Control Overhead Reduction in Cluster-Based VANET Routing Protocol

Ahmad Abuashour(✉) and Michel Kadoch

Department of Electrical Engineering, École de Technologie Suprieure, 1100 Notre-Dame Street West, Montreal, QC H3C 1K3, Canada
ahmad.abuashour.1@ens.etsmtl.ca, michel.kadoch@etsmtl.ca

Abstract. Vehicular Ad-Hoc NETworks (VANETs) are unique form of Mobile Ad-Hoc NETworks (MANETs), where the nodes act as vehicles moving with relatively high mobility, and moving in a predefined routes. The mobility in VANETs causes high topology changes and in turn leads to excessive control overhead and frequent link communication failures. Traditionally, clustering techniques have been used as the main solution to reduce the control overhead messages in VANET, in which the network is divided into multiple clusters and selecting one of the Cluster Members (CMs) as a Cluster Head (CH). Still, a problem occurs when the control overhead messages increase due to periodically forwarding of CM HELLO (CMHELLO) messages between the CMs and the CH, and when the CH periodically broadcasts an CH advertisement (CHADS) messages to declare itself to the CMs. In this paper, we propose a Control Overhead Reduction Algorithm (CORA) which aims to reduce the control overhead messages in a clustered topology. Therefore, we develop a new mechanism for calculating the optimal period for updating or forwarding the CMHELLO messages between the CMs and the CH. Finally, we evaluate the performance of our proposed work by comparing with other recent researches that published in this field. Based on the simulation results, the CORA algorithm significantly reduces the CMHELLO messages, where it generates the minimum percentage of CMHELLO messages compared with other techniques proposed on this field.

Keywords: CH · CM · CMHELLO · CHADS · VANET MANET · CORA

1 Introduction

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), or/and 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 long battery life.

Y. Zhou and T. Kunz (Eds.): AdHocNets 2017, LNICST 223, pp. 106–115, 2018.
https://doi.org/10.1007/978-3-319-74439-1_10

The Cluster-Based Routing (CBR) protocol combines the features of both proactive [1] and reactive [2] routing protocol. The nodes in the clustered network are grouped together in a particular area called clusters. 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, clustering management and frequent CH elections increase the clustered control overhead. The clustered control overhead messages are produced by forwarding or broadcasting of control messages between the CMs and the CH, and these massages are classified into CMHELLO messages and CHADS messages, respectively. When the generated control overhead messages are increasing in a cluster topology, then the available bandwidth resources are decreasing. The main objectives of this paper is minimizing the number of generated clustered control overhead messages in a cluster-based VANET topology. Therefore, we propose a Control Overhead Reduction Algorithm (CORA) that optimizes the updating or forwarding period of CMHELLO messages in a clustered-based topology.

This paper is outlined as follows; in Sect. 2, we present a literature review that related to control overhead reduction techniques in clustered topology. Section 3 presents the CORA algorithm in a clustered highway scenario. Section 4 shows the simulation, results and analysis. Finally, Sect. 5 concludes this article, respectively.

2 State of Arts

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

VANETs are an autonomous systems formed by connected vehicles without the need for 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 of control overhead message. Therefore, the excessive amount of control overhead messages yield to consume high amount of available bandwidth resources.

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 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. The process of electing CH is out of scope in this paper. In general, dividing the network into multiple clusters reduces the communication overhead and improves the network efficiency.

In the cluster, CMs and CH should periodically exchange the control overhead messages. The CMHELLO message is one of important control overhead messages that 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 CMHELLO messages as follows:

Tao et al. [3] proposed a Cluster-Based Directional Routing Protocol (CBDRP) for highway scenario. The CMs exchange the control overhead packets that contain the location of the cluster, location of vehicle, and the velocity of the vehicle. A CH distributed algorithm is used to select one CH among CMs, the selected CH has full information about its CMs. CBDRP concentrates to reduce the routing overhead packet from source to destination, without considering the control overhead packets that produced by the CMs in each cluster.

