MoZo: A Moving Zone Based Routing Protocol Using Pure V2V Communication in VANETs

Dan Lin, Jian Kang, Anna Squicciarini, Yingjie Wu, Sashi Gurung, and Ozan Tonguz

Abstract—Vehicular Ad-hoc Networks (VANETs) are an emerging field, whereby vehicle-to-vehicle communications can enable many new applications such as safety and entertainment services. Most VANET applications are enabled by different routing protocols. The design of such routing protocols, however, is quite challenging due to the dynamic nature of nodes (vehicles) in VANETs. To exploit the unique characteristics of VANET nodes, we design a moving-zone based architecture in which vehicles collaborate with one another to form dynamic moving zones so as to facilitate information dissemination. We propose a novel approach that introduces moving object modeling and indexing techniques from the theory of large moving object databases into the design of VANET routing protocols. The results of extensive simulation studies carried out on real road maps demonstrate the superiority of our approach compared with both clustering and non-clustering based routing protocols.

Index Terms—Moving zone, vehicle clustering, MoZo routing protocol, VANETs

1 INTRODUCTION

Vehicular Ad-hoc Networks (VANETs) enable vehicles to communicate with one another and create a large network with vehicles acting as the network nodes. Considering the huge number of vehicles (hundreds of millions worldwide on the road on a daily basis), the benefits of VANETs would be tremendous. Various types of information (e.g., traffic conditions, advertising news and e-coupons) can be shared among vehicles via VANETs as long as minor delays are acceptable in the specific applications of interest. For example, a vehicle can send inquiries to vehicles around certain landmarks to obtain up-to-date parking information. Another interesting emerging application, called Infotainment, provides multimedia services to subscribed vehicles in a particular location by using vehicle-to-vehicle (V2V) communication.

A key requirement for the realization of VANET applications is the availability of efficient and effective routing protocols for message dissemination. Without well-defined and efficient routing protocols, vehicles may be unable to share important messages and enjoy the benefits of the advanced technologies offered by VANETs. To address these issues, many VANET routing protocols have been proposed. Broadly, these existing protocols can be classified into five main categories, namely broadcasting protocols [1], route-discovery protocols [2], [3], [4], position-based protocols [5], [6], clustering-based protocols [7], [8] and infrastructure-based protocols [9]. While effective for specific applications and contexts, these protocols are still limited in their applicability and practical use. The broadcasting protocols rely on large message dissemination, and hence may cause a high communication overhead and message congestion on the network. To prevent this, broadcast storm mitigation techniques have been proposed [10]. The route-discovery protocols require to discover a route before sending out a message, and hence may not be suitable for applications with strict time constraints. The position-based protocols require vehicles to pass messages to nearby vehicles moving towards the final destination of the message. Such protocols require each vehicle to maintain information about neighboring vehicles, resulting in frequent message exchange between each pair of vehicles, and hence their overall communication cost is typically higher than clustering-based protocols which arrange vehicles into clusters and only need the cluster heads to maintain neighboring information. The infrastructure-based routing protocols heavily rely on road-side units (RSUs) which are currently not widely available and have experienced a very slow deployment rate due to their high cost.

Among all types of protocols, clustering-based protocols appear to be the most promising one as they attempt to capture the mobility of VANET nodes in a natural way and provide relatively stable units (i.e., the clusters of vehicles) for communication. However, most of existing clustering-based approaches [8], [11], [12] focus on how to cluster vehicles but do not provide the follow-up routing strategies. There lacks study on whether the expected improvement on routing efficiency can offset the overhead (i.e., computing delay, amount of message exchanged for clustering) incurred by gaining stable clusters. For example, if forming stable clusters of vehicles requires significant more message exchanges

- *D. Lin and S. Gurung are with the Department of Computer Science, Missouri University of Science & Technology, Rolla, MO 65409. E-mail: {lindan, sgy99}@mst.edu.*
- *J. Kang and Y. Wu are with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian Province, China. E-mail: {n130320039, yjwu}@fzu.edu.cn.*
- *A. Squicciarini is with the College of Information Sciences & Technology, Pennsylvania State University, University Park, PA 16801. E-mail: asquicciarini@ist.psu.edu.*
- *O.K. Tonguz is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: tonguz@ece.cmu.edu.*

Manuscript received 18 Nov. 2015; revised 24 June 2016; accepted 2 July 2016. Date of publication 19 July 2016; date of current version 31 Mar. 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2016.2592915

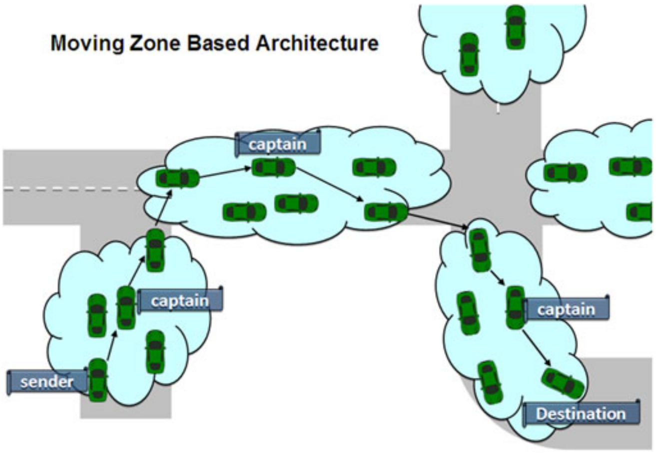


Fig. 1. Moving zone based architecture in VANETs.

than simply delivering messages without using clusters, such clustering may not be useful in practice.

In this paper, we propose a comprehensive routing solution that delivers messages in VANETs via a self-organized moving-zone based architecture formed using pure vehicle-to-vehicle communication. We compare our proposed routing protocol with both clustering-based approaches and non-clustering based approaches to demonstrate the advantages of our approach. Fig. 1 illustrates an overview of the key concepts behind our proposal, where the cloud symbol denotes moving zones and arrows indicate the message propagation route. Our approach integrates moving object modeling and indexing techniques [13] to vehicle management. Moving object techniques allow us to provide a realistic cluster-based representation, in that vehicles are grouped together according to their actual moving patterns. Further, the use of indexes allow for efficient movement information storage and management. Specifically, our approach reduces the update frequency since vehicles no longer need to periodically send location updates to the cluster head (called "captain vehicle"). Instead, vehicles just need to update their movement functions when their moving direction or speed change dramatically. Second, unlike cluster heads in other existing protocols, the captain vehicle in our protocol has the ability to estimate vehicle positions in the near future so that decisions (e.g., zone splitting, message routing) can be made without requiring constant location updates from member vehicles. Third, the use of index reduces the need for the captain vehicle to contact and examine every member vehicles for each event or operation since information of vehicles affected by the event can be quickly accessed via the index. As demonstrated by our experimental results, our approach significantly reduces the existing routing protocols' communication overhead to 1/10 while providing higher delivery rates.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 presents moving zone construction, Section 4 introduces the routing protocols, and Section 5 discusses zone maintenance. Section 6 reports the experimental results. Finally, Section 7 concludes.

2 Related Work

Many types of routing protocols have been proposed for VANETs, as surveyed in [14], [15], [16]. Since our work is closely related to clustering-based approaches, in this section we first briefly review works under this category. Then, we will discuss two approaches (CBDRP [17] and BRAVE [18]) in more details since they have been selected for comparison in our experimental study.

2.1 Clustering-Based Approaches

Although VANETs share some similar features with Mobile Ad-hoc Networks (MANETs), clustering techniques for MANETs cannot be directly applied to VANETs. Fan et al. [19] attempted to adapt MANET algorithms to VANETs. However, such adaptation still cannot address the unique characteristics that VANETs possess [20]. For example, energy is no longer an issue in VANETs and vehicles have high mobility making network topology highly dynamic.

