In the simulation setting, BRAVE sets the exchanging time of the beacon message to 2 seconds to keep monitoring the vehicles location. BRAVE considered as reactive routing protocol. In general, reactive routing protocol reduce the control overhead messages compared to proactive routing protocol. However it still suffers of high control overhead compared to CBR protocols.

Dan et al.[23] proposed a MOving-ZOne-based (MoZo) architecture. MoZo consist of multiple moving zones that group vehicles based on the movement similarity. The selected CH is responsible for managing information about CMs as well as the forwarding packets. The control overhead updating period for the CMs in Mozo architecture varies between moving function of 5 m/s or 4 seconds.

This article proposes a novel Cluster-Base Life-Time Routing (CBLTR) protocol in a segment topology, an Intersection Dynamic VANET Routing (IDVR) protocol in a grid topology, and Control Overhead Reduction Algorithm (CORA)in a clustered topology. The objectives of this article are to increase the route stability and average throughput in a segment topology, reduce end-to-end delay in a grid topology scenario, and reduce the control overhead messages in the clusters. In the next three sections, we analyze the methodologies we followed to achieve our objectives, respectively.

III. CLUSTER-BASED LIFE-TIME ROUTING (CBLTR) PROTOCOL

In this section, we present the steps and algorithms to improve the routing stability in a bidirectional segment scenario, as follows: First, the segment is divided into multiple stationary clusters. Then, a new distributed CH election algorithm is proposed to select a CH based on specific parameters. Finally, a new routing protocol is proposed to select the most suitable candidate CH based on CH's neighbors and destination location.

A. Cluster dividing

The segment is a bidirectional road, and each segment is divided into multiple clusters that equal half of the transmission range of a standard vehicle[13]. We assume that all vehicles have a predefined knowledge of cluster coordination and identification. Each vehicle must be assigned to one cluster at each unit of time based on its location, and with a unique ID for each vehicle and cluster. Figure 1 presents a segment with two clusters, it also shows the cluster edges between the clusters. At any unit of time, if each vehicle enters any cluster zone (enters the cluster edge lines between the clusters), then it becomes a member of this cluster and must send A HELLO message to the CH of the cluster (more details of CH election and sending HELLO messages are explained in section III.B and section V, respectively).

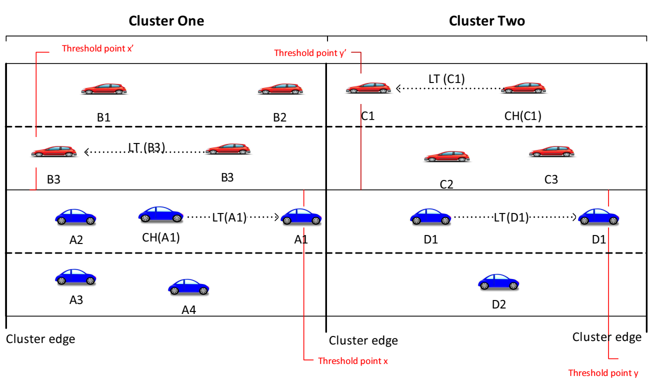


Fig. 1. cluster dividing and CH election

B. Cluster Head (CH) election

Each vehicle that enters a predefined stationary cluster zone should periodically calculate specific cost value, which is called Life-Time (LT). The LT of each vehicle depends on the current velocity of the vehicle as well as the distance to the predefined directional cluster edge (using an Euclidean distance equation). The vehicle with the maximum LT is elected as a CH, then it remains as the CH till it arrives at the directional threshold point; this means there are no new election until the current CH arrives at the predetermined directional threshold point. The directional threshold point is defined as a point distant from the directional edge of the cluster. The distance that separates these two points is calculated by considering the CH velocity, and the time it takes to proceed until the re-election process. The distance from the directional threshold point to the directional edge of the cluster must be enough for a CH vehicle to handover the CH function to another vehicle without losing the communication. This ensures that any vehicle in each cluster can successfully complete the re-election process. For example, if the handover time (re-election time and the time to forward the CH information to the new CH) is equal to $0.2s$, then, the threshold distance (D_{th}) is calculated dynamically based on the current CH velocity. Equation 1 shows and illustrates the calculations of the threshold distance in each specific cluster.

$$D_{th}(CID) = V_{CH}(CID) \times HOT \tag{1}$$

Where:
$D_{th}(CID)$: Threshold Distance for specific cluster
$V_{CH}(CID)$: CH Velocity for specific cluster
HOT: Hand-Over Time
CID: Cluster IDentification

Example: For a CH speed of $50Km/h$,D_{th} of cluster ID 1 equal $D_{th}(1)$ = $\frac{50\times1000m}{3600s}$ $\times$ 0.2s= 2.7 $meter$

Therefore, before 2.7 meters of the directional cluster edge, the Hand-Over process should be invoked, and this value varies based on the current CH velocity. In Equation 2, the LT is periodically calculated for each vehicle within each cluster, using the distance from the current location of the vehicle to the directional edge of the cluster and the vehicle velocity.

$$LT(i) = d_{ith}/(V_i) \tag{2}$$

Where:
$LT(i)$: Life-Time of vehicle i.
d_{ith} : Distance between vehicle i and directional edge of the cluster
V_i : Velocity of vehicle i

In Figure 1, we present a simple process for electing the CH at specific time. In cluster 1, vehicles A1 and B3 are moving in opposite directions and each has the maximum LT in its direction, but the LT of vehicle A1 is greater than the LT of vehicle B3, therefore the vehicle A1 is elected to be as CH for cluster 1. The same election process will proceed in cluster 2, and also because the LT of the vehicle C1 is greater than the LT of the vehicle D1, then the vehicle C1 is elected as CH in cluster 2. Each elected CH (A1 and C2) keeps its status as CH until it arrives to its corresponding threshold point (x and y', respectively). When any CH arrives at its corresponding threshold point, then a new election process should start.

In Algorithm 1, each vehicle enters any cluster becomes member of that cluster, in (lines 1 to 9) we classified the vehicles based on its location in real time. Then the LT is calculated for all vehicles within their associated clusters at any given time. The LT is calculated based on the time that each vehicle will remain in the cluster (as in lines 10 to 23), depending mainly on the distance to the upcoming directional

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

edge of the cluster, as well as the velocity. The vehicle that has the maximum LT at a specific time will be selected as the CH and remains as the CH till it arrives at the directional threshold point. At this time; a new election should be invoked, and a new CH must be selected. The purpose of not updating the CHs all the time is to reduce the control overhead messages produced from the re-election process, in other words, to maximize the LT for the CH.

Algorithm 1 LT calculation and CH election in the segment

for $t = anytime\ \ and\ \ t <= simulationtime$ **do**
  for $VID = 1\ \ to\ \ i <= Numberofveh$ **do**
    for $CID = 1\ \ to\ \ CID <= Numberofclus$ **do**
      if $location(VID) = Location(CID)$ **then**
        Add VID to MCID; add VID to the members of this cluster(CID)
      end if
    end for
  end for
end for
for $t = anytime\ \ and\ \ t <= simulationtime$ **do**
  for $CID = 1\ \ \mathtt{to}\ \ CID <= Numberofclus$ **do**
    while $VID \in CID$ and $VID < MCID(CID)$ **do**
      distance(i)=abs(dirclusedge(VID)-loc(VID))
      LT (VID) = distance(VID)/velocity(VID)
    end while
    CH=VID (index (max (LT)))
    if $location(CH) =$ *threshold point(CH)* **then**
      update CH
    else
      keep old CH
    end if
  end for
end for

Figure 2 presents a flowchart of CH election in each cluster. Each vehicle calculates the LT that it requires to reach the directional edge of the cluster. In each cluster, there is a distributed election algorithm that elects the vehicle that remains within its cluster the maximum LT time. The elected CH should announce itself periodically (every τ second), by forwarding a CH ADvertiSement (CHADS) message. At each time, if any vehicle enters a new cluster zone, its default status is CM, then it should wait (τ second) to receive a CHADS message. If the vehicle receives a CHADS message, then it keeps the status as CM, otherwise, the vehicle announces itself as the CH. The CH should periodically (every τ second) advertise its status until it reaches its predetermined threshold point; then to avoid any communication interruption, the CH asks for early re-election process by advertising a LEAVE message. At this time, the CM with maximum LT will be elected as the new CH. If the CH arrives at the predefined threshold point and the cluster is empty of other CMs, then the CH keeps moving until it finds another CH (more details of exchanging the control overhead messages are explained in section V).