Pedro et al. [4] proposed a Beacon-less Routing Algorithm for Vehicular Environment (BRAVE), the proposed protocol objects to reduce the control overhead messages in a 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 that each vehicle participates in the routing protocol still required to exchange a beacon message among them. In the simulation setting, BRAVE sets the exchanging time of the beacon message to 2 s to keep monitoring the vehicles location. In general, reactive routing protocol reduce the control overhead messages compared with proactive routing protocol, however it still suffering of high control overhead compared with CBR protocols.

Dan et al. [5] 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 is varied between the changing of moving function of 5 m/s or 4 s. In [6], a periodically live message is broadcasted by every node for announcement of it's existence in the cluster. Also, this paper does not consider the live message size and the period of updating these messages in its evaluation.

In the literature, the authors do not provide any guidelines to exploit the cluster resources. Though the main properties of any clustering algorithm are high CMHELLO message and CHADS messages. In addition, most of the literature ignoring the control overhead message size. To the best of our knowledge there are no researches that investigated to reduce the control overhead message by optimizing the control overhead exchanging period time for the CMs and CH. Therefore, in this paper we propose a Control Overhead Reduction Algorithm (CORA) that optimizes the updating period of CMHELLO messages in each cluster.

3 Control Overhead Reduction Algorithm

In VANET, the CBR protocols do not require that every vehicle knows 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 via CMHELLO messages. CMHELLO message is one kind of the control overhead messages that we discuss in this paper. The CMHELLO messages inform the CH about CM identity and it could combine other parameters; such as current location, direction, velocity, and life time. The increasing size of CMHELLO messages consider an important issue that degrade the performance of any mobile and limited resources networks. Furthermore, the frequently exchanging of CMHELLO message negatively impact the network performance. Therefore, in this section we propose a new algorithm that reduces the number of control overhead messages, which called CORA algorithm. CORA is based on the assumption that each vehicle in the VANET environment can know its current location and cluster ID by using a digital map and Global Positioning System (GPS). Also we used Cluster-Based Life-Time Routing (CBLTR) protocol [7], which outperforms many other cluster protocols in terms of increasing the average throughput and stability in clustered network. The CBLTR [7] protocol selects the CH based on maximum LT among the CMs and the CH maintains it's status as CH until arrives to a predefined threshold point. Therefore, the CBLTR protocol significantly reduced the CH election processes.

In general, each vehicle must be defined as CM or CH at any time. Algorithm 1 explains the CORA algorithm as follows; initially, each vehicle enters any cluster coordination zone sets its status as CM by default (lines 2 and 3). Then, it waits for τ second (line 4), if it does not receive any message, it changes its status to CH and starts periodically (every τ second) forwarding CHADS message (lines 10 and 11), this message consists of CH identification information and the remaining LT that the CH predict to spend in the cluster zone. Otherwise, it stays as CM and replies with only one CMHELLO message which consists of the CM identification and the remaining LT that the CM predict to spend in the cluster zone (lines 5–8). The remaining LT is varied among vehicle due to the velocity variation. The objective of periodically exchanging CHADS message is to inform newly-arrived CMs that an active CH exist. When the CH receives all replies from the CMs within its

Algorithm 1. CORA PROTOCOL

```
1: for t = 1 to end of simulation time do
2:     if any vehicle enters the cluster Zone then
3:         vehiclestaus = CM
4:         wait τ sec
5:         if CM receives CH ads message then
6:             reply to CH by CMHELLO message
7:             Containes < CMID, CMLT >
8:         else
9:             vehicle staus = CH
10:            every τ sec send CH ads message
11:        end if
12:    end if
13: end for
```

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 while the CHLT is not expired. In other word, the CMHELLO messages that produced by the CMs are proportional to the number of CH changes instead of specific period of time. Thus, that yields to significantly minimize the control overhead messages in each cluster.