As for VANETs, one of the earliest vehicle clustering algorithm is proposed by Kayis and Acarman [21]. The proposed passive clustering algorithm conducts clustering only when data is to be communicated. The clustering is based on predefined speed intervals such as [0, 30 mph] and [30 mph, 45 mph]. Vehicles traveling within the same speed interval form a cluster, and the vehicle which first claim to be the cluster head becomes the cluster head. However, the speed interval is not sufficient to capture the similarity in mobility. For example, two vehicles with very similar speeds 29 and 31 mph are grouped in different clusters. Moreover, this approach does not consider location proximity either. Vehicles which are very close to one another may stay together for certain period of time even they travel at different speeds. In [22], Chen et al. use only the distance between vehicles as the clustering criteria in that vehicles close to one another are grouped in the same cluster. Further, their approach relies on *a central server* to handle cluster merging and splitting events, while our approach is fully decentralized. In [23], Want et al. proposed a priority-based clustering algorithm. Each vehicle calculates its priority according to its estimated travel time and speed deviation. A vehicle having longer travel time and less speed deviation will have higher priority, and have higher chance to become the cluster head. Each vehicle shares its cluster information with its neighbors. This approach requires continuous communication among vehicles. The authors do not discuss how to take advantage of these clusters for routing, and hence it is not clear whether the overhead introduced by clustering will be offset during information routing. In [7], Shea et al. developed a clustering algorithm, called affinity propagation. Neighboring vehicles exchange their IDs, current positions, current velocities, etc., and compute affinity function to select the cluster head. However, this approach may not be suitable for routing purposes as a large number of messages are exchanged exhausting the available bandwidth. In [24], [25], clustering is done for some specific applications, for instance to calculate the amount of traffic. Such approaches do not care about the stability of clusters and are not suitable for supporting routing protocols. To increase cluster stability, [11] and [12] both consider vehicle mobility during clustering, but they did not provide any routing strategies. In addition, clustering is also used for message authentication purpose and is conducted when needed. Most recently, Hadded et al. [8] develop a vehicle clustering approach based on a multi-objective genetic algorithm. To sum up, all the aforementioned works mainly focus on measuring the stability of generated clusters, but did not provide any routing algorithm on how to use the generated clusters to conduct efficient message routing. Unlike these works, our approach provides a complete routing algorithm that consists of efficient clustering, long-term maintenance and efficient routing.

There have been some work which include both clustering and routing algorithms. However, some of these approaches rely on infrastructure support which may not be available soon in the near future due to the deployment cost. For example, in [26], Alawi et al. propose to find the route from a vehicle to the closest infrastructure using the signal strength as a guiding criteria. Similarly in [27], message delivery is conducted with the aid of the infrastructure. Unlike these works, we aim to design an approach which utilizes V2V communications only. Related works using pure V2V communication are summarized as follows. In [28], Little and Agrawal proposed to utilize a cluster header and a trailer at the front and the rear end of each cluster for information routing. However, a detailed election protocol is not presented. In [29], Goonewardene et al. designed a vehicle precedence algorithm to adaptively identify the nearby 1-hop neighbors and select optimal cluster heads based on vehicle locations and velocities. The main limitation of this approach is that the proposed algorithm requires each vehicle to keep sending out update information to neighbors which can introduce lots of communication overhead. In [30], Luo et al. form clusters based on geographically divided grids, but they did not consider velocity and direction which are important for accommodating the dynamic nature of VANETs. In [31], Ohta et al. use positions and moving direction of vehicles for clustering. Unlike these studies, we consider the rich mobility information during the clustering. Also, the cluster heads in [31] need to continuously broadcast MEP (cluster MEmber Packet), and it is only at this point that it can discover neighboring clusters for further routing. Another recent work is by Song et al. [17], who consider moving directions for cluster head selection. In summary, none of the existing clustering approaches on pure V2V environments considers the use of moving object techniques to reduce communication overhead, improve efficiency and effectiveness, as we present in this paper.

However, these clustering algorithms have at least one of the following limitations: (i) Clusters are formed based on partitioning of road networks instead of object mobility, which reduces the lifetime of clusters; (ii) Clustering process requires each vehicle to periodically broadcast messages or has complicated voting mechanism, which can incur high communication overhead; (iii) Clustering needs assistance of road-side units which may not be available in many environments; (iv) Clustering is focused on small-scale scenarios (e.g., hundreds of vehicles). Our proposed research will overcome these limitations. A preliminary version of our work appears in [32], where we presented the basic idea of the moving-zone based architecture. In this paper, we make the following new contributions. First, we develop major extensions to the previously proposed architecture by providing detailed algorithms for zone construction and zone maintenance. Second, we compare our approach with two representative approaches, one is clustering-based and the other is non-clustering-based. Third, we conduct an extensive experimental study on a large number of vehicles.

2.2 Representative Approaches for Comparison

In order to thoroughly evaluate our approach, we compare our approach with both clustering-based approaches and non-clustering based approaches. As the representative approach of clustering-based routing protocol, we select CBDRP [17] since its schema is most similar to ours. As the representative approach of non-clustering-based routing protocols, we select BRAVE [18] because it has shown to outperform many existing protocols including GSR [33], SAR [34], A-STAR [35], GPCR [36], GeOpps [37].

The CBDRP (Clustering-Based Directional Routing Protocol) first divides each road into equal-length segments. Vehicles in the same road segment and moving at the same direction are grouped in one cluster, and the vehicle closest to the center of the cluster is the cluster head. To route a message, the source sends the message to its cluster head, and the cluster head establishes the routing path first and then forwards the message along the path. There are two major limitations of this approach. First, the clusters are formed based on fixed partitioning of roads, without considering the similarity of movement among vehicles. As a result, the members in each cluster are updated very frequently, and this incurs heavy communication overhead. Second, the routing protocol requires the establishment of the path beforehand. The path may need to be maintained when the actual message is forwarded due to the dynamic nature of vehicles. The broken path problem is more severe when the distance between the sender and the receiver is far from each other. Such routing protocol not only introduces extra communication cost but also delays the transmission of messages.

The BRAVE (Beacon-less Routing Algorithm for Vehicular Environments) approach adopts an optimistic routing approach to reduce the message overhead in traditional broadcasting approaches. In BRAVE, a forwarding vehicle which has a message to send out will broadcast the message to its 1-hop neighbors. Every neighbor receiving the message will send back a response message. After the forwarding vehicle receives response message, it will broadcast a select message to indicate which neighbor has been selected to forward the message. There have been some variants of BRAVE, such as BIIR [38] which achieves slightly better performance than BRAVE by reducing the message overhead to 2/3. Compared with BRAVE and its variants, our work does not rely on broadcasting but more target-oriented communication. The message overhead in our approach is an order of magnitude less than BRAVE.

3 Moving Zone Based Vehicle Management Architecture

The MOving-ZOne-based (MoZo) architecture consists of multiple moving zones that are formed by vehicles with similar movement patterns. A captain vehicle is elected for each zone and is responsible for managing information about other member vehicles as well as the message dissemination. In the following sections, we first introduce how to model vehicle movement, and then present the detailed algorithms for zone construction.

3.1 Vehicle Movement Modeling

We assume that each vehicle is equipped with an on-board unit (OBU) for networking and computing messages, a global positioning system (GPS), and a digital map. Vehicles communicate with one another using data link technology (e.g., ASTM E2213-03[39]), within a range of 10 s minimum travel time (the minimum range is 110 meters and maximum is 300 meters) [40]. Further, we assume each vehicle has a unique identity which can be either a pseudonym or a real identity. Existing security and privacy protection techniques can be integrated with our approach while a detailed discussion of this possibility is beyond the scope of this paper.

We represent the road network as a graph whereby edges represent roads and vertexes represent intersections. The two ends of the roads are designated as the starting and ending points respectively. We model vehicle's movement as a linear function of time. Specifically, let $r(st, ed)$ be a road segment, where st is the starting point of the road and ed is the end point of the road. Given a vehicle on road r, let l_u be the vehicle's distance to st at time t_u, and let v be vehicle's speed at t_u. Let δ denote the vehicle's moving direction, which has value 1 if the vehicle moves toward ed, otherwise -1 if the vehicle moves toward st. Let t'_u denote the next possible update timestamp when the vehicle changes its moving speed or direction. Then, the vehicle's position at timestamp t ($t_u \leq t \leq t'_u$) is computed as follows: $l(t) = l_u + \delta \cdot v \cdot (t - t_u)$.

This model will be adopted by the captain vehicle to estimate its member vehicles relative positions on a road. Vehicles need to send the update message to the captain vehicle if they change their moving directions or speed dramatically. The movement of vehicles is modeled as a straight line between two consecutive update messages, which is analogous to the widely adopted idea of using line segments to approximate curvy roads. Since moving functions usually change much less frequently than locations, the adoption of such modeling will reduce the need for the member vehicles to send location updates to the captain vehicles constantly. The moving function will also play an important role in determining the members of a moving zone as discussed in the subsequent sections.

3.2 Moving Zone Construction

Moving zone construction starts from a vehicle logging onto the VANET. The vehicle will execute the joining protocol to find a nearby moving zone or form its own zone. The zone forming criteria is configured based on the similarity of vehicle movement. The captain vehicle of each zone maintains a moving object index that manages up-to-date information about all its member vehicles. In what follows, we discuss the operations that need to be conducted at member vehicle side and the captain vehicle side respectively.

3.2.1 Member Vehicle Side

When a vehicle V_s enters the VANET, it sends a hello message to its one hop neighbors. The hello message consists of its unique identifier V_s, current road ID (ID_r) and moving direction (δ). The vehicle waits for τ amount of time to accumulate the responses to its hello message. τ is the estimated total time for a single message to be received, processed by the receiver, and transmitted and propagated back to the sender within the communication range of the sender vehicle.