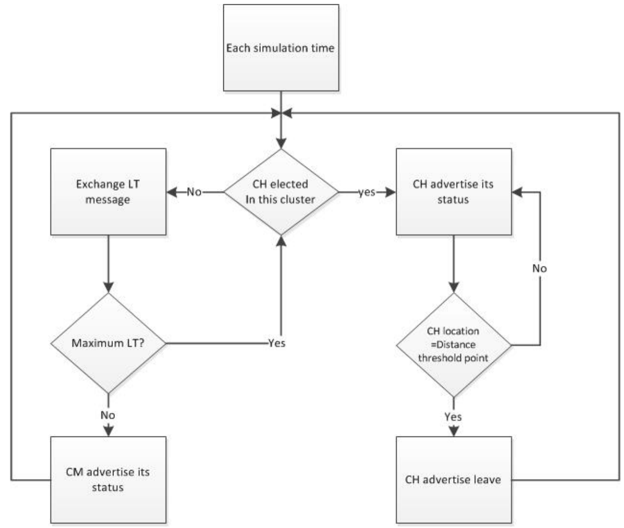


Fig. 2. Flow chart of CH election

C. Routing procedure in the segment

The CBLTR protocol aims to propagate the packets within the segment through the selected CHs. Each CH builds its routing table and stores in it the adjacent CH IDentification (CHIDs) and its associated locations. Figure 3 shows the contents of the CH routing table, which contains the CH IDentification (CHID), its location, its LT, and expiry time. The expiry time is used to keep updating the routing table contents. When the local CH receives a packet, it searches in its routing table for the candidate CHs that are located close to the destination regardless of the CH's direction, then it forwards the packet to the next CH that has the maximum LT. If two candidate CHs with equal LT are available for forwarding the packets, then the CH in the same direction of the local CH is selected. If there are no relaying CHs to the destination, then as a recovery process the local CH follows a store-and-forward process; it stores the packet in a specific buffer and keeps moving till it finds another relaying CH.

Routing Table			
CHID	Location	LT	Expiry time

Fig. 3. CH Routing table

Algorithm 2 shows a pseudo code of the CBLTR protocol that presents the steps of propagating the packets within the segment. At any time of the simulation, if the vehicle receives a packet, it first checks its CH routing table, and then selects the CHs that are closest to the destination in another table, called the candidate CH table (which has the same structure as the CH routing table in Figure 3). The CH with maximum LT of the candidate CH table is selected as the next forwarder vehicle. In case LT values are equal, then the one closest to the destination is selected regardless of its direction. Finally, if the CH routing table is empty, then the current CH follows a store-and-forward process.

Figure 4 illustrates the process of calculating the throughput. The filled circle is the CHs and the unfilled circle is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

Algorithm 2 CBLTR protocol

for $t = anytime \ \ and \ \ t <= simulationtime$ **do**
  if *Packet received by CH at time=t* **then**
    Check Routing table of the CH
    if *Routing table NOT empty* **then**
      Store the closest CHs to destination in Candi CH table
      if *Candi CH table has 2 or more CH with same maximum LT* **then**
        Next CH= CH that closest to the destination
      else
        Next CH= CH with maximum LT
      end if
    else
      Store and Forword
    end if
  end if
end for

the CMs. The LT is calculated for the CHs in each cluster, and only the candidate with maximum LT is selected as the CH regardless of its direction. The throughput is the rate of successful data delivery over a communication channel. In Equation 2, each CH calculates the Transmit(T) time which is the same as the LT for its associated cluster. Each cluster has two directional CHs that move in opposite directions. After calculating the LT time for each CH within each cluster, we select the maximum LT in each cluster. Each CH remains as CH until it arrives to its predetermined corresponding threshold point. The throughput is then calculated by multiplying the transmission rate (S) by the fraction of T that each CH will remain in its associated cluster.

In Equation 3, we calculate the throughput for n clusters in a bidirectional segment.

$$\begin{aligned} Throughput = S_1 \times \frac{max(T_1, T_1')}{max(T_1, T_1') + H} \cdot \\ S_2 \times \frac{max(T_2, T_2')}{max(T_2, T_2') + H} \cdot \\ ....S_n \times \frac{max(T_n, T_n')}{max(T_n, T_n') + H} \end{aligned} \tag{3}$$

Where:
n: Maximum number of clusters on the segment
S_i: The transmission rates for the cluster CH of cluster i
T_1, T_1': The transmit time of CH in cluster i in two directions
H: The Hand-Over time

In Equation 4, for simplicity, we assume that the Ts are the same for the entire segment. We take the average of all Ts for the segment as follows:

$$T_{avg} = \frac{\sum_{k=1}^{n} T_k}{n} \tag{4}$$

By substitute T_{avg} in Equation 3, we calculate the average throughput as in Equation 5:

$$\begin{aligned} AverageThroughput = S_1 \times \frac{T_{avg}}{T_{avg} + H} \cdot S_2 \times \frac{T_{avg}}{T_{avg} + H} \cdot \\ ....S_n \times \frac{T_{avg}}{T_{avg} + H} \end{aligned} \tag{5}$$

But

$$S_2 = S_1 \times \frac{T_{avg}}{T_{avg} + H}, S_3 = S_2 \times \frac{T_{avg}}{T_{avg} + H} \tag{6}$$

In general,

$$S_n = S_{n-1} \times \frac{T_{avg}}{T_{avg} + H}, for \quad n \geq 2 \tag{7}$$

By substitute Equation 7 in 5, we obtain

$$AverageThroughput = S_1 \times \prod_{i=1}^{n} (\frac{T_{avg}}{T_{avg} + H})^i \tag{8}$$

Where:
T_{avg}: Average transmit time for the segment.
i: cluster sequence number.
n: Maximum number of of clusters in the segment.
H: Hand-Over time.
S_1: data rate of the first cluster in the segment

In Equation 8, we calculate the average throughput for any segment size. In addition, it determines the degree of stability for any segment. The segments with higher average throughput indicates higher segment stability. In section VI.A we present more theoretical analysis of LT in a cluster.

IV. Intersection Dynamic VANET Routing (IDVR) Protocol

IDVR is a new Intersection Dynamic VANET Routing protocol. There are two main contributions of this protocol. First, we use the CHs in relaying the packets from the source to the destination; then the CHs are selected based on maximum LT. By relaying the packets via CHs, we increase the segment stability and reduce the probability of link failure[24]. Second, we propose an Intersection Dynamic VANET Routing (IDVR) protocol, which computes the optimal route to the destination taking into account the real-time traffic from source to destination, and the current source and destination intersection location. The IDVR algorithm works in real-time and recursively operates at each intersection until it arrives at the final destination. Our objectives are to increase the route stability and average throughput, and to reduce end-to-end delay in a grid topology scenario. In the next subsections, we analyze the methodology we followed to achieve our objectives.

A. *Software Defined Network (SDN)*

A Software Defined Network is used to provide flexibility to networks and to introduce new features and services

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

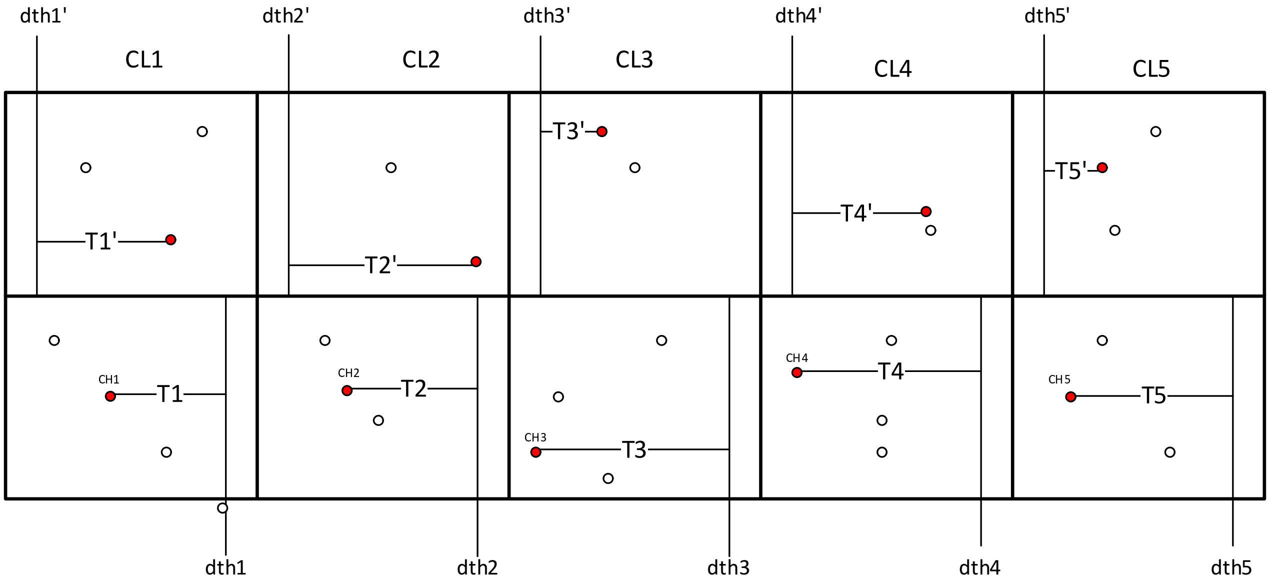


Fig. 4. Average throughput calculation for a bidirectional segment

to VANETs. Ian Ku et al.[25] evaluate the performance of SDN-based VANET architecture with other traditional MANET/VANET routing protocols, including GPSR, OLSR, AODV, and DSDV. The results show that the PDR is much higher when adopting SDN in VANET environments.