To calculate the number of CHADS message within the simulation time, first we divide the elected CH remaining LT time by the period of exchanging time τ (τ is a constant value), as in Eq. 1:

$$AdsCH_{ijk} = \frac{CHLT_{ijk}}{\tau} \tag{1}$$

where:

$AdsCH_{ijk}$: 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 j in segment ID k.

τ: The periodic exchanging time for CHADS message. Next, we calculate the overall CHADS messages for all elected CHs in the same cluster within the simulation time, as in Eq. 2:

$$TotalAdsclus_{jk} = \sum_{i=1}^{x} AdsCH_{ijk}, 0 < TotalAdsclus_{jk} < simulationtime \tag{2}$$

where:

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

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

$$TotalCHAds_k = \sum_{j=1}^{y} TotalAdsclus_{jk} = \sum_{j=1}^{y}\sum_{i=1}^{x} \frac{CHLT_{ijk}}{\tau} \tag{3}$$

where:

$TotalCHAds_k$: The number of CHADS message produced by CHs in segment ID k.

y: Total number of clusters within the segment.

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

In Fig. 1, 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 from the CH while its LT time does not expire.

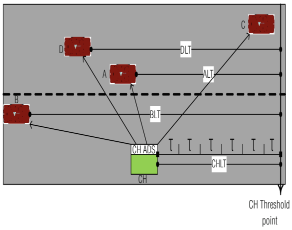

Fig. 1. CHADS message

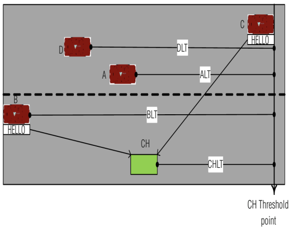

Fig. 2. CMHELLO enters and leaves the cluster

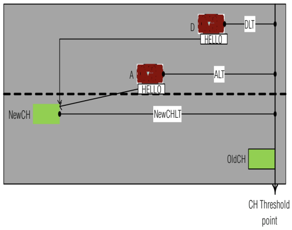

Fig. 3. CMHELLO when new CH selected

On the other hand, when any vehicle enters the cluster zone, its default status is CM. It should exchange the CMHELLO message with the CH. So, in this paper the main contribution is to minimize the number of 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). In this case we have two scenarios; if the CMLT is greater than CHLT, then the number of CMHELLO message equals to the number of CH changes within the CMLT plus two (the mandatory two CMHELLO messages when the CM enters the cluster and before leaves the cluster), otherwise; the CM generates the CMHELLO message only two times; that is when it enters the cluster and before leaves the cluster. Figure 2 explains a scenario of CM CMHELLO message; first, when vehicles enter the cluster zone (as vehicle B), then it should send CMHELLO message, and when the vehicle leaves the cluster zone, then it sends another CMHELLO message (as vehicle C), whereas the vehicles (vehicle A and D) that already in the cluster zone and within the CHLT do not require to send any CMHELLO message. Figure 3 explains another scenario when the CH (Old CH) arrives to the threshold point (the point that the current CH should select another CH), the old CH sends an CHADS message informing the CMs for the new CH, in the meantime; all the CMs (vehicle A and D) should send the CMHELLO message to the new CH.

The following Equation describes mathematically the two scenarios in Figs. 2 and 3:

$$NumCM_{ijk} = \begin{cases} numCH_{ijk} + 2, & \text{if } CMLT_{ijk} > CHLT_{jk} \\ 2, & \text{if } CMLT_{ijk} \leq CHLT_{jk} \end{cases} \tag{4}$$

where:

$NumCM_{ijk}$: The number of CMHELLO 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 i in cluster ID j in segment k.

We can mathematically formulate the total of CMHELLO messages for a specific cluster by following expression:

$$TotalHELLO_k = \sum_{j=1}^{y} NumCM_{ijk} \quad (5)$$

where:

$TotalHELLO_k$: The 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 specific pre-divided cluster segment as the following Equation:

$$TotalCMHELLO_m = \sum_{j=1}^{p} TotalHELLO_j \quad (6)$$

where:

$TotalCMHELLO_m$: The total number of CMHELLO 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 that produced from the CMs and the periodically CHADS messages that produced by the CHs. As the following Equation:

$$TotalAdsmessage_k = TotalCMHELLO_k + TotalCHAds_k \quad (7)$$

4 Simulation, Results, and Analysis

By using the SUMO version 0.28.0 traffic generator and Matlab version R2016b, we implement and evaluate our proposed protocol. In Table 1, we present the simulation parameter we used to evaluate the performance of our proposed work.