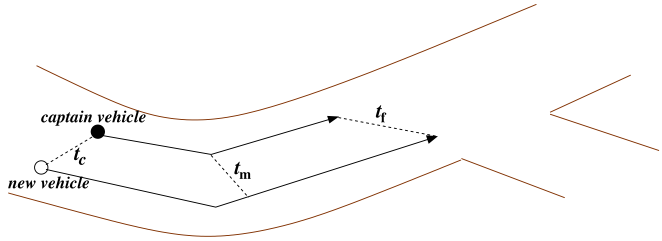


Fig. 2. Anticipated trajectories.

If a captain vehicle moving in the same direction (δ) receives the hello message, it sends a response to the corresponding vehicle. The response includes its unique identifier V_{cap}, current location l, speed v, and the next intersection Int that it is heading to. We will discuss how to select the captain vehicles in Section 5.

When τ expires, the vehicle calculates a similarity score for each response received from the neighboring captain vehicles. The goal is to assign a higher score to the captain vehicle which will stay closer to the vehicle for a longer time period so that the vehicle can find a zone in which it can stay longer. To accomplish this, we define the similarity score based on the average distance between the two vehicles' anticipated trajectories within a certain time period. The computation includes the following three steps.

The first step is to determine how far into the future the anticipated trajectories should be considered, i.e., the time period for computing the average distance. Fig. 2 shows an example. The two arrow lines indicate the anticipated trajectories of the captain vehicle and the new vehicle respectively, and we can see that they are up to the intersection of the roads. We consider the following timestamps after which the vehicle is likely to update its moving parameters.

- Let t_1 (or t_2) be the timestamp that the sender vehicle (or the captain vehicle) reaches the next intersection. To maintain a high prediction accuracy, we do not predict beyond the intersection since trajectories after this point are hard to be predicted based on current movement function.
- Let t_3 be the timestamp when the distance between the two vehicles exceeds the communication range, because these two vehicles will not be in the same zone after t_3.
- Let t_4 be the possible timestamp that the new vehicle may send an update to the captain vehicle. t_4 is computed as $t_4 = t_c + \tau_u$, where t_c is the current timestamp and τ_u is the maximum interval between two consecutive updates of a member vehicle that is recorded by the captain vehicle.

The first three timestamps can be easily computed using the vehicles' current moving speed and direction. Finally, the earliest timestamp among the four: $t_f = \min(t_1, t_2, t_3, t_4)$, will be selected. The time period to be considered is thus $\Delta_t = t_f - t_c$.

Protocol: Vehicle Joining Event (V)
Vehicle V**:**
1. Send HELLO messages to neighbors
Captain Vehicle V_c**:**
2. Receive the HELLO message from V
3. **If** V_c moves on the same road at the same direction as V
4. Send a response message to V
Vehicle V**:**
5. **While** wait time is less than τ
6. Receive response messages
7. **If** no response message is received
8. ZoneConstruction(V_c) // Form the moving zone itself
9. **Else**
10. **For** each responding captain vehicle V_c
11. Compute the similarity score $Sim(V, V_c)$
12. Send the join request to V_c with the highest score
Captain Vehicle V_c**:**
13. Receive the join request from V
14. Send a confirmation message to V
15. ZoneConstruction(V)

Fig. 3. Protocol for vehicle joining event.

The second step is to compute the positions of the two vehicles at timestamps $t_c + \frac{1}{2}\Delta_t$, and t_f, respectively. These positions are relative positions on the corresponding road, i.e., the distance from the road starting point. Together with their current locations, these three sample positions are used to represent the vehicles' anticipated trajectories. The reason to choose sample points instead of using integral of moving functions is to reduce the computational complexity and satisfy the strict temporal constraints of VANETs.

Finally, the similarity score of the two vehicles is defined as shown in Definition 1.

Definition 1. *Given two vehicles V_1 and V_2, let l_{c1} (l_{c2}), l_{m1} (l_{m2}), and l_{f1} (l_{f2}) denote the positions of the two vehicles at t_c, the middle timestamp and t_f, respectively. Let w_c, w_m and w_f be weight values, where $w_c > w_m > w_f$. The similarity score of V_1's and V_2's projected moving trajectories is computed as follows:*

$$S_{V_1V_2} \triangleq \frac{\Delta_t}{w_c|l_{c1} - l_{c2}| + w_m|l_{m1} - l_{m2}| + w_f|l_{f1} - l_{f2}|}. \quad (1)$$

The above equation integrates the effects of two factors. First, the numerator in the formula is the time interval during which the two vehicles' trajectories are considered. A higher value will be returned for vehicles that stay together for a longer time period Δ_t. Second, the denominator in the formula is the distance between the two vehicles at the three sample timestamps. A higher similarity value will be returned for vehicles which stay closer to one another, i.e., have shorter distance. The distance between vehicles is computed as a weighted distance. The use of decreasing weights allows modeling of predicted positions that become less accurate as time passes.

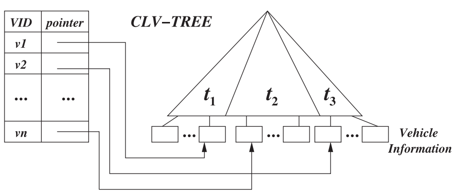


Fig. 4. The structure of the CLV-tree.

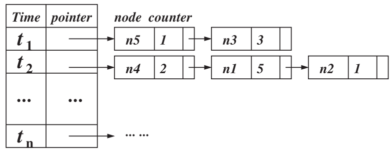


Fig. 5. The structure of LE queue.

After computing the similarity scores with respect to the neighboring captain vehicles, the vehicle selects the captain vehicle with the highest score and sends a join request to the captain vehicle. The join request consists of the vehicle's ID, current position and moving speed. The respective captain vehicle will send a confirmation message to this vehicle to complete the joining process.

In case that there is no moving zone nearby, the vehicle will form a new moving zone of its own and becomes the initial captain vehicle. As time passes, this new moving zone may have more members, and the initial captain vehicle may conduct a captain vehicle re-assignment as discussed in Section 5.

Fig. 3 summarizes the joining protocol. The "ZoneConstruction()" function in line 15 is discussed in the next section.

3.2.2 Captain Vehicle Side

Each captain vehicle needs to keep up to date information about its member vehicles in order to carry out message dissemination and zone maintenance. To achieve this, we propose two simple yet effective data structures to be maintained by each captain vehicle. One is the Combined Location and Velocity Tree (CLV-tree). The other is the Leaving Event queue.

The CLV-tree is a hybrid moving object index consisting of a B^+-tree and a hash table. Fig. 4 illustrates an example CLV-tree. Each entry in the leaf node of the B^+-tree stores a member vehicle's identity, the latest update timestamp t_u, location l_u and speed v_u at t_u, its index key, and estimated leaving timestamp t_{ex}. Each row in the hash table has two entries: one stores the vehicle's identity, and the other stores the pointer linking to the leaf node that contains the vehicle. Both base structures are very efficient in terms of insertion and deletion, which will not impose much workload to the captain vehicle.

The Leaving Event (LE) queue stores the estimated timestamps when member vehicles may be out of the communication range of the captain vehicle, in an ascending order. As shown in Fig. 5, each entry in the LE queue contains a leaving timestamp and a pointer to a list of nodes that contain the vehicles leaving at that timestamp. A counter is associated with the node to record the number of leaving vehicles.This LE queue is updated whenever a vehicle joins the zone or sends an update to the captain vehicle. Upon receiving the latest movement information of a vehicle, the captain vehicle computes the leaving timestamp t_{ex}. Note that t_{ex} may be infinity when the vehicle traveling at the same speed and direction as the captain vehicle. In that case, no entry in the LE queue is needed for that vehicle. The LE queue will be used during the zone maintenance phase which will be discussed later in Section 5. The overall protocol at the captain vehicle side is outlined in Fig. 6.

Algorithm: ZoneConstruction (V)
Input: V is a vehicle that sends the join request
Begin Algorithm
1. Compute the index key for V
2. Insert V to the CLV-tree
3. Node$\leftarrow$ leaf node in CLV that contains V
4. Compute V's leaving time t_{ex}
5. **If** t_{ex} is a finite number then
6.   **If** t_{ex} exists in LE queue
7. $L_{ex} \leftarrow$ list of nodes pointed by t_{ex}'s entry in LE
8.   **If** Node exists in L_{ex}
9.   Increase the counter of Node in L_{ex}
10.   **else** insert Node to L_{ex}
11.   **else** create a new entry for t_{ex} and Node in LE

End Algorithm

Fig. 6. Zone construction at captain vehicle side.

