A Moving Zone Based Architecture for Message Dissemination in VANETs

Sashi Gurung
Department of Computer Science
Missouri University of Science and Technology
Email:sgy99@mst.edu

Dan Lin
Department of Computer Science
Missouri University of Science & Technology
Email: lindan@mst.edu

Anna Squicciarini
College of Information Sciences & Technology
Pennsylvania State University
Email: asquicciarini@ist.psu.edu

Ozan K. Tonguz
Department of Electrical and Computer Engineering
Carnegie Mellon University
Email: tonguz@ece.cmu.edu

***Abstract*—Vehicular Ad-hoc Networks (VANETs) are an emerging field, whereby vehicle-to-vehicle communication can contribute to many new applications such as infotainment services. Most VANET applications are based on routing protocols, the development of which however are challenging due to the dynamic nature of nodes (vehicles) in VANETs. To adequately capture the characteristics of VANET nodes, we propose a novel and unique approach that integrates moving object modeling and indexing techniques into VANET routing protocols. In particular, 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. The experiments have been carried out on real road maps and the results obtained demonstrate the superiority of our approach compared to existing solutions.**

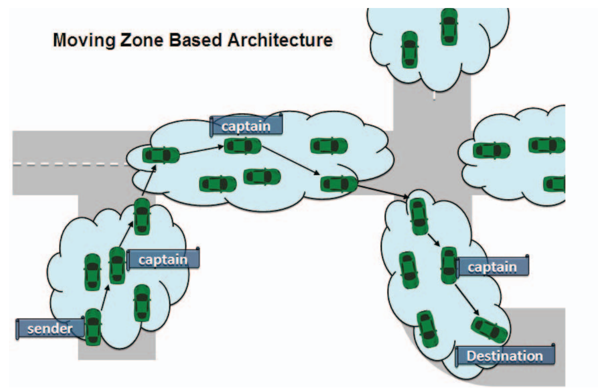

Fig. 1. Moving Zone Based Architecture in VANETs

I. 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), the benefits of VANETs would be tremendous. Various types of information, such as traffic conditions [1], [2], advertising news and e-coupons, can be shared among vehicles via VANETs. For example, a vehicle can send inquiries to vehicles around certain landmarks to obtain the up-to-date parking information. Another interesting emerging application, called Infotaiment services, 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 [3]. Existing Internet or MANET routing protocols are not suitable, due to the unique characteristics of VANETs, which include high mobility of VANET nodes, frequent changes in topology, and limited life time [4]. To address these issues, several VANET routing protocols have been proposed, which can be broadly classified into five main categories, namely broadcasting protocols [5], route-discovery protocols [6]–[8], position-based protocols [9], [10], clustering-based protocols [11] and infrastructure-based protocols [12]. 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, the development of clustering-based protocols is still at an early stage. Previously proposed clustering strategies are relatively straightforward in that they use simple clustering criteria, require a large amount of message exchange between member vehicles and cluster heads, and lack implementation details. Moreover, to date only a few simple message dissemination protocols have been proposed.

In this paper, we propose a more advanced decentralized clustering approach to construct a moving-zone based architecture in VANETs (as shown in Figure 1). Our approach integrates moving object modeling and indexing techniques 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. Our approach greatly

978-3-901882-48-7

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 when their moving direction or speed changed dramatically. Second, our approach allows the captain vehicle to estimate vehicle positions in the near future so that decisions (e.g., captain vehicle reassignment) can be made in a timely manner. Third, the use of index improves the scalability of our approach. There is no need to examine all member vehicles for each event or operation. Related vehicles can be quickly accessed via the index. As demonstrated by our experimental results, our approach is suitable even for highly dense roads, where a large number of vehicles travel at different speeds.

The rest of the paper is organized as follows. Section II reviews related work. Section III presents the moving-zone-based architecture, followed by Section IV which introduces the new routing protocol. Section V reports the experimental results. Finally, Section VI concludes the paper.