We use SDN to define the candidate routes between two intersections; SDN requires creating a table that includes segment IDs, as well as average throughput (as calculated based on Equation 8), and this information must be updated periodically. Figure 5 shows the contents of the SDN table. The design of full SDN architecture is beyond the scope of this article. The SDN provides upon request the candidate routes between the source intersection and the destination intersection (the intersection closest to the destination location) using the Dijkstra algorithm. Each candidate route consists of a series of intersections and the corresponding weight.

SDN table parameters		
Segment ID	Average throughput	Expiry time

Fig. 5. SDN parameters

B. Intersection Cluster Head (ICH)

When any vehicle enters the intersection cluster zone, it wait for τ second. If it receives any CHADS message, then it announce itself as CM and sends a HELLO message to the ICH, otherwise it announce itself as ICH. In Figure 6, there are 4 vehicles want to enter the cluster intersection zone. The first vehicle that enters the intersection zone will announce itself as ICH, and any vehicle enters after that will annouce itself as CM. The ICH Keeps its status as ICH and periodically forward CHADS message until it arrives to its corresponding threshold point. At the moment when the ICH arrives at the threshold point a new election process should be invoked. At this time, the new ICH will be elected among the CMs that are located within the cluster intersection zone and has the maximum LT.

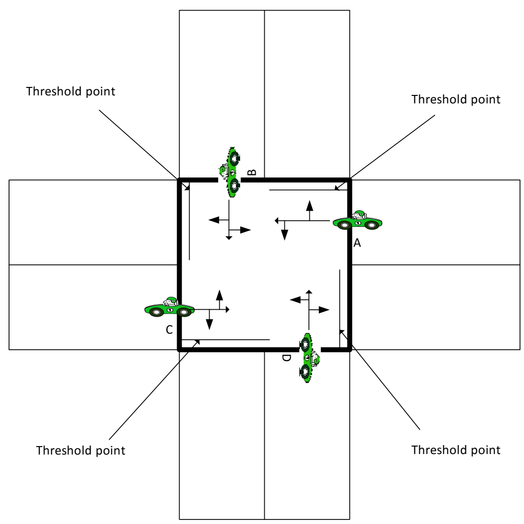


Fig. 6. ICH election

In Algorithm 3, we show how the vehicles join the intersection cluster coordination within the simulation time. Furthermore, we explain how to select the Intersection CH (ICH). First, the vehicle with maximum LT is elected as the ICH and maintains its status until it reaches a predefined threshold point. When it arrives at the threshold point, a new ICH should be elected and all data should propagate to the new elected ICH. The packets are propagated via pre-selected CHs in each segment; as they arrive at the intersection, the packets are propagated via ICH.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

Algorithm 3 ICH election in the intersection

for $t = anytime\ \ and\ \ t <= simulationtime$ **do**
  for $VID = 1\ \ to\ \ i <= Numberofveh$ **do**
    for $ICID = 1\ \ to\ \ ICID <= Numofinter$ **do**
      if $location(VID) = Location(ICID)$ **then**
        Add VID to MICID; add VID to the members of this intersection cluster(ICID)
      end if
    end for
  end for
end for
for $t = anytime\ \ and\ \ t <= simulationtime$ **do**
  for $ICID = 1\ \ to\ \ ICID <= Numinter$ **do**
    while $VID \in CID$ and $VID < MCID(CID)$ **do**
      distance(i)=abs(dirclusedge(VID)-loc(VID))
      LT (VID) = distance(VID)/velocity(VID)
    end while
    ICH=VID (index (max (LT)))
    if $location(ICH) =$ *threshold point(ICH)* **then**
      update ICH
    else
      keep old ICH
    end if
  end for
end for

C. An Intersection Dynamic VANET Routing (IDVR) Protocol

When the packets arrive at the intersection cluster, the ICH determines the real-time optimal route that the packet is supposed to follow to reach the desired destination, taking into account the maximum of the minimum average throughput for all candidate routes (more details in section VI.B). The SDN provides the candidate routes between the current intersection and the destination intersection. In Figure 7, each candidate route has unique identification (R_{ID}), which consists of a series of intersections and the corresponding weight. The weight for each route is calculated by computing the average throughput (as in Equation 8) for each segment, and then selecting the minimum value. When there are no vehicles at the intersection cluster zone, the current CH follows the rule of store-and-forward, by storing the packets inside the CH buffer and continuing to move until it reaches another CH within its transmission range and closer to the destination intersection than itself.

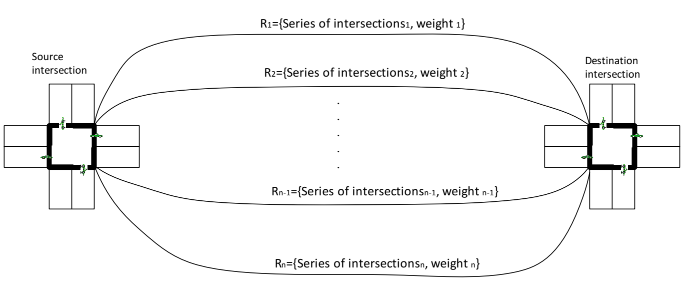


Fig. 7. set of candidate route

Algorithm 4 explains in pseudo code the IDVR protocol. In IDVR, each forwarder node (source node) obtains all possible routes to the desired destination and store them in specific buffer ($routeset$), as in line 2. Then it calculates the minimum number of intersections from itself to the desired destination and stores it in another buffer ($minseg$), as in line 3. To limit the routing search, first, we define a constraint to search only for routes located between a predefined minimum number of intersection ($minseg$) and a predefined maximum number of intersection ($maxseg$). The routes that pass successfully this constraint is stored in ($cons1valid$) buffer, as in line (5-10). Second, we check the routes validity in ($cons1valid$) buffer; all the segments for each route in ($cons1valid$) should be greater than a predefined specific threshold value. We assigned a binary value of one for each segment that has a throughput value that is greater than a predefined specific threshold value, and a binary value of zero for each segment that does not have a greater value than a predefined specific threshold value, as in lines (11-17). Finally, we multiply the binaries value for each route in ($cons1valid$). The routes that passed the previous two constraints will be stored in ($cons2valid$) buffer, as in line (18-22). To calculate the weight for each route, we calculate the average throughput for each segment within the route, then select the minimum average throughput value as the weight for the route. The route weight is stored in ($validrouteset$) buffer, as in line (23-31). The optimal route is the route that has the maximum route weight among ($validrouteset$), as in lines 32 and 33.

Each route consists of a series of segments. Let us consider that we have n routes, as follows:

$$SCSR = (R_1, R_2, ..., R_n) \tag{9}$$

Where:

$SCSR$: : Set of Candidate Shortest Route

R_n: Weight for route n

n: Maximum number of routes

To calculate the average throughput for each route, we should calculate the average throughput for each segment within the route, and then select the minimum average throughput (see Equation 10). Finally, a maximum of the minimum average throughput for SCSRs will be selected as the next segment of the selected route (as in Equation 11).

$$AVGR_{ID} = min(AvgS(1), AvgS(2), ..., AvgS(x)) \tag{10}$$

$$\text{optroute} = max(AVGR_1, AVGR_2, ..., AVGR_n) \tag{11}$$

Where:

R_{ID}: Route ID.

$AVGR_{ID}$: Average throughput of the route R_{ID}.

$AvgS(x)$: Average throughput of segment x in the R_{ID}.

x: Total number of segment in the route R_{ID}.

n: Total number of valid routes.

$optroute$: Optimal route.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

Algorithm 4 IDVR Algorithm

```
for t = anytime and t <= simulationtime do
    routeset = shortestroute(S, D)
    minseg = min(shortestroute(S, D))
    maxseg = minsegement + 2
    for i = 1 to maxsizeof(routeset) do
        if minseg < numseg(routeset(i)) < maxseg then
            cons1valid = routeset(i)
            numofsegcons1valid = sizeof(cons1valid)
        end if
    end for
    for j = 1 to numofsegcons1valid do
        if cons1valid(j) > threshold then
            cons1valid(j) = 1
        else
            cons1valid(j) = 0
        end if
    end for
    for y = 1 to sizeof(cons1valid) do
        if multiplication(cons1valid) = 1 then
            cons2valid(y) = cons1valid(y)
        end if
    end for
    for i = 1 to maxsizeof(cons2valid) do
        validrouteID = cons2valid(i)
        for j = 1 to sizeof(validrouteID) do
            weight(j) = avgthr(validrouteID(j))
        end for
        RouteIDweight(i) = Minimum(weight)
        routeID(i) = validrouteID
    end for
    validrouteset = (routeID, RouteIDweight)
    selectedroute = maximum(validrouteset)
    returnrouteID(selectedroute)
end for
```

V. Control Overhead Reduction Algorithm (CORA)

In VANET, the CBR protocols do not require every vehicle to know the entire topology information. Only the selected CH vehicles require to know the topology information and other CMs only require to periodically exchange their information with the CH. HELLO message is one kind of the control overhead messages that we discussing in this article. Any CM should inform the CH about its identity by sending a HELLO message, in addition it could combine other parameters such as current location, direction, velocity, and LT. The increasing size of HELLO messages is an important issue that degrade the performance of any mobile and limited networks resources . Furthermore, the frequently exchanging of HELLO message negatively impact the network performance.