Algorithm: SearchCLV-tree(x,y,δ)
Input: (x,y) is the query location,
and δ is the moving direction
Output: propagation vehicle V_p
1. $L_v \leftarrow \emptyset$
1. For each time partition t_i in the CLV-tree
2.   qKey$\leftarrow$Encode(x,y,δ,t_i)
3.   Node$\leftarrow$ CLV.root
4.   While (Node is not leaf node)
5.   Find the entry e in Node that contains qKey
6.   Node$\leftarrow$ e
7.   Add vehicles in Node to L_v
8. For all vehicles in L_v
9. $V_p \leftarrow$ the one nearest to (x,y)

End Algorithm.

Fig. 8. CLV-tree query algorithm.

4 The MoZo Routing Protocol

We now discuss how to take advantage of the MoZo architecture to route a message to a specified destination for the example applications discussed in the introduction. In particular, suppose that a vehicle has a piece of information (I) that it would like to share with vehicles around location $l(x, y)$. The overall routing protocol is summarized in Fig. 9. It specifically consists of the following steps.

Step 1: The sender vehicle sends a message in form of $\langle ID_s, I, l(x, y)\rangle$ to its captain vehicle, where ID_s is the sender vehicle's unique identity, I is the message and $l(x, y)$ is the location of the message destination.

Step 2: Upon receiving the message, the captain vehicle first checks if the message destination is within its moving zone. If not, it looks for the member vehicle in its moving zone which is closest to the message destination, and forwards the message to the selected member vehicle.

The algorithm for finding a good candidate vehicle for the message propagation (or propagation vehicle) is the following. The captain vehicle first computes the shortest route to the destination $l(x, y)$ using the Dijkstra algorithm, and then computes the intersection point l_e of the shortest route and its communication range as shown in Fig. 7. Member vehicles which are around this location l_e and move towards the message destination are considered good candidate vehicles for propagating the message. To find such vehicles in the moving zone, the captain vehicle will execute a query algorithm.

Specifically, the captain vehicle generates a query key by encoding the expected location l_e and the desired moving direction. The obtained query key will be treated as the key belonging to a virtual vehicle. A virtual insertion algorithm will be conducted to locate the leaf node in the index for this virtual vehicle. Once the leaf node is found, this virtual insertion algorithm will stop, which is unlike the regular insertion algorithm that actually inserts a data to the index. The vehicles in the resulting leaf node contain similar keys to the virtual vehicle. In other words, they are likely to be near the location l_e. For further verification, the location of each vehicle at the current timestamp will be computed based on their latest moving function. The vehicle which is closest to l_e is chosen as the propagation vehicle (denoted as V_p). Fig. 8 outlines the search algorithm. If not any candidate vehicle can be reached, the captain vehicles will wait for μ seconds and then try to find the candidate vehicle again. Up to three attempts will be made to deliver one message.

If the message is located in the current moving zone, the captain vehicle will deliver the message to the member vehicle near the message destination. In particular, the captain vehicle will encode the message destination to the query key and employ the aforementioned query algorithm to locate the receiver vehicles.

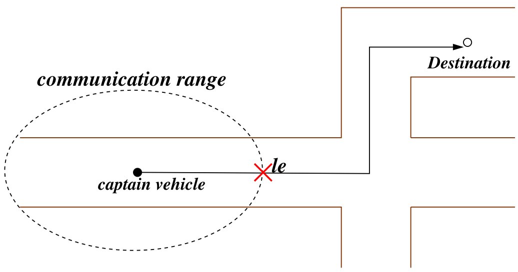


Fig. 7. Computation of message delivery route.

Protocol: Message Routing
Vehicle V_{sender}**:**
1. Send $M = \langle ID_s, I, l(x, y)\rangle$ to its captain vehicle V_c

Captain Vehicle V_c**:**
2. Receive message M
3. If $l(x, y)$ inside the zone
4. $V_{receiver}$ $\leftarrow$SearchCLV-tree(x,y,δ)
5.   Send message M to $V_{receiver}$
6. Else
7.   compute intersection point $l_e(x', y')$
8. V_p $\leftarrow$SearchCLV-tree(x',y',δ)
9.   Send message M to V_p

Vehicle V_p**:**
10. Receiver message M
11. Sort the captain vehicle list
12. For each V_c' in the captain vehicle list
13.   If V_c' is available and move towards l(x,y)
14.   Send message M to V_c' and done
15. Ping neighbors
16. For all responding vehicles
17.   Find the V_p' closest to and move towards l(x,y)
18. Send message M to V_p'

Vehicle V_p'**:**
19. Receive message M
20. If V_p' is a captain vehicle
21.   Conduct operations from step 2
22. Else
23.   Send message M to its captain vehicle V_c'
24. V_c' conduct operations from step 2

End Protocol.

Fig. 9. Message routing protocol.

Step 3: If the message is received by the selected propagation vehicle (V_p), V_p will be responsible for sending the message to vehicles in nearby moving zones. This operation will utilize the previously stored information about nearby captain vehicles. In particular, each vehicle keeps a list of captain vehicles which responded to the hello message sent when the vehicle requested to join a moving zone. Vehicle V_p checks its list to find the captain vehicles which have an update timestamp not earlier than the current time minus 2τ (τ is the wait time introduced in Section 3.2.1), and move toward the message destination. V_p sorts these vehicles in an ascending order of their distance to the message destination. Then V_p pings these vehicles. Once V_p receives responses, V_p selects the captain vehicle that is on the top of the sorted list and sends out the message. If no response is received within τ in Equation (1), which is possible since the captain vehicles in the list may have already changed their moving functions, V_p will ping its one hop neighbors. Based on the response from neighbors, V_p will select the one closest to the message destination as the next propagation vehicle.

Step 4: There are two cases in this step. In case a captain vehicle from a different moving zone receives the message from V_p, this captain vehicle starts the tasks of Step 2. In case a regular vehicle from a different moving zone receives the message, the vehicle will forward the message to its captain vehicle and the captain vehicle will start the tasks as per Step 2 as well.

5 Moving Zone Maintenance

Zone maintenance is a continuous process that monitors the quality of the existing moving zones and conducts zone reformation accordingly to ensure the success of message routing. To maximize the information usage and reduce the communication overhead, the maintenance process leverages information collected during message routing. It includes four major tasks: (1) handling vehicle updates; (2) selecting a replacement captain vehicle; (3) zone splitting; and (4) zone merging.

5.1 Handling Vehicle Updates

We start from the first task. A member vehicle transmits new movement information to its captain vehicle only when its moving function (described by speed and direction) has changed dramatically. In-between two consecutive updates, the captain vehicle estimates the vehicle's location using its latest speed and direction. This update strategy can dramatically reduce the amount of updates compared to existing works which requires each vehicle to update its location every timestamp.

Upon receiving an update from a member vehicle, the captain vehicle sends back a ping message to confirm that it receives the update information. Then, the captain vehicle updates the CLV-tree and the LE queue. It follows the hash table of the CLV-tree to locate the leaf node that stores the old information of the vehicle, and then insert the new information to the CLV-tree as discussed in Section 3.2.2. The captain vehicle also computes the new leaving time for the vehicle and updates the LE queue accordingly.

5.2 Captain Vehicle Reassignment

At certain time point, there may be a need to find a new captain vehicle to replace the current one. For example, the current captain vehicle changes its moving function significantly and will soon be out of reach for most of its member vehicles. Or, the captain vehicle notices that there is a member vehicle which is more suitable to be the captain than itself. The second case can be detected when the captain vehicle notices that its information is stored at the left or rightmost leaf node of the CLV-tree, which implies that its movement pattern has become less similar to its member vehicles.

Once the current captain vehicle can no longer serve this role, it conducts the *captain vehicle re-assignment process*. A good captain vehicle is expected to stay relatively in the center of the moving zone and moves at an average speed with most of other member vehicles. We propose the following heuristic approach to quickly locate such a candidate captain vehicle. In particular, we take advantage of the CLV-tree instead of examining all member vehicles' movement functions. Recall that the CLV-tree has three indexing timestamps and organizes vehicles based on their relative positions on the road. The vehicle stored in the middle of the largest time partition of the CLV-tree, will be selected as the new captain vehicle.

The reason of such selection is two fold. First, the time partition that contains the largest number of member vehicles implies that these vehicles have been updated not long ago and the information will be up-to-date for a while. Second, the vehicle stored in the middle of this partition is the one which has the positions in the middle of this group of vehicles. After the candidate captain vehicle is identified, the current captain vehicle will contact the candidate vehicle and pass information about member vehicles to it. The new candidate vehicle will broadcast a message to inform current members about its new status.

5.3 Zone Splitting

Vehicles in the same zone have similar but still different movement functions. The -possibly small- difference among vehicles moving patterns is accumulated and may eventually enlarge the distance between vehicles. Consequently, after certain time period, some vehicles in the same zone may be out of communication range of one another. Hence, the zone should be split periodically. The specific algorithm is the following.