We first implemented a bidirectional highway scenario with length 10000 m, then we divided the highway to fixed sizes of clusters of length 250 m each. The vehicles enters the highway scenario in fixed rate which equals 1 vehicle/sec, when any vehicle arrives any end of the highway, it makes a U turn and drives back in the opposite direction. The SUMO traffic generator keeps safety distance between the vehicles, and the distance distribution between the vehicle follow an exponential distribution. All the vehicles remain in the highway until the end of the simulation. The simulation starts to gather the results after all vehicle entering the Highway scenario.

Based on Eq. 6, we calculate the number of CMHELLO messages in each cluster, we assumed here that the vehicles use the same architecture of CMHELLO message in terms of size. In Fig. 4, we compare our results with three other

Table 1. Simulation parameters

Parameter	Value
Simulation time	500 s
Topology type	Highway
Number of cluster	40
Number of vehicles in each direction	200
Vehicles arrival rate	1 vehicle/sec
Communication range	250
Vehicle range speed	(10–60) kmph
CH protocol used	CBLTR

Table 2. The mean and percentage of HELLO messages generated by the CMs

Protocol Name	Mean	Percentage of HELLO messages
CORA	215.25	2.5%
MoZo	1215.8	13.9%
BRAVE	2431.6	27.8%
CBDRP	4863.3	55.8%

protocols that mentioned in the literature; CBDRP, BRAVE, and MoZo protocols. In CBDRP protocol, the CMs in each cluster are updated vary quickly, and this yields to produce high CMHELLO messages. In BRAVE protocol, the CMHELLO interval is 2 s. In MoZo protocol, the authors assume that the vehicles need to send CMHELLO 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 s. CORA outperforms all previous protocols in terms of the number of CMHELLO messages that generated in each cluster within a period of time. The CORA protocol minimizes the number of CMHELLO messages due to avoid periodically exchanging of CMHELLO message, CORA propagate the CMHELLO messages in three scenarios; which are: when the CM enters the cluster zine, second; when the CM leave the cluster zone, and when new CH announces about itself. In general, CORA calculate the optimal number of CMHELLO messages in each cluster.

In Table 2, we present a numerical results to validate the performance of the CORA protocol. Column 2 calculates the average number of CMHELLO messages that generated within the simulation time by CORA, MoZo, BRAVE, and CBDRP protocols. Column 3 calculates the percentage number of HELLO messages that generated by the CMs, the percentage is calculated by dividing the average number of CMHELLO messages that generated by any algorithm to the overall CMHELLO messages that generated by all algorithms. The CORA algorithm significantly reduces the CMHELLO messages, where it generates the minimum percentage of CMHELLO messages, which is equal to 2.5%, and the main reason of that returns to forward the CMHELLO messages only in three scenarios that we explained in the previous section. In contrast, MoZo, BRAVE, and CBDRP algorithms, show high number of CMHELLO messages, and the reason of that because all of these protocols forward periodically the CMHELLO messages.

We evaluate the performance of CORA algorithm in terms of the total number of control overhead messages. Based in Eq. 7, the total of number of control