Therefore, in this section we first propose a new algorithm called Control Overhead Reduction Algorithm (CORA), that aims to reduce the number of control overhead messages in a clustered topology. We then present a new design for HELLO message, by minimizing the number of parameters in HELLO message. CORA is based on the assumption that each vehicle in the VANET environment knows its current location and cluster ID by using a digital map and Global Positioning System (GPS). In the following section, we describe how CORA algorithm is able to minimize the HELLO messages between the CMs and the CH.

A. Exchanging of control messages

In general, each vehicle must be defined as CM or CH at any time. Algorithm 5 explains in pseudo code the CORA algorithm, initially, each vehicle enters any cluster coordination zone sets its status as CM by default. Then it should wait for τ second (lines 2 and 3), if it does not receive CHADS message, then it changes its status to CH and starts periodically (every τ second) to forward CHADS message (lines 9 and 10). If the CM receives the CHADS message then it stays as CM and replies with only one HELLO message. The replied message consists of the CM Identification and the remaining LT required to leave the cluster zone (lines 4 to 7). The remaining LT is varied among vehicle due to the velocity variation. The CHADS message consists of CH Identification and the remaining LT. The objective of periodically exchanging CHADS message is to inform new CMs arrival that an active CH exists. When the CH receives all replies from the CMs within its associated LT, the CH is capable to calculate the Candidate CH (CCH) before leaving the cluster. Therefore, the CMs do not require to periodically update their information with the CH. In other word, the HELLO messages produced by the CMs are proportional to the number of CH changes instead of specific period of time. This finally yields to significantly minimizing the control overhead messages in each cluster.

Algorithm 5 CORA protocol

```
1:  for t = anytime and t <= simulationtime do
2:      if anyvehicleenterstheclusterZone then
3:          vehiclestaus = CMandwaitτsec
4:          if CMreceivesCHadsmessage then
5:              vehiclestaus = CM
6:              replytoCHbyonemessage
7:              Containes < CMID, CMLT >
8:          else
9:              vehiclestaus = CH
10:             everyτsecsendCHads
11:         end if
12:     end if
13: end for
```

To calculate the number of CHADS message within the simulation time, we first divide the elected CH remaining LT time by the period of exchanging time τ (τ is a constant value). The results give us the number of CHADS message for one CH, as in Equation 12. Then to get the overall CHADS messages, we calculate the summation for all elected CHs in the same cluster within the simulation time, as in Equation 13.

$$AdsCH_{ijk} = \frac{CHLT_{ijk}}{\tau} \tag{12}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

Where:

$AdsCH_{ijk}$: The total number of CHADS messages produced from CH with ID i in cluster j in segment ID k.

$CHLT_{ijk}$: The remaining LT for CH with ID i in cluster ID j in segment ID k.

τ : The periodic exchanging time for CHADS message.

$$TotalAdsclus_{jk} = \sum_{i=1}^{x} AdsCH_{ijk}, \quad where \quad 0 < TotalAdsclus_{jk} < simulationtime \tag{13}$$

Where:

$TotalAdsclus_{jk}$: The number of CHADS message produced from CHs in cluster ID j in segment ID k.

x : The maximum number of elected CH within the simulation time for cluster j.

To calculate the total CHADS messages generated in a segment, we do the summation of the number of CHADS messages for each cluster, as follow:

$$\begin{aligned} TotalCHAds_{k} &= \sum_{j=1}^{y} TotalAdsclus_{jk} \\ &= \sum_{j=1}^{y} \sum_{i=1}^{x} \frac{CHLT_{ijk}}{\tau} \end{aligned} \tag{14}$$

Where:

$TotalCHAds_{k}$: The number of CHADS message produced from CHs in segment ID k

$CHLT_{ijk}$: The remaining LT for CH with ID i and Cluster ID j in segment ID k.

y : The maximum number of of clusters within the segment.

Since τ is constant value, then the number of CHADS messages produced by the CH are proportional to the CH LT value in each cluster.

In Figure 8, the CH forwards CHADS messages every τ seconds to all of its CMs until its LT expires. Each selected CH should periodically forward an CHADS messages to announce itself in the cluster zone. The vehicles A,B,C, and D are CMs that receive CHADS messages from the vehicle CH while its LT time does not expire.

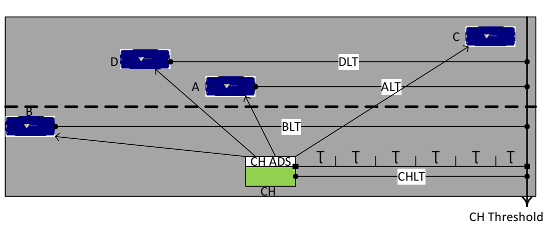


Fig. 8. CHADS message periodically (every τ seconds)

On the other side, when any vehicle enters the cluster zone, then its default status is CM. It should exchange the HELLO message with the CH. In this article the main contribution is to minimize the number of CM HELLO (CMHELLO) messages by taking into consideration CHLT. When any vehicle enters the cluster zone, it sends a CMHELLO message to the CH (if it receives the CHADS after τ second), there are then two scenarios: First, if the CM Life Time (CMLT) is greater than CHLT, the number of HELLO message equals to the number of CH changes within the CMLT plus two (the mandatory two HELLO messages when the CM enters the cluster and before leaves the cluster), else the CM will generate the CMHELLO message only two times, when it enters the cluster and before it leaves the cluster, and this is because CMLT is shorter than CHLT. Figure 9 explains a scenarios of exchanging the CMHELLO message; first, when vehicles enters the cluster zone(as vehicle B), then it should send CMHELLO message; and when the vehicles leave the cluster zone, it sends another CMHELLO message(as vehicle C). While the vehicles (vehicle A and D) already in the cluster zone and within the CHLT do not require to send any HELLO message. Figure 10 explains another scenario when the CH (Old CH) arrives to the threshold point (the point that the current CH should select another CH), the CH sends an CHADS message informing the CMs of the new CH. At this time all the CMs (vehicle A and D) should send the CMHELLO message to the new CH.

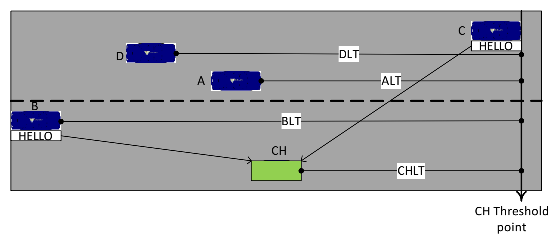


Fig. 9. CMHELLO message when enters and leaves the cluster

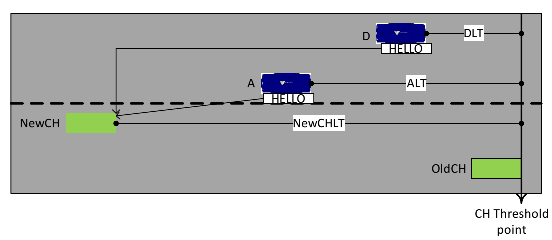


Fig. 10. CMHELLO message when new CH selected

The following Equation describes mathematically the two scenarios in Figure 9 and 10:

$$NumCM_{ijk} = \begin{cases} numCH_{ijk} + 2, & \text{if } CMLT_{ijk} > CHLT_{jk} \\ 2, & \text{if } CMLT_{ijk} \leq CHLT_{jk} \end{cases} \tag{15}$$

Where:

$NumCM_{ijk}$: The number of HELLO message produced by CM with ID i in cluster ID j in segment ID k.

$numCH_{ijk}$: The number of CH changes within $CMLT_{ijk}$.

$CMLT_{ijk}$: The remaining LT for CM with ID i in cluster ID j in segment ID k.

$CHLT_{jk}$: The remaining LT for current CH with in cluster ID j in segment ID k.

The following Algorithm explains how to calculate the overall CMHELLO messages in each cluster:

Algorithm 6 Total number of CMHELLO messages

```
total of CM ads = 0
for i = 1 to MaxnumberofCMs do
    if CMLT_i > CHLT then
        total of CM ads =2 + Num of CH changes Within CMLT_i
    else
        total of CM ads =2 + total of CM ads
    end if
end for
return total of CM ads
```

We can mathematically formulate the total of CMHELLO messages for specific cluster as the following Equation:

$$TotalHELLO_k = \sum_{j=1}^{y} NumCM_{ijk} \tag{16}$$

Where:

$TotalHELLO_k$: Total number of CMHELLO messages produced from CMs in cluster k.

y : Total number of CM in the cluster ID k.