The captain vehicle monitors the leaving timestamps of member vehicles using the LE queue. When a leaving timestamp t_{ex} is approaching (e.g., current time is $t_{ex} - \tau$), the captain vehicle sums up the counters in the list linked to this timestamp, which is the total number of vehicles that will leave at t_{ex}. If this number is smaller than $\varphi \cdot N_v$ (N_v is the number of vehicles in the zone, and φ is a percentage parameter), the captain vehicle will locate these vehicles in the CLV-tree and send a message to inform each of them that they will soon be out of current moving zone. Here, φ is a tunable threshold. Upon receiving the notification from the captain vehicle, the leaving vehicles will prepare to execute the vehicle joining protocol (Fig. 6) to find new moving zones at the leaving time.

Otherwise, if a large number of vehicles (i.e., more than N_l) is about to leave, the captain vehicle will split the zone into two new zones. The zone splitting leverages the

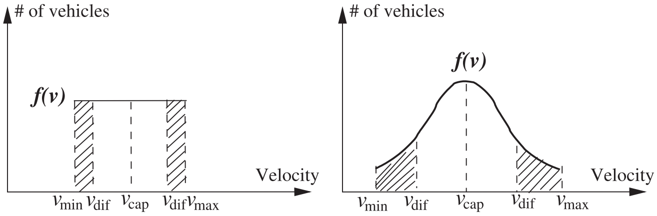


Fig. 10. Velocity distribution.

CLV-tree. Considering the properties of the CLV-tree which groups vehicles with similar movement functions in nearby nodes, the captain vehicle generates the first zone containing vehicles stored in the first half nodes of each time partition in the CLV-tree, while the second zone containing the remaining vehicles. Then, the captain vehicle selects the new captain vehicle for each newly constructed moving zone using a procedure similar to the captain vehicle reassignment process. The only difference is that the new captain vehicle for each zone is selected from the half of the CLV-tree instead of the original CLV-tree. After that, the current captain vehicle informs the two new captain vehicles their zone members. The new captain vehicles take over the management task from here and notify their member vehicles. Zone splitting has the following benefits. A large number of vehicles leaving at the same time are reassigned efficiently and simultaneously. This saves time in sending out individual notification messages to each leaving vehicle as well as the time to execute vehicle joining protocol separately by each leaving vehicle.

The following theorem gives a formula for computing the estimated zone splitting time.

Theorem 1. *Given a moving zone, let N_v be the total number of vehicles in the zone, $f(v)$ denote the distribution function of vehicle's velocity, and let v_{min} and v_{max} be the minimum and maximum vehicle speeds of the vehicles in the zone respectively, and let v_{cap} denote the captain vehicle's speed. Let R denote the one-hop communication radius. The time interval (t_s) from the zone being constructed till zone splitting is computed as*

$$t_s = \frac{\frac{1}{2}R}{v_{dif}}, \tag{2}$$

where, v_{dif} is obtained by solving the following equation:

$$\int_{v_{min}}^{v_{cap}-v_{dif}} f(v)dv + \int_{v_{cap}+v_{dif}}^{v_{max}} f(v)dv = \varphi N_v. \tag{3}$$

Proof. We prove this theorem by assuming two common distributions of $f(v)$. Specifically, we consider vehicles' speed in a moving zone follows either a uniform distribution or a normal distribution as illustrated in Fig. 10. In either distribution, according to the captain vehicle selection criteria, the captain vehicle's speed is expected to be the mean speed so that the captain vehicle moves along with most of its member vehicles for a long time.

Recall that zone splitting occurs when there are more than φN_v of vehicles leaving, i.e., out of the communication range of the captain vehicle. Since vehicles which have speed more different from the captain vehicle will leave the zone earlier, these φN_v vehicles are likely to have speed close to either v_{min} or v_{max} as indicated by the shaded area in the figure. At the splitting moment, the number of vehicles in the shaded area would be φN_v, and hence we obtain Equation (3).

In Equation (3), the only unknown variable is v_{dif}. Solving the equation for v_{dif}, we can obtain the value of v_{dif}. Among the φN_v leaving vehicles, the one with speed closest to the captain vehicle will be the last to leave the zone. As shown in the figure, the possible speeds of the latest leaving vehicles are $v_{cap} - v_{dif}$ and $v_{cap} + v_{dif}$. Therefore, the leaving time of the vehicles with speed $v_{cap} - v_{dif}$ or $v_{cap} + v_{dif}$ is the splitting time. Given the communication range of R, we consider that in average case the latest leaving vehicles are $\frac{1}{2}R$ away from the captain vehicle. Then, we compute the splitting time as follows:

$$t_s = \frac{\frac{1}{2}R}{v_{cap} - (v_{cap} - v_{dif})} = \frac{\frac{1}{2}R}{v_{dif}}$$

$$t_s = \frac{\frac{1}{2}R}{(v_{cap} + v_{dif}) - v_{cap}} = \frac{\frac{1}{2}R}{v_{dif}}.$$

□

Theorem 2 indicates that the less the speed difference between the captain vehicle and the member vehicles, the later the zone splitting will occur. This is also in line with our algorithm that aims to find vehicles with similar moving trend.

5.4 Zone Merging

As time passes, some moving zones may overlap with one another. Heavily overlapped moving zones introduce unnecessary management and communication overhead. If vehicles in overlapping zones are merged into one and managed by only one captain vehicle, only one captain vehicle needs to respond to joining requests from other vehicles instead of multiple captain vehicles. Therefore, we propose the following zone merging protocol.

The zone merging protocol is typically initialized by a relay vehicle which detects the need of merging. We first discuss the timing and the necessary conditions for the merging. When two moving zones get closer to one another, the distance between their captain vehicles is shorter. When the distance is less than the communication range, half of the two moving zones are nearly overlapping and have potential to be merged. This situation can be detected by the vehicles at the borders of zones utilizing information collected during routing, without any extra message communication. In particular, recall that during the message routing, the relay vehicles receive response from their neighboring captain vehicles. According to the response, the relay vehicles know the number of nearby moving zones as well as the distance to their captain vehicles. When a relay vehicle detects that there are more than two captain vehicles within half of its communication range and these captain vehicles have been in this range for at least two message transmissions, the relay vehicle will execute the merging protocol. Note that here we consider two conditions for merging. One is the distance between the two moving zones, and the other is the duration of the moving zones

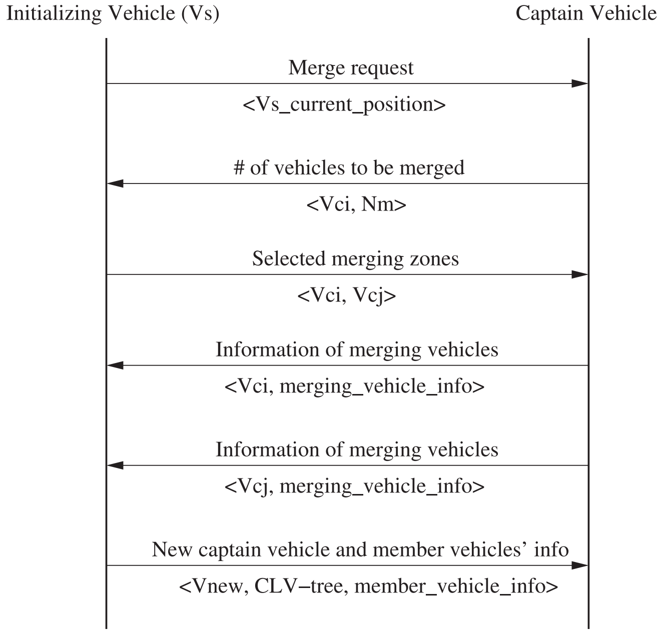


Fig. 11. Zone merging protocol.

being in the close range. This is because, if a moving zone is just quickly passing by another one, the overlap of the two zones is temporary and would not affect the overall performance in the long term.

The merging protocol is sketched as follows.

- The initializing vehicle (denoted as V_s) sends a merging request including its current position to the captain vehicles that qualify the merging condition.
- Each captain vehicle (denoted as V_{c_i} which receives the merging request, computes the number of vehicles (denoted as N_m) in its zone that are within the communication range of V_s. If many vehicles in V_{c_i}'s zone are also in the communication range of V_s, that means the two zones are overlapping and they would be better off by being merged. To quantify the amount of vehicles, we use the same threshold N_l. If N_m is greater than N_l, V_{c_i} will send a message containing N_l back to V_s.
- Upon receiving the response from the captain vehicles, V_s selects two zones which contain the maximum numbers of vehicles to be merged.
- V_s broadcasts the captain vehicles of the selected zones. When the selected captain vehicles receive this message, they send the merging vehicle information to V_s and inform the remaining vehicles to start finding new zones. V_s constructs a CLV-tree to store the received vehicle information and selects the median vehicle in the tree to be the new captain vehicle.
- V_s passes all vehicle information to the new captain vehicle, and the new captain vehicle informs its members about the change.