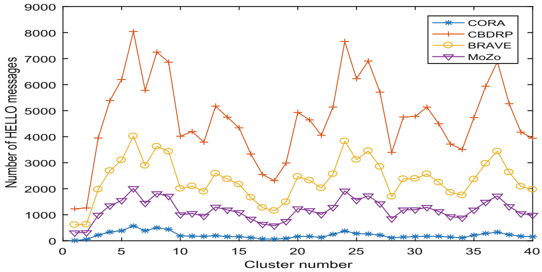

Fig. 4. Number of CMHELLO message in highway scenario

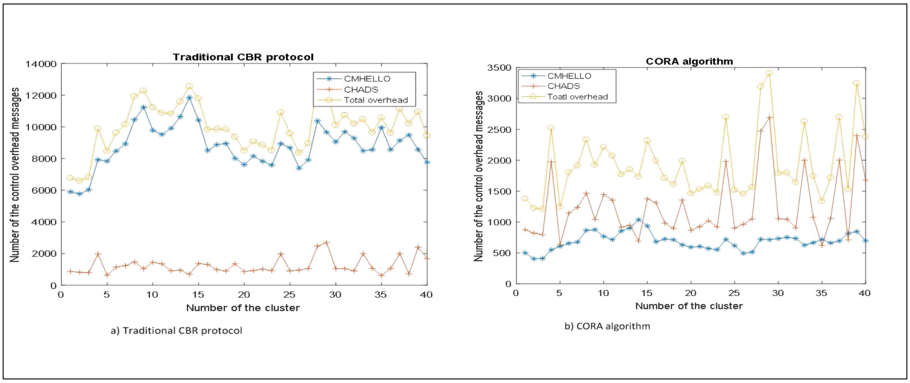

Fig. 5. CORA vs traditional CBR protocol

messages are the summation of all messages that forwarded by the CMs and broadcasted by the CHs in a specific period of time. In Fig. 5, we present the total number of control overhead for the CORA algorithm and another traditional CBR protocol (such as CBDRP). As shown in Fig. 5a, in traditional CBR protocol all the vehicles in the clusters should forwards or broadcasts the control overhead messages periodically and depending mainly on time. Therefore, an excessive amount of generated control overhead messages are produced in a traditional CBR protocols. In contrast, Fig. 5b shows that CORA algorithm achieves a significant reducing of CMHELLO messages, and the reason of that because the CM only forward CMHELLO messages only in three cases; when the CMs enters or leaves the cluster zone or CH election process notification received. In other words, the CMs mainly depend on the location to forwards the CMHELLO message rather than the times.

5 Conclusion

In this paper, we propose a Control Overhead Reduction Algorithm (CORA), which aims to reduce the number of CMHELLO messages that generated by the CMs in the clusters, a new mechanism for calculating the optimal period for updating or exchanging CMHELLO messages is proposed. CORA propagate the CMHELLO messages in three scenarios: when the CM enters the cluster zone, second; when the CM leave the cluster zone, and when new CH elected. Based in the simulation results, CORA significantly minimized the number of CMHELLO message in each cluster and in any segment in general.

References

1. Spaho, E., Ikeda, M., Barolli, L., Xhafa, F., Younas, M., Takizawa, M.: 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, pp. 108–113, November 2012
2. Ding, B., Chen, Z., Wang, Y., Yu, H.: An improved AODV routing protocol for VANETS. In: 2011 International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–5, November 2011
3. Song, T., Xia, W., Song, T., Shen, L.: A cluster-based directional routing protocol in VANET. In: 2010 IEEE 12th International Conference on Communication Technology, pp. 1172–1175, November 2010
4. Ruiz, P.M., Cabrera, V., Martinez, J.A., Ros, F.J.: Brave: Beacon-less routing algorithm for vehicular environments. In: The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010), pp. 709–714, November 2010
5. Lin, D., Kang, J., Squicciarini, A., Wu, Y., Gurung, S., Tonguz, O.: MoZo: a moving zone based routing protocol using pure V2V communication in VANETs. IEEE Trans. Mob. Comput. **16**(5), 1357–1370 (2017)
6. Singh, S., Rajpal, N., Sharma, A.: Address allocation for MANET merge and partition using cluster based routing. SpringerPlus **3**(1), 605 (2014). http://dx.doi.org/10.1186/2193-1801-3-605
7. Abuashour, A., Kadoch, M.: A cluster-based life-time routing protocol in VANET. In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 213–219, August 2016