Also, we can mathematically formulate the total of CMHELLO messages for a specific segment as the following Equation:

$$TotalCMHELLO_m = \sum_{j=1}^{p} TotalHELLO_j \tag{17}$$

Where:

$TotalCMHELLO_m$: The total number of HELLO message produced from CMs in segment ID m.

p : Total number of clusters in the segment ID m.

Finally, the total control overhead messages within the simulation time equal the summation of CMHELLO messages produced from the CMs and the periodical broadcasting of the CHADS messages produced by the CHs. As the following Equation:

$$TotalAdsmesage_k = TotalCMHELLO_k + TotalCHAds_k \tag{18}$$

B. *Designing of control overhead messages*

In this section, we propose a new design for CHADS messages and CMHELLO messages. In the literature, many researchers assume different sizes of control overhead message. Mohammad et al.[26] assume that the messages generated by the CH contains highway ID, CH ID, direction, specific weight value. In contrast, the CMHELLO messages are periodically broadcasted and contains CMID, highway ID, direction, position, and speed. Dan et al. [23] propose a new CBR protocol that groups the vehicle moving in the same direction in one cluster. The CMs sends periodically a CMHELLO message that contains vehicle ID, location, speed, and the direction of next intersection. Based on this information, an Algorithm is proposed to select the CH.

In Figure 11 and 12, we present the contents of CMHELLO and CHADS messages, respectively. The CMHELLO message consists of CMID and current CMLT (the time that the current CM requires till arrive at the threshold point), and the CHADS message consists of CHID, and current CHLT. An important point we have to mention is that the CH broadcasts the CHADS messages periodically (every τ second). While the CMHELLO messages are forwarded to the CH in three cases: First, when the CM enters the cluster zone. Second, when the CM leave the cluster zone. Third, when a new CH announces itself. Therefore, in the first contribution we minimize these messages to two parameter (vehicle ID, and vehicle LT) instead of many parameters mentioned in the literature, such as, location, speed, and direction. In the second contribution we optimize the number of CMHELLO messages to be forwarded only in three cases(when entering and leaving the cluster, and when CH changes), instead on exchanging in terms of time period.

CMHELLO message	
CMID	Current CMLT

Fig. 11. CMHELLO message

CHADS message	
CHID	Current CHLT

Fig. 12. CHADS message

VI. MATHEMATICAL MODEL

In this section, we present the theoretical analysis of LT in a cluster and the grid topology mathematical model design, as follow:

A. *Theoretical analysis of LT in a cluster*

In this section, we explain the theoretical analysis of the LT cost value that is used in CH election.

Each vehicle within its corresponding cluster periodically calculates the LT value. Therefore, let us assume a vehicle with ID 1 has a LT value equal to LT_1; LT_1 is the LT that the vehicle with ID 1 stays active until it reaches its corresponding threshold point (th). The LT value depends mainly on the speed and the vehicle location. If the location of vehicle ID 1 on the cluster is l_1, then the absolute distance between the vehicle and the the corresponding threshold point is denoted by $D_{l_1,th}$. $D_{l_1,th}$ is a random variable that takes values within $[0, d_{max}]$,where d_{max} is the maximum distance to the directional cluster edge. The maximum LT is calculated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

based on the maximum distance to the directional cluster edge and the minimum allowed speed on that cluster.

At the segment, the vehicles are moving only in two directions with one dimension (X or Y axis). let us assume that the segment is divided into fixed size clusters (as in Figure 4). For simplicity, we assume that the shape of the cluster is rectangular with a length of d_{max}, and that the velocity of the vehicles follows a uniform distribution. Therefore, the probability density function (pdf) and the Cumulative Distribution Function (CDF) of the velocity (v) are determined as in the following equations respectively:

$$p(v) = \begin{cases} 0 & \text{,if} \quad v < Min_v \\ \frac{1}{Max_v - Min_v} & \text{,if} \quad Min_v \leq v \leq Max_v \\ 0 & \text{,if} \quad v > Max_v \end{cases} \tag{19}$$

$$P(v) = \begin{cases} 0 & \text{,if} \quad v < Min_v \\ \frac{v - Min_v}{Max_v - Min_v} & \text{,if} \quad Min_v \leq v < Max_v \\ 1 & \text{,if} \quad v \geq Max_v \end{cases} \tag{20}$$

Where:

Min_v: Minimum allowed velocity in the cluster

Max_v: Maximum allowed velocity in the cluster

In order to transfer p_v in terms of time (t) in seconds, then we should multiply p_v by d_{max}/t^2, as in the following equation:

$$p(t) = \begin{cases} 0 & \text{,if} \quad t < \frac{d_{max}}{Max_v} \\ \frac{d_{max}}{(Max_v - Min_v)t^2} & \text{,if} \quad \frac{d_{max}}{Max_v} \leq t \leq \frac{d_{max}}{Min_v} \\ 0 & \text{,if} \quad t > \frac{d_{max}}{Min_v} \end{cases} \tag{21}$$

By assuming that each vehicle is equipped with GPS, then each vehicle is capable of determining the distance between its location and its corresponding threshold point. Then the LT_i equals the distance between vehicle i and the directional threshold of the cluster divided by the velocity of vehicle i (as in Equation 2). The Valid LT (VT) for any vehicle can be denoted as follows:

$$VT_i = \frac{d_{max} - d_{ith}}{V_i} \tag{22}$$

To obtain the probability value of the VT_i, we integrate the pdf of Equation 11 from $-\infty$ to VT_i as follows:

$$P(VT_i) = \int_{-\infty}^{VT_i} p(t) dVT$$

$$= \begin{cases} 0 & , VT_i < \frac{d_{max}}{Max_v} \\ \frac{d_{max}}{(Max_v - Min_v)} \left(\frac{Max_v}{d_{max}} - \frac{1}{VT_i} \right) & , \frac{d_{max}}{Max_v} \leq VT_i \leq \frac{d_{max}}{Min_v} \\ 1 & , VT_i > \frac{d_{max}}{Min_v} \end{cases} \tag{23}$$

The segment VT value is determined by multiplying the cluster VT from the first cluster adjacent to the intersection at the beginning of the segment to the last cluster adjacent to the intersection at the end of the segment. Therefore, the probability value of the Valid LT(VT) for the segment equals the multiplication of the CDF for all the clusters in the segment, as follows:

$$P_{seg}(VT) = P_1(VT_1) \times P_2(VT_2) .... \times P_n(VT_n), \\ where VT_n \in [\frac{d_{max}}{Max_v}, \frac{d_{max}}{Min_v}] \tag{24}$$

Where:

$P_{seg}(VT)$: Probability that the segment has Valid LT

VT_n: Valid LT of vehicle n

$P_n(VT_n)$:Probability that cluster n has valid LT n (VT_n)

In Table I, we present the numerical results for the probability of segment LT validity in terms of velocities and the size of clusters. Based in Equation 24, we calculate the probability that the segment has valid LT, when the segment is divided into different cluster sizes, different segment sizes (size of one,two,three, and four clusters), and different ranges of velocity.

B. The design of grid topology

In this section we design a grid topology that consists of a series of segments and intersections. In Figure 13, each segment and each intersection has a unique identification. Let us assume that the grid dimensions are n horizontal intersections and m vertical intersections; thereby, we have $n \times m$ intersections and $(n-1) \times (m-1)$ segments.

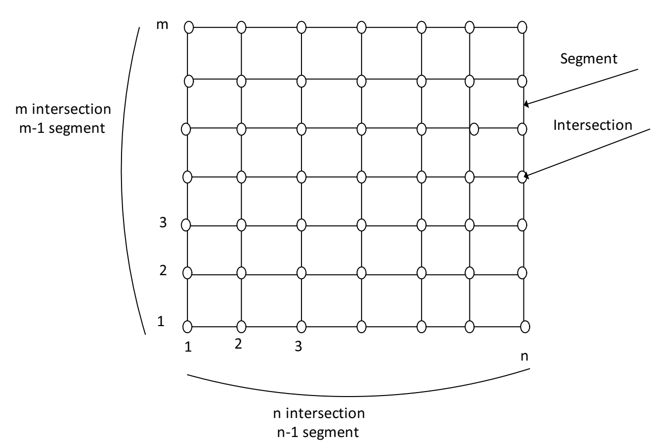


Fig. 13. Grid topology

The Intersections Set (IS) contains all intersection IDs in the topology, as follows:

$$IS = [1, 2, 3...., ((n \times m) - 1), (n \times m)] \tag{25}$$

Where:

$n \times m$:Maximum number of intersections in the grid topology

The Segments Set (SS) is defined by a new set that contains all segments in the grid topology, as follows:

$$SS = [S_1, S_2, ..., S_{max}] \tag{26}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

TABLE I
THE PROBABILITY OF SEGMENT LT VALIDITY

Range of velocity (Km/h)	Cluster Size=250m				Cluster Size=500m			
	1 clsuter	2 clusters	3 clusters	4 clusters	1 cluster	2 clusters	3 clusters	4 clusters
(10-30)	0.6731	0.4531	0.305	0.2053	0.6746	0.4551	0.3070	0.2071
(30-50)	0.5421	0.2939	0.1593	0.0864	0.5634	0.3174	0.1788	0.1008
(50-70)	0.5327	0.2838	0.1512	0.0805	0.5381	0.2896	0.1558	0.0838
(70-90)	0.3864	0.1493	0.0577	0.0223	0.4767	0.2272	0.1083	0.0516

Where:

S_i: Segment ID i

max: Maximum number of segments, which equal $m \times (n-1) + n \times (m-1)$

The SDN defines all candidate routes from the Source Intersection (SI) to the Destination Intersection (DI). We define two constraints for route validity, as follows:

1) The Number of Segments for each Candidate Route ($NSCR$) should be varied between the minimum number of segments between SI and DI and the maximum number of segments. NSCR falls within the following range:

$$NSCR = [minsegments, maxsegments] \tag{27}$$

Where:

$minsegment$: Minimum number of segments in the shortest route between SI and DI.