Fig. 11 illustrates the messages exchanged between the initializing vehicle and captain vehicles. Note that if during the execution of one merging protocol, the captain vehicles receive merging requests initialized by other relay vehicles, these requests will be ignored to avoid duplicate merging.

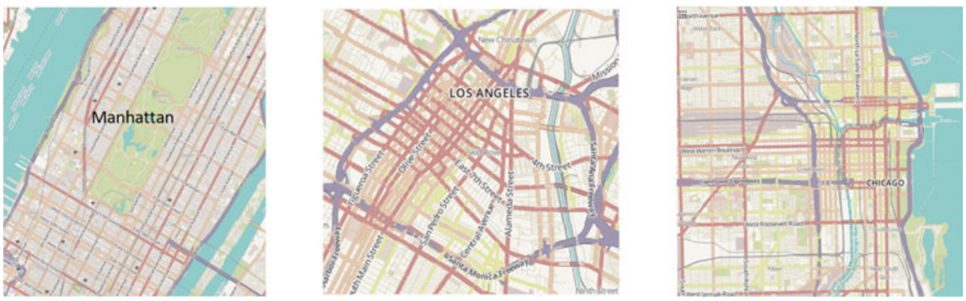
6 Experimental Study

In this section, We first introduce the experimental settings, and then report the experimental results.

6.1 Experimental Settings

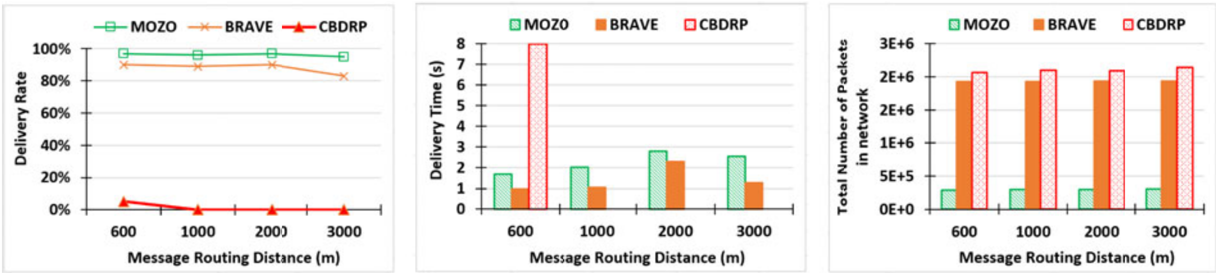
We compare our approach with two approaches, i.e., CBDRP [17] and Brave [18] (as described in Section 2.2), representing the clustering-based routing protocols and non-clustering-based routing protocols, respectively.

The experiments were conducted using the Network Simulator NS-2 (version 2.35) and vehicular mobility simulator SUMO (version 0.23.0) under Ubuntu 15.04 (64 bit). The SUMO simulates the vehicles' continuous movements along the roads of three real maps as shown in Fig. 12: Manhattan (4.5 × 5.5 km), Los Angeles (5 × 4.5 km) and Chicago (6 × 7 km). Due to the limitation of the simulation platform, we simulate vehicles up to 1,400 on each map. The vehicles' starting positions are randomly distributed on the road map. Similar to [18], vehicles move at a maximum speed of 30 mile/hour inside the city and 60 mile/hour on the highway. We use NS-2 and SUMO to simulate scenarios with and without traffic light controls. In both scenarios, vehicles will slow down when approaching the intersections and wait in the queue to make their turns. The vehicle behavior in the simulator is very close to that in the real life. NS-2 implements 802.11 physical and MAC models for vehicle-to-vehicle communication and the maximum transmission range is set to 500 m. Unless noted otherwise, we use the Manhattan map and set the total number of vehicles in the VANET to 800. By default, 100 512-byte messages are generated for each run of the experiments and the default distance between a message source and destination is 2,000 m. The simulation was run for 50 seconds to insert all vehicles



(a) Manhattan (b) Los Angeles (c) Chicago

Fig. 12. Maps used in the experiments.



(a) Delivery rate (b) Delivery time (c) Total Number of Packets

Fig. 13. Effect of message delivery distance.

and let vehicles move around on the network for a bit. After 50 s, vehicles issue message requests and the total simulation time is 200 seconds.

In the simulation, we vary the following parameters: (i) the distance between the message source and destination, (ii) the number of total vehicles in the VANET at the same time; (iii) the total number of messages to be delivered at the same time; (iv) the map topology. The range of these parameters will be elaborated along with the performance analysis in the following section.

The performance is measured using the following criteria: (i) message delivery time; (ii) successful delivery rate; (iii) communication overhead in terms of the total amount of messages received by all the vehicles during the whole simulation time which includes the message to be delivered, and maintenance messages for updating vehicles locations with the captain vehicles and zone merging and splitting. The reported result is the average of 10 independent runs for the same configuration.

6.2 Experimental Results

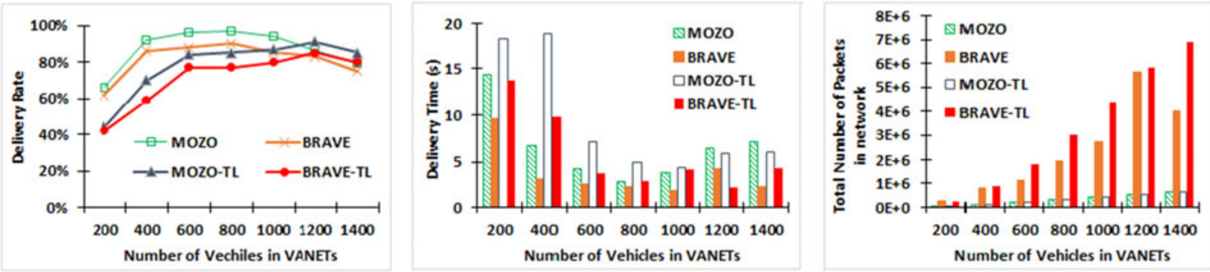
In all approaches, the maximum attempts to deliver the message is set to 3, and each message will be kept by a vehicle for maximum 15 s in the message queue for delivery. The beacon interval in other two approaches is set to 2 s as reported in [18]. In MoZo, vehicles need to send updates to the captain vehicle when they deviate from their original moving functions more than 5 m/s or the time from the last update is more than 4 s (Note that this maximum update interval of 4 s can be increased to further reduce the message overhead in our approach). The zone splitting threshold is set to 30 percent, and the weight value assignment for trajectory prediction is: $w_c = 0.5$, $w_m = 0.3$, $w_f = 0.2$, which have been identified to be the best parameters in most cases after multiple rounds of experiments under different scenarios.

6.2.1 *Effect of Message Delivery Distance*

In the first set of experiments, we randomly select message senders, and then select message destinations which are d meters away. We vary d from 600 meters to 3,000 meters. For each value of d, 100 messages are generated. As shown in Fig. 13a, MoZo achieves the highest delivery rate among all, BRAVE has slightly lower delivery rate while CBDRP is the lowest. The CBDRP's delivery rate drops quickly to zero when the distance is increased to 1,000 m. The reason is the following. The clusters of vehicles established by MoZo are more stable than CBDRP since MoZo models vehicle movement into a near future while CBDRP only considers vehicles' current moving directions. Moreover, CBDRP needs to explore the route first before sending the actual message. Due to the frequent changes of the clusters in CBDRP, the established route needs to be frequently maintained and may not be valid when the actual message is sent. When the route distance becomes longer, the probability of the established route being invalid increases, which severely affects the message delivery rate. Unlike CBDRP, BRAVE does not explore the whole route before sending the message. Instead, BRAVE only detects next available message forwarder and hence it adapts to the dynamic nature of the VANET network topology much better than CBDRP. However, compared with MoZo that relies on clusters of vehicles for delivery, BRAVE may not always be able to find the forwarder since it is possible that an individual vehicle cannot find any forwarding vehicle on the direction to the message destination. In MoZo, the captain vehicle of the message sender has contacts with more vehicles and hence there is a higher chance of finding qualifying forwarding vehicles, leading to higher delivery rate. This also demonstrates the advantages of clustering-based routing protocols. Another observed trend for all approaches is that the message delivery rate decreases when the message routing distance increases. This is because the longer the distance, the higher the probability that the message being dropped in the middle of the route due to various reasons such as sparse distribution of vehicles on a certain road.

We also measure the message delivery time which is measured from the sender vehicle sending out the message till the recipient vehicle receives the message. Fig. 13b shows the results. The time taken by CBDRP is longest and not reported for distance greater than 600 meters because CBDRP has no message being delivered for long distance. As for MoZo and BRAVE, they perform similarly while MoZo delivers messages slightly slower than BRAVE. This is because the cluster maintenance and forwarder selection algorithms in MoZo are more complex than that in BRAVE. We would like to mention that we have also conducted additional experiments by reducing the number of message attempt to 1 for MoZo. In that case MoZo achieves shorter delivery time but similar delivery rate as BRAVE.