$maxsegment$: Maximum number of Segments between SI and DI.

2) To find the route validity for each segment in the SS, we define a binary variable that finds the route connectivity of each segment within its corresponding route. In Equation 28, C_i is the connectivity status of segment i .

$$C_i = \begin{cases} 1 & if avgthr(i) >= Thresholdvalue \\ 0 & otherwise \end{cases} \tag{28}$$

When all the segments are defined as valid within the route, the route is valid. In Equation 29, if the product of C_i is equal to 1, then this route is valid

$$\prod_{i=1}^{MNSCR} C_i = 1 \tag{29}$$

Where:

$MNSCR$: Maximum Number of Segments for the Candidate Route.

In Equation 30, each candidate route calculates the average throughput value which equals the minimum average throughput of the segment within the route

$$MinAvgSet = \begin{cases} Avgeragethroughput(P_1) \\ Avgeragethroughput(P_2) \\ \\ \\ Avgeragethroughput(P_{z-1}) \\ Avgeragethroughput(P_z)) \end{cases} \tag{30}$$

Where:

z: maximum number of valid candidate route

The optimal route based on our Algorithm is the maximum of the MinAvgSet, as in Equation 31:

$$Optimalroute = max(MinAvgSet) \tag{31}$$

At each intersection, the ICH forwards the packets to the first segment of the optimal route. At each intersection, the ICH dynamically recalculates the optimal route to the desired destination. Therefore, the throughput for the route to the desired destination is the product of the throughput for the first segment in the optimal route. Remember here that the optimal route is determined at each intersection in real-time. In Equation 32, the throughput in a grid topology is calculated by multiplying the average throughput (calculated in Equation 8) of the first segment in the current optimal route by the throughput of the first segment of the next optimal route at the next intersection, and so on, until arriving at the desired destination, as follows:

$$GThr = AvgThr_1 \times AvgThr_2 \times ... \times AvgThrt_n \tag{32}$$

Where:

$GThr$: the throughput for the route in a grid topology.

$AvgThr_n$: average throughput of first segment in the optimal route number n.

An end-to-end ($E2Edelay$) delay in a grid topology is the time that the packet takes to arrive at the destination, which is the commutative delay for the first segment of the optimal route at each intersection. In Equation 33, we calculate the $E2Edelay$ for any two vehicles in a grid topology.

$$E2Edelay = delay_1 + delay_2 + ... + delay_n \tag{33}$$

Where:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

$E2Edelay$: end-to-end delay between any vehicles in a grid topology

$delay_n$: end-to-end delay between two adjacent intersections of first segment in the optimal route number n.

VII. SIMULATION AND ANALYSIS

By using the SUMO version 0.28.0 traffic generator and Matlab version R2016b, we evaluate the performance of our proposed protocols in different scenarios and in terms of different performance metrics, as follows:

1) Evaluating the CBLTR protocol: To evaluate the proposed CBLTR protocol, we implemented a bidirectional segment of 1000 meter long and 20 meter wide. The segment is divided into fixed sizes of clusters of 250 meter length. We initially distributed 40 vehicles on the segment using uniform distribution, and we gave each vehicle a constant velocity randomly selected from predefined velocity ranges, as follows: 10-30km/h, 30-50km/h, and 50-70km/h. In Figure 14, we present the simulation results of CBLTR, CBDRP, AODV-CV, and CBVANET protocols, respectively, in terms of average throughput and speed range. We calculate the average throughput for the segment based on Equation 8, assuming that the transmission rate is 2 Mbps. The simulation time is 90 seconds, and we calculate the average throughput periodically (every 5 seconds). The results show that the CBLTR protocol significantly outperforms CBDRP, AODV, and CBVANET protocols in terms of throughput and link stability. CBVANET also improves the throughput compared with CBDRP and AODV-CV; however the link is not stable, since this protocol depends only on minimum velocity to elect the CH regardless of its location, thus increasing the number of CH re-election processes as well as reducing the CHLT. In addition, the CBLTR protocol maintains higher communication stability by increasing the velocity range compared to CBDRP, AODV-CV, and CBVANET protocols.

We compare the CBLTR protocol with CBVANET and CBDRP protocols. CBVANET selects the CH based on minimum node velocity, regardless of the distance to the directional cluster edge; and CBDRP selects the CHs based on the distance to the cluster center. The CBLTR protocol shows a significant improvement in average throughput compared to the CBDRP protocol. Note that CBDRP selects the CH based on location without considering mobility parameters. However, CBVANET improves the average throughput compared to CBDRP. But our proposed protocol CBLTR outperforms CBVANET in terms of average throughput and stability.

Based on Equation 8, we calculate the analytical solution for the optimal throughput that we can gain from the segment. Figure 15 shows the analytical solution of average throughput in terms of transmit time. We ran the simulation for 200 seconds. In addition we assumed speed range(10-60)km/h, and distributed 80 vehicles within the segment using poisson distribution with arrival rate 1 vehicle per second. The results start to monitor when all vehicles enter the segment. The figure clearly proves that our proposed protocol outperforms others protocols in terms of average throughput. In addition, the CBLTR protocol is closer to the optimal solution.

Cluster stability can be defined through different mechanisms, but the main mechanisms are CH duration and the number of CH changes. CH duration is the period of time that the CH maintaines its status as CH; maximizing CH duration is useful to improve cluster stability, as well as minimizing the control overhead that yields from frequent reelection processes. The number of CH changes is the number of vehicles that change its status from CH to CM within a period of time. The analysis shows that frequently changing CH minimizes network stability [27]. In contrast, the CBLTR protocol elects the CH in each cluster based on periodical calculation of the LT. The selected vehicle maintains and advertises its status as CH until it arrives at the predefined threshold point. In Figures 16 and 17, we ran the simulation for 500 seconds; then we calculated the average CH duration and the average number of CH re-election by comparing the CBLTR election algorithm with other CH election algorithms mentioned in the literature. We evaluate the performance in terms of average CH duration and average number of CH reelection processes. The results show that the CBLTR protocol outperforms other election algorithms in terms of average CH duration and the average number of CH changes.

TABLE II
SIMULATION PARAMETERS FOR IDVR PROTOCOL

Parameter	Value
Simulation time	1000 second
Topology type	Grid topology
Number of intersection	25
Number of segments	40
Communication range	250
Vehicle range speed	$(10-60)kmph$
Packet size	512byte
Data sending rate	2 Mbps
Number of clusters per segment	4
Threshold value	1 Mbps
Maximum number of segments of valid route	Minimum number of segments in a valid route + 2
Type of communication in the segments	CBLTR [24]

2) Evaluating the IDVR protocol: To evaluate the IDVR protocol, we assess the performance of the IDVR algorithm in terms of end-to-end delay and throughput for a grid topology. We compare our results with three other intersection routing protocols. The grid topology characteristic and our simulation parameters are presented in Table II. The simulation scenario uses a grid topology of $4000m \times 4000m$ area that consists of 25 intersections and 40 bidirectional segments. We used SUMO to generate mobility for 500 to 1000 vehicles randomly distributed by the simulation time, and the velocities are uniformally distributed to the vehicle within a range of 10 km/h to 60km/h. The type of routing protocol used in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2733380, IEEE Access

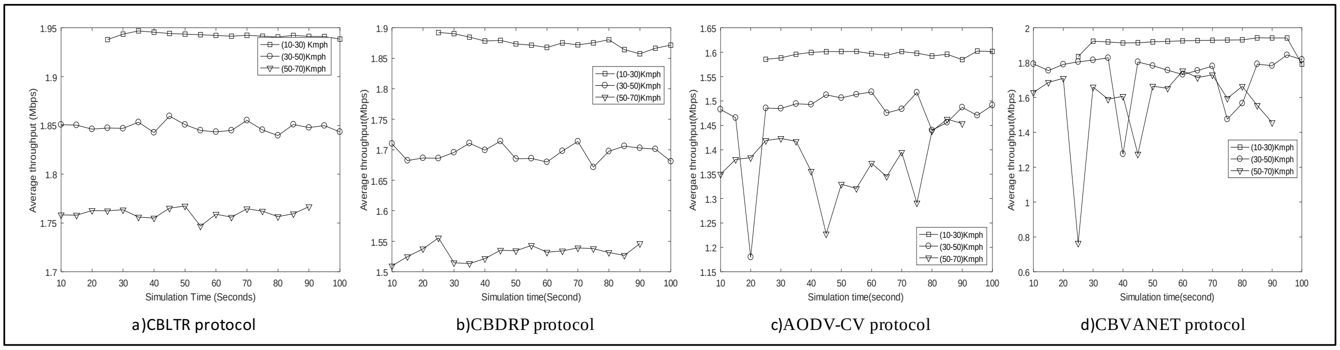


Fig. 14. Average throughput calculation for bidirectional segment, Figure (a) represents our proposed protocol CBLTR protocol, Figure (b) represents CBDRP protocol, Figure (c) represents AODV-CV protocol, Figure (d) represents CBVANET protocol

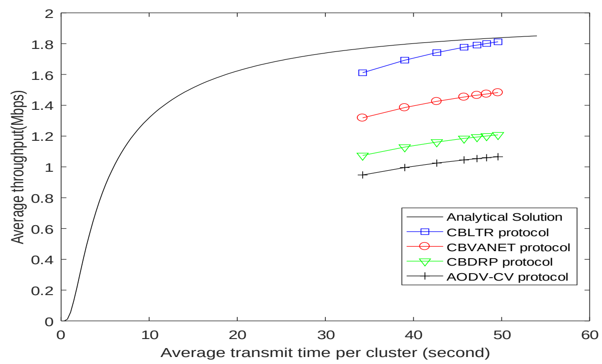


Fig. 15. Comparison between optimal throughput with simulation results

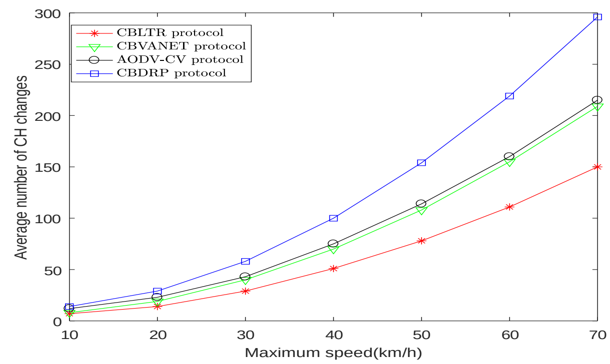


Fig. 17. Average number of CH changes vs speed

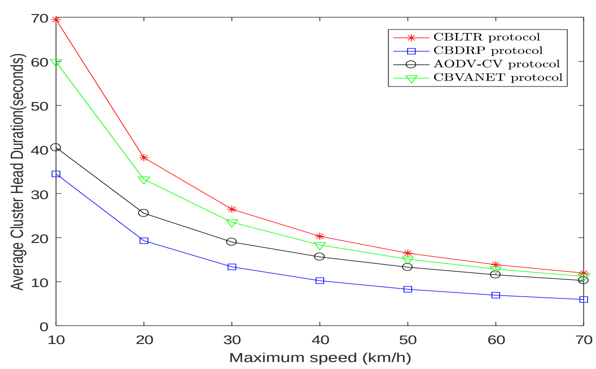


Fig. 16. Average Cluster Head Duration vs speed

segment is the CBLTR protocol [24]. CBLTR outperforms other routing protocols in terms of average throughput, by taking into account the maximum LT for selecting CH and the next forwarded nodes.

At the intersection, the IDVR protocol selects the next street based on the stability for the route between the source and destination. IDVR uses the CBLTR protocol to propagate the packet within the segment. In addition, it takes into account the route validity, so that all the segments within the selected route are connected and stable at each intersection. IDVR is a real-time dynamic protocol; each time the packet reaches the intersection, ICH recursively applies IDVR protocol between the current intersection and the desired destination intersection. In the literature, many researchers are investigating intersection routing protocols, such as VDLA, IRTIV, and GPCR. VDLA adopts sequential selection of intersections to construct the routes; the selection is based on real-time traffic density, the traffic load of the corresponding road segment, and the distance to the destination. IRTIV aims to find the shortest connected route to the destination in a city scenario, by taking into account the real-time segment density estimated by a completely distributed approach based on the periodic exchange of Hello messages. GPCR selects the next street that has a node with the shortest route to the destination.

In Figures 18 and 19, we compare the IDVR protocol, VDLA, IRTIV, and GPCR in terms of throughput and end-to-end delay based on Equations 32 and 33, respectively. As in our simulation results, we prove that the IDVR protocol significantly outperforms VDLA, IRTIV, and GPCR in terms of end-to-end delay and throughput.

3) Evaluating the CORA algorithm: To evaluate the performance of CORA protocol, we implemented a bidirectional highway scenario with length 10000 meters, then we divided the highway to fixed sizes of clusters of length 250 meter each. In Table III, we present the simulation parameter we used. The vehicles enters the highway scenario in fixed rate which equals 1 vehicle/sec. The simulation starts to gather the results after all vehicle enter the Highway scenario.

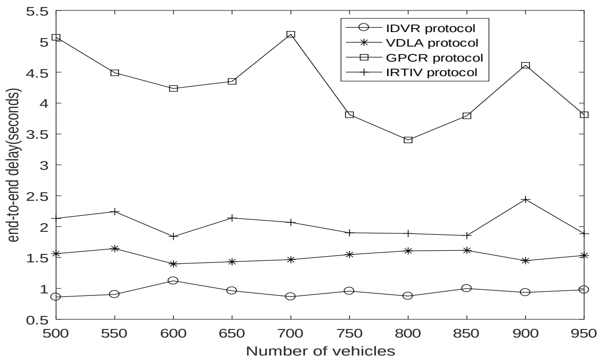


Fig. 18. End-to-end delay comparison in a grid topology

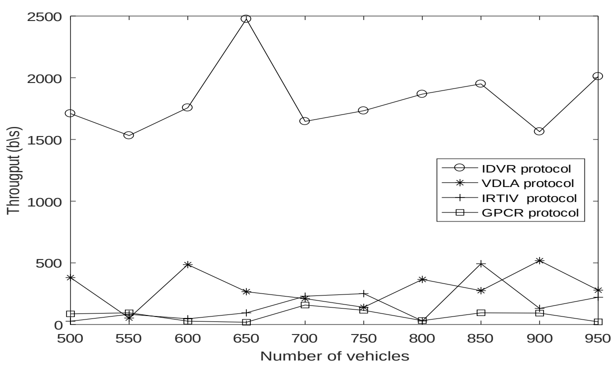


Fig. 19. Throughput comparison in a grid topology

TABLE III
SIMULATION PARAMETERS OF CORA ALGORITHM

Parameter	Value
Simulation time	500 second
Topology type	Highway scenario
Number of cluster	40
Number of vehicles in each direction	100
Vehicles arrival rate	1 vehicle/sec
Communication range	250
Vehicle range speed	$(10-60)kmph$
CH protocol used	CBLTR [24]

We calculate the number of CMHELLO messages in each cluster, to be more fair in our comparison we assumed that the vehicles use the same architecture of HELLO message, since in general we want to validate the CORA algorithm in terms of reducing the number of CMHELLO messages. In Figure 20, we compare our results with three other protocols mentioned in the literature; CBDRP, BRAVE, and MoZo protocols. In CBDRP protocol, the CMs in each cluster are updated very quickly, and this yields to produce many HELLO messages. In BRAVE protocol, the HELLO interval is 2 second. In MoZo protocol, the authors assume that the vehicles need to send HELLO updates messages when they deviate from their defined original moving function more than 5 m/s or the time from the last update which equals to 4 seconds.

CORA outperforms all previous protocols in terms of the number of CMHELLO message. CORA protocol minimizes the CMHELLO message by avoiding periodically exchanging of HELLO messages. CORA propagate the CMHELLO messages in three scenarios which are: when the CM enters the cluster zone, second; when the CM leave the cluster zone, and when new CH announces itself. In general, CORA calculate the optimal number of CMHELLO messages in each cluster.