Finally, we would like to emphasize that the major advantage of MoZo is the substantially smaller number of messages transmitted in the VANET to accomplish the same task of message delivery. Fig. 13c shows the total number of packets received by vehicles in the VANET in order to deliver 100 messages to the destinations at the



(a) Delivery rate (b) Delivery time (c) Total Number of Packets

Fig. 14. Effect of number of vehicles in VANETs with and without traffic light controls.

specific distance (varying from 600 to 3 km). We can see that the communication overhead in MoZo is about an order of magnitude less than the other two approaches. The CBDRP has the highest communication overhead since it needs to establish route first and then send the actual message. The BRAVE is more efficient than CBDRP as it adopts a beaconless strategy. BRAVE does not need to discover the entire route before sending the message, but it still needs the vehicle that receives the message to broadcast to neighboring vehicles to identify the next forwarding vehicle. Unlike BRAVE, MoZo does not use broadcasting to find candidate forwarding vehicles. Instead, using MoZo, vehicles that need to send a message only need to contact their captain vehicle. MoZo uses the captain vehicle to monitor member vehicles and select forwarding candidates by keeping their moving functions and a very small number of moving function updates. Such low communication overhead accomplished by MoZo will be important for scaling VANET applications in the real world.

6.2.2 Effect of the Number of Vehicles

In what follows, the default road distance between a pair of the message sender and receiver is set 2,000 meters (4 times of the communication range). Since the CBDRP has close to 0 delivery rate when the message delivery distance is more than 600 meters, we report the results of the MoZo and BRAVE in the remaining experiments.

Fig. 14 shows the performance when varying the total number of vehicles in the VANET from 200 to 1,400 under two scenarios: (i) without traffic light control and (ii) with traffic lights (denoted as “-TL” in the figure). The reason for setting the range to [200, 1,400] is the fact that this range is sufficient for revealing the underlying performance trends and where the optimal performance is observed. The range also covers the cases where the vehicle density increases from low to high. Specifically, when there are 200 vehicles in the whole network, the average vehicle density is about 0.58/100 meters; when there are 1,400 on the road, the average density is about 4.08/100 meters. There are a total of 12 traffic lights on the Manhattan map.

It is worth noting here that, in all the conducted experiments, the proposed MoZo scheme achieves better delivery ratio, similar delivery time, and much smaller communication overhead compared to BRAVE, due to the same reasons discussed in the previous experiments. Here, we discuss some other interesting trends observed for both approaches. Specifically, Fig. 14a shows that the delivery rate first increases with the number of vehicles and then decreases in the scenarios without traffic light control. This is because when there are very few vehicles on the roads, it may be hard to find the nearby forwarding vehicles and hence messages are dropped after the wait time. On the other hand, when there are many vehicles on the roads, there are two possible scenarios causing the low delivery rate. One possible scenario is that too many vehicles cause the traffic jam and non-uniform distribution of vehicles, resulting in fewer vehicles within communication range in the middle of the routing paths. The other scenario is that the vehicle distribution is uniform during certain period of time, but due to the increased amount of communication among the large number of vehicles, the communication channel is jammed and hence causes some messages being dropped. Therefore, the delivery rate reaches the optimal point when vehicles on the road are relatively uniformly distributed within the communication range. However, in the case with traffic light controls, we can see that the delivery ratio does not decrease when there are a large number of vehicles in the network. This could be attributed to the traffic light controls that help direct traffic better and hence reduce some traffic congestions.

Fig. 14b shows that when the number of vehicles in the VANET increases, the delivery time first decreases and then increases. Such behavior is related to the delivery rate. In the scenarios when the delivery rate is low, vehicles typically need to make multiple attempts of deliver the messages or wait longer to deliver the message, which increases the overall delivery time.

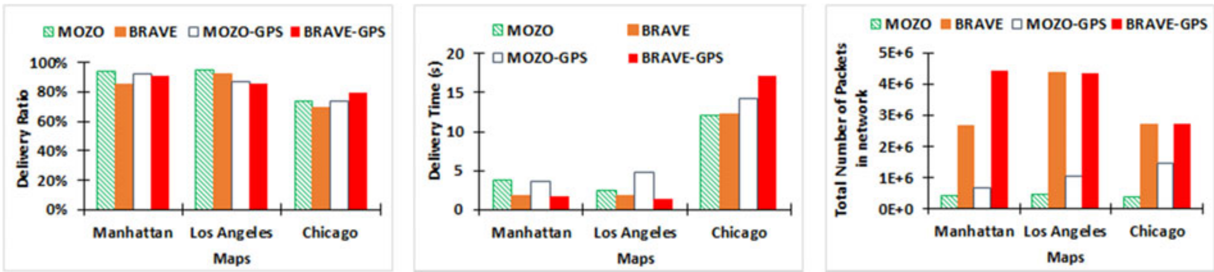
Fig. 14c shows that the communication overhead when one uses BRAVE increases much faster than our approach under both scenarios (with and without traffic light controls). This again proves the benefit of the proposed MoZo scheme that eliminates a large number of unnecessary message exchanges. In addition, we also see that the communication overhead increases with the increase of vehicles. The reason is straightforward. The more vehicles, the more communication among vehicles.

6.2.3 Effect of Number of Messages to Be Delivered

In this set of experiments, we evaluate the performance of the routing protocols by varying the number of messages to be delivered from 100 to 500 in the VANET with 800 vehicles. The distance between the sender and the destination is still 2,000 m. As shown in Figs. 15a and 15b, when the number of messages increases, the delivery rate decreases slightly. Correspondingly, the delivery time increases. The possible reason is that the more messages to

(a) Delivery rate (b) Delivery time (c) Total Number of Packets

Fig. 15. Effect of number of messages to be delivered.



(a) Delivery rate (b) Delivery time (c) Total Number of Packets

Fig. 16. Effect of map topology and GPS errors.

be delivered, the higher probability that the communication channel becomes jammed at certain point of the routing path and hence causes the lost of the message. We again observe that our proposed MoZo achieves better delivery rate even when more half of vehicles (500) in the VANET sent message requests simultaneously. As previously mentioned, this is because MoZo is capable of reaching more potential forwarding vehicles than BRAVE due to the use of clusters. Moreover, in Fig. 15c, we observe consistent small communication overhead in MoZo, which is less than 1/10 of that in BRAVE. This demonstrates the advantages of clustering-based strategy adopted by MoZo.

6.2.4 Effect of Road Topology and GPS Errors

In the last round of experiments, we study the effect of road topology as well as GPS errors on the performance of routing protocols using the three real maps shown in Fig. 12. The total number of vehicles in each network is 1,000 and the message delivery distance is 2,000 meters. To simulate the GPS errors, each vehicle's position is shifted from its true position to anywhere within 15 meters (the civilian GPS' accuracy range). The results with the consideration of GPS errors are denoted using "-GPS" in Fig. 16. Observe that the proposed MoZo scheme achieves a better delivery rate, similar delivery time, and much lower communication overhead compared to the BRAVE scheme in most cases. We also observe that the map topology does not affect the performance much when the size of the map is similar, i.e., Manhattan and Los Angeles. When the map is larger (i.e., Chicago), the vehicle density decreases and hence it decreases the delivery rate while increasing the delivery time. Moreover, after introducing the GPS errors, the impact on the overall performance is very small, while our proposed MoZo has been affected slightly more than the BRAVE protocol. This is because the MoZo scheme has fewer location updates, whereas Brave requests location information more frequently and hence has more chances to correct the GPS errors.

7 Conclusion

This paper presents a novel moving-zone based architecture and a corresponding routing protocol for message dissemination in VANETs by using vehicle-to-vehicle communications only (i.e., without using vehicle-to-infrastructure communications). To the best of our knowledge, this is the first study that applies moving object techniques to vehicular networks. The moving object modeling and indexing techniques have been leveraged in various tasks including zone construction and maintenance as well as information dissemination. The proposed approach greatly reduces communication overhead and improves message delivery rate compared to other existing approaches.

References

[1] O. K. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar, "Broadcasting in VANET," in *Proc. IEEE Mobile Netw. Veh. Environments*, 2007, pp. 7–12.

[2] V. Naumov and T. Gross, "Connectivity-aware routing (CAR) in vehicular ad-hoc networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, 2007, pp. 1919–1927.

[3] W. Viriyasitavat, O. K. Tonguz, and F. Bai, "UV-CAST: An urban vehicular broadcast communication protocol," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 116–124, Nov. 2011.

[4] O. K. Tonguz, N. Wisitpongphan, and F. Bai, "DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 17, no. 2, pp. 47–57, Apr. 2010.

[5] P. Samar, M. R. Pearlman, and Z. J. Haas, "Independent zone routing: An adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, pp. 595–608, Aug. 2004.

[6] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 9, no. 1, pp. 69–72, 2005.

[7] C. Shea, B. Hassanabadi, and S. Valaee, "Mobility-based clustering in VANETs using affinity propagation," in *Proc. IEEE Global Telecommun. Conf.*, 2009, pp. 1–6.

[8] 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 *Proc. IEEE Congr. Evol. Comput.*, 2015, pp. 994–1002.

[9] Y. Peng, Z. Abichar, and J. M. Chang, "Roadside-aided routing (RAR) in vehicular networks," in *Proc. IEEE Int. Conf. Commun.*, 2006, pp. 3602–3607.

[10] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 84–94, Dec. 2007.

[11] I. Tal and G.-M. Muntean, "User-oriented cluster-based solution for multimedia content delivery over VANETs," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, 2012, pp. 1–5.

[12] Y. Shi, L. H. Zou, and S. Z. Chen, "A mobility pattern aware clustering mechanism for mobile vehicular networks," in *Proc. Appl. Mechanics Mater.*, vol. 130, 2012, pp. 317–320.

[13] C. S. Jensen, D. Lin, and B. C. Ooi, "Continuous clustering of moving objects," *IEEE Trans. Know. Data Eng.*, vol. 19, no. 9, pp. 1161–1174, Sep. 2007.

[14] J. Bernsen and D. Manivannan, "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification," *Pervasive Mobile Comput.*, vol. 5, no. 1, pp. 1–18, 2009.

[15] J. Kakarla, S. S. Sathya, B. G. Laxmi, and B. R. Babu, "A survey on routing protocols and its issues in VANET," *Int. J. Comput. Appl.*, vol. 28, no. 4, pp. 38–44, 2011.

[16] S. Singh and S. Agrawal, "VANET routing protocols: Issues and challenges," in *Proc. Recent Adv. Eng. Comput. Sci.*, 2014, pp. 1–5.

[17] T. Song, W. Xia, T. Song, and L. Shen, "A cluster-based directional routing protocol in VANET," in *Proc. 12th IEEE Int. Conf. Commun. Technol.*, 2010, pp. 1172–1175.

[18] P. Ruiz, V. Cabrera, J. Martinez, and F. Ros, "Brave: Beacon-less routing algorithm for vehicular environments," in *Proc. IEEE 7th Int. Conf. Mobile Ad-hoc Sensor Syst.*, 2010, pp. 709–714.

[19] P. Fan, G. Haran, J. Dillenburg, and P. Nelson, "Cluster-based framework in vehicular ad-hoc networks," in *Proc. Ad-Hoc Mobile Wireless Netw.*, 2005, pp. 32–42.

[20] M. Aoki and H. Fujii, "Inter-vehicle communication: Technical issues on vehicle control application," *IEEE Commun. Mag.*, vol. 34, no. 10, pp. 90–93, Oct. 1996.

[21] O. Kayis and T. Acarman, "Clustering formation for inter-vehicle communication," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2007, pp. 636–641.

[22] J. Chen, C. Lai, X. Meng, J. Xu, and H. Hu, "Clustering moving objects in spatial networks," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2007, pp. 611–623.

[23] Z. Wang, L. Liu, M. Zhou, and N. Ansari, "A position-based clustering technique for ad hoc intervehicle communication," *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)*, vol. 38, no. 2, pp. 201–208, Mar. 2008.

[24] W. Liu, J. Feng, Z. Wang, and H. Shan, "Multi-thresholds clustering objects in a road network," in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, 2008, pp. 686–689.

[25] K. Ibrahim and M. Weigle, "CASCADE: Cluster-based accurate syntactic compression of aggregated data in VANETs," in *Proc. IEEE GLOBECOM Workshops*, 2008, pp. 1–10.

[26] M. A. Alawi, R. A. Saeed, and A. A. Hassan, "Cluster-based multihop vehicular communication with multi-metric optimization," in *Proc. Int. Conf. Comput. Commun. Eng.*, 2012, pp. 22–27.

[27] K. Mershad, H. Artail, and M. Gerla, "We can deliver messages to far vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1099–1115, Sep. 2012.

[28] T. Little and A. Agrawal, "An information propagation scheme for VANETs," in *Proc. Int. IEEE Conf. Intell. Transp. Syst.*, 2005, pp. 155–160.

[29] R. Goonewardene, F. Ali, and E. Stipidis, "Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks," *IET Intell. Transport Syst.*, vol. 3, no. 2, pp. 148–158, Jun. 2009.

[30] Y. Luo, W. Zhang, and Y. Hu, "A new cluster based routing protocol for VANET," in *Proc. 2nd Int. Conf. Netw. Security Wireless Commun. Trusted Comput.*, 2010, pp. 176–180.

[31] Y. Ohta, T. Ohta, and Y. Kakuda, "An autonomous clustering-based data transfer scheme using positions and moving direction of vehicles for VANETs," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2012, pp. 2900–2904.

[32] S. Gurung, D. Lin, A. Squicciarini, and O. K. Tonguz, "A moving zone based architecture for message dissemination in VANETs," in *Proc. 8th Int. Conf. Netw. Service Manag.*, 2012, pp. 184–188.

[33] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proc. IEEE Intell. Vehicles Symp.*, 2003, pp. 156–161.

[34] J. Tian, L. Han, and K. Rothermel, "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks," in *Proc. IEEE Intell. Transp. Syst.*, vol. 2, 2003, pp. 1546–1551.

[35] B. -Chong Seet, G. Liu, B.-Sung Lee, C.-Heng Foh, K.-Juan Wong, and K.-Kee Lee, "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications," in *Proc. Netw.*, 2004, pp. 989–999.

[36] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, "Geographic routing in city scenarios," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 1, pp. 69–72, 2005.

[37] I. Leontiadis and C. Mascolo, "GeOpps: Geographical opportunistic routing for vehicular networks," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, 2007, pp. 1–6.

[38] N. Kumar and M. Dave, "BIIR: A beacon information independent VANET routing algorithm with low broadcast overhead," *Wireless Pers. Commun.*, vol. 87, pp. 869–895, 2015.

[39] *Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems*, www.astm.org/Standards/E2213.htm Standard ASTM E2213, 2003

[40] J. Lee, "Design of a network coverage analyzer for roadside-to-vehicle telematics networks," in *Proc. 9th ACIS Int. Conf. Softw. Eng. Artificial Intell. Netw. Parallel/Distrib. Comput.*, 2008, pp. 201–205.

Dan Lin received the PhD degree in computer science from the National University of Singapore, in 2007. She was a post doctoral research associate at Purdue University for two years. She is an associate professor and director of Cybersecurity Lab at the Missouri University of Science and Technology. Her main research interests cover many areas in the fields of database systems and information security.

Jian Kang received the BS degree in computer science from Fuzhou University, China, in 2013. He is currently working toward the master's degree at Fuzhou University, China. His research interest includes data preserving privacy.

Anna Squicciarini received the PhD degree in computer science from the University of Milan, Italy, in 2006. She is an associate professor at the College of Information Sciences and Technology, Pennsylvania State University. Her research is currently funded by the US National Science Foundation, Hewlett-Packard, and a Google Research Award.

Yingjie Wu received the PhD degree in computer science from Southeast University, China, in 2012. He is an associate professor at Fuzhou University, China. He is currently a visiting scholar at the Pennsylvania State University. His main research interests include data privacy and recommender systems.

Sashi Gurung received the bachelor's degree in computer engineering from the Institute of Engineering, Nepal. She received the PhD degree in computer science from the Missouri University of Science and Technology, in 2015. Her research interests include location privacy preserving in mobile objects databases, routing, and trustworthiness evaluation in VANET.

Ozan K. Tonguz is a tenured full professor in the Electrical and Computer Engineering Department, Carnegie Mellon University (CMU), Pittsburgh, Pennsylvania. He currently leads substantial research efforts at CMU in the broad areas of telecommunications and networking. He has published more than 300 papers in IEEE journals and conference proceedings in the areas of wireless networking, optical communications, and computer networks. He is the author (with G. Ferrari) of the book *Ad Hoc Wireless Networks: A Communication-Theoretic Perspective* (Wiley, 2006). He co-founded Virtual Traffic Lights, LLC, a CMU spinoff, in December 2010, which specializes in providing solutions to several transportation problems, such as safety and traffic information systems, using vehicle-to-vehicle and vehicle-to-infrastructure communications paradigms. His current research interests include vehicular ad hoc networks, wireless ad hoc and sensor networks, self-organizing networks, bioinformatics, and security. He currently serves or has served as a consultant or expert for several companies, major law firms, and government agencies in the United States, Europe, and Asia.