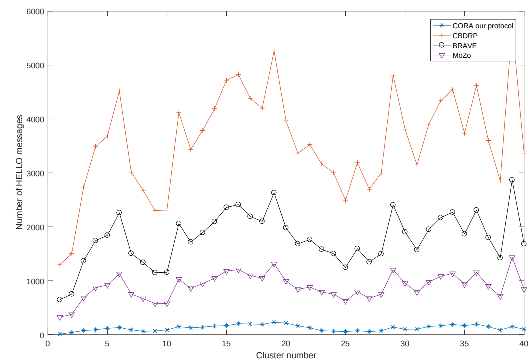


Fig. 20. Number of HELLO message in highway scenario

VIII. CONCLUSION

This article proposed three algorithms that improve the performance of CBR protocols in any VANET environment. First; a novel Cluster-Base Life-Time Routing (CBLTR) protocol in a segment topology is introduced. The CHs are elected based on maximum LT, and the re-election process is required only when the CHs reach their corresponding threshold point. Based on the simulation results, CBLTR protocol shows a significant improvement in terms of average throughput. The enhancement in CBLTR protocol is a new mechanism to select new CHs. The selected CHs have longer LT span making the protocol more stable.

Second; an Intersection Dynamic VANET Routing (IDVR) protocol in a grid topology is proposed. Each time the packet reaches the intersection, ICH recursively applies the IDVR protocol between the current intersection and the desired destination intersection, taking into account the stability of the connected route. The IDVR protocol selects the optimal route based on its current location, destination location, and a maximum of the minimum average throughput for SCSRs. IDVR increases the overall network efficiency, by increasing the route throughput, and decreasing end-to-end delay. As in our simulation, we have proved that the IDVR protocol outperforms VDLA, IRTIV, and GPCR in terms of end-to-end delay and throughput.

Finally; we proposed a Control Overhead Reduction Algorithm (CORA), which aims to reduce the control overhead messages in the clusters, by developing new mechanism for calculating the optimal period for updating or exchanging control messages between the CMs and the CH. CORA propagate the HELLO messages in three scenarios: when the

CM enters the cluster zone, second; when the CM leave the cluster zone, and when new CH announces itself. Based in the simulation results, CORA significantly minimized the number of HELLO messages in each cluster and in the segment with multiple clusters in general.

REFERENCES

[1] S. Tayal and M. R. Triphathi, "Vanet-challenges in selection of vehicular mobility model," in *2012 Second International Conference on Advanced Computing Communication Technologies*, Jan 2012, pp. 231–235.

[2] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments," *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1–51, July 2010.

[3] E. Spaho, M. Ikeda, L. Barolli, F. Xhafa, M. Younas, and M. Takizawa, "Performance of olsr and dsdv protocols in a vanet scenario: Evaluation using cavenet and ns3," in *2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications*, Nov 2012, pp. 108–113.

[4] B. Ding, Z. Chen, Y. Wang, and H. Yu, "An improved aodv routing protocol for vanets," in *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, Nov 2011, pp. 1–5.

[5] O. Abedi, R. Berangi, and M. A. Azgomi, "Improving route stability and overhead on aodv;routing protocol and make it usable for vanet," in *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, June 2009, pp. 464–467.

[6] X. Yu, H. Guo, and W. C. Wong, "A reliable routing protocol for vanet communications," in *2011 7th International Wireless Communications and Mobile Computing Conference*, July 2011, pp. 1748–1753.

[7] C. Wu, S. Ohzahata, and T. Kato, "Can we generate efficient routes by using only beacons? backbone routing in vanets," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2013, pp. 2929–2934.

[8] R. Chai, B. Yang, L. Li, X. Sun, and Q. Chen, "Clustering-based data transmission algorithms for vanet," in *2013 International Conference on Wireless Communications and Signal Processing*, Oct 2013, pp. 1–6.

[9] L. Liu, Z. Wang, and W. K. Jehng, "A geographic source routing protocol for traffic sensing in urban environment," in *2008 IEEE International Conference on Automation Science and Engineering*, Aug 2008, pp. 347–352.

[10] D. Luo and J. Zhou, "An improved hybrid location-based routing protocol for ad hoc networks," in *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, Sept 2011, pp. 1–4.

[11] T. Camp, J. Boleng, and L. Wilcox, "Location information services in mobile ad hoc networks," in *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*, vol. 5, 2002, pp. 3318–3324 vol.5.

[12] H. Saleet, R. Langar, K. Naik, R. Boutaba, A. Nayak, and N. Goel, "Intersection-based geographical routing protocol for vanets: A proposal and analysis," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 9, pp. 4560–4574, Nov 2011.

[13] T. Song, W. Xia, T. Song, and L. Shen, "A cluster-based directional routing protocol in vanet," in *2010 IEEE 12th International Conference on Communication Technology*, Nov 2010, pp. 1172–1175.

[14] A. Louazani, S. M. Senouci, and M. A. Bendaoud, "Clustering-based algorithm for connectivity maintenance in vehicular ad-hoc networks," in *2014 14th International Conference on Innovations for Community Services (I4CS)*, June 2014, pp. 34–38.

[15] B. Ramakrishnan, "Cbvanet: A cluster based vehicular adhoc network model for simple highway communication," 2011.

[16] M. Jerbi, S. M. Senouci, T. Rasheed, and Y. Ghamri-Doudane, "Towards efficient geographic routing in urban vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5048–5059, Nov 2009.

[17] C. Zhao, C. Li, L. Zhu, H. Lin, and J. Li, "A vehicle density and load aware routing protocol for vanets in city scenarios," in *2012 International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct 2012, pp. 1–6.

[18] O. S. Oubbati, N. Lagraa, A. Lakas, and M. B. Yagoubi, "Irtiv: Intelligent routing protocol using real time traffic information in urban vehicular environment," in *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*, March 2014, pp. 1–4.

[19] S.-C. Lo, Y.-J. Lin, and J.-S. Gao, "A multi-head clustering algorithm in vehicular ad hoc networks," *International Journal of Computer Theory and Engineering*, vol. 5, no. 2, p. 242, 2013.

[20] S. A. Mohammad and C. W. Michele, "Using traffic flow for cluster formation in vehicular ad-hoc networks," in *IEEE Local Computer Network Conference*, Oct 2010, pp. 631–636.

[21] Z. Y. Rawashdeh and S. M. Mahmud, "A novel algorithm to form stable clusters in vehicular ad hoc networks on highways," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 15, 2012. [Online]. Available: http://dx.doi.org/10.1186/1687-1499-2012-15

[22] P. M. Ruiz, V. Cabrera, J. A. Martinez, and F. J. Ros, "Brave: Beaconless routing algorithm for vehicular environments," in *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*, Nov 2010, pp. 709–714.

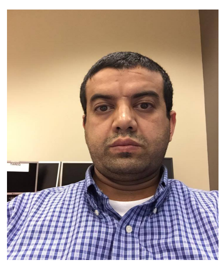
[23] D. Lin, J. Kang, A. Squicciarini, Y. Wu, S. Gurung, and O. Tonguz, "Mozo: A moving zone based routing protocol using pure v2v communication in vanets," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1357–1370, May 2017.

[24] A. Abuashour and M. Kadoch, "A cluster-based life-time routing protocol in vanet," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug 2016, pp. 213–219.

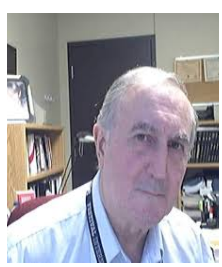
[25] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, and E. Cerqueira, "Towards software-defined vanet: Architecture and services," in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, June 2014, pp. 103–110.

[26] M. Hadded, R. Zagrouba, A. Laouiti, P. Muhlethaler, and L. A. Saidane, "A multi-objective genetic algorithm-based adaptive weighted clustering protocol in vanet," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 994–1002.

[27] S. Ucar, S. C. Ergen, and O. Ozkasap, "Vmasc: Vehicular multi-hop algorithm for stable clustering in vehicular ad hoc networks," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2013, pp. 2381–2386.



Ahmad Abuashour a received the B. Eng degree in computer engineering from Jordan University of Science and Technology, and the M.Eng in electrical engineering from Concordia University. He is currently working towards his Ph.D. degree in electrical engineering at École de technologie superieure (ETS),University of Quebec, Canada. His current research interests in Intelligent Transportation System (ITS),vehicular Ad hoc network, cluster-based routing protocol, network management and monitoring, and quality of service. He concentrate in vehicular Ad-Hoc NETwork (VANET) routing protocols.



Michel Kadoch (S86M91SM04) received the Ph.D. degree from Concordia University in 1992. He is currently a Full Professor with the Ecole de technologie superieure, University of Quebec, Montreal. He has authored a book entilted Protocoles et rseaux locaux : Accs Internet (PUQ Press, 2012). He has established the Research Laboratory Laboratoire de gestion informatique et de telecommunication which performs research on data communication networking. His current research stems from 5G, SON LTE, HetNet, WMN, and resource allocation to performance analysis. He served as a TPC Member and an Editor. He is also an Adjunct Professor with Concordia University, Montreal. He has been holding the position of the director of the M.Eng. program since 2001. He is supervising post doctorates, Ph.D. and master students, and undergraduate students with the Laboratory. His fields of interest are wireless ad hoc networks, network management, MPLS networks and VPN, quality of service, telecommunication protocols, wireless networking, traffic engineering, and performance analysis of telecommunication networks .