II. Related Work

Among clustering-based approaches, studies such as [11], [13]–[17] only provide clustering algorithms but do not have a specific algorithm about the use of clusters for routing purposes. Works that consider both clustering and routing are [18]–[21]. In [18], 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 reported. In [19], 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 [20], 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. The most recent work seems to be [21] by Song et al. They consider only moving direction for cluster head selection. In summary, none of the existing approaches considers the use of moving object techniques to reduce communication overhead, improve efficiency and effectiveness, as we present in this paper.

III. 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 other member vehicles as well as the message dissemination.

We consider vehicles moving on a road network represented as a graph with roads being edges and intersections being vertexes. Each vehicle's relative position on a road is modeled as a linear function of time: $l(t) = l_u + \delta \cdot v \cdot (t - t_u)$, where l_u is the vehicle's distance to the starting point of the road at time t_u, δ is vehicle's moving direction along the road, and v is the vehicle's speed. The vehicle's moving direction has value 1 if the vehicle moves toward the end point of the road, otherwise it is -1. $l(t)$ predicts the vehicle's position at a near future timestamp t ($t \geq t_u$). The functions are valid between two consecutive location updates sent by individual vehicles to their captain vehicles.

Moving zone construction starts from a vehicle logging onto the VANET. 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.

If a captain vehicle moving in the same direction (δ) receives the hello message, it sends back a response to the message sender. The response includes its unique identifier V_{cap}, current location, speed, and the next intersection that it is heading to. We will discuss how the captain vehicle in each zone is selected later in this section.

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. To implement this, we define the following similarity score based on the average distance between the two vehicles' anticipated trajectories within a certain time period.

$$Sim(V_1, V_2) \triangleq \frac{\Delta_t}{w_c|l_{c1} - l_{c2}| + w_m|l_{m1} - l_{m2}| + w_f|l_{f1} - l_{f2}|} \quad (1)$$

In Equation 1, l_{c1} (l_{c2}), l_{m1} (l_{m2}), and l_{f1} (l_{f2}) denote the positions of the two vehicles at current time t_c, the middle timestamp and future timestamp t_f, respectively. The 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. The future timestamp t_f is estimated as the time when the vehicle may change its moving function, e.g., approaching intersections, which can be determined by the road topology. 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. Moreover, the distance between vehicles is computed as a weighted distance, where $w_c > w_m > w_f$. The use of weights allows modeling of predicted positions that become less accurate as time passes. In the experiments, weights are assigned as 0.5, 0.3 and 0.2, respectively.

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.

Each captain vehicle needs to keep up-to-date information about its member vehicles in order to carry out message dissemination and zone maintenance (e.g., captain vehicle reassignment). 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. The B^+-tree uses a special encoding to store moving objects according to their location proximity, while the hash table provides quick access to objects according to their identities. Both data structures are very efficient in terms of insertion and deletion, which will not impose much workload to the captain vehicle.

In particular, each entry in the leaf node of the B^+-tree stores a member vehicle's **index key**, identity, the latest update timestamp t_u, location l_u, and speed v_u at t_u. The challenge is to design the index key that helps group vehicles with similar movement patterns into the same leaf node. Our idea is to use the relative positions of vehicles at the near future timestamp to be the index key. The rationale behind this approach is two-fold. First, vehicles' current locations may not be of immediate use for message dissemination or other tasks. It is more likely that the member vehicles participate in the events after they join the moving zone. Second, vehicles close to one another at current timestamp and move at similar speed, are expected to be close to one another in the near future. Therefore, vehicles' locations at a near future timestamp reflects the combined effect of the similarity of their current locations and moving speed to certain degree. To determine this near future timestamp, we partition the time axis into intervals of half of the maximum update interval τ_u, and we obtain a list of timstamps: t_s, $t_s + 0.5\tau_u$, $t_s + \tau_u$, $t_s + 1.5\tau_u$, $t_s + 2\tau_u$, ..., where t_s is the time that the captain vehicle becomes effective. Vehicles' location updates within each interval are mapped to the ending timestamp of that interval. To distinguish locations corresponding to different time intervals in the tree, we use a two-bit binary value to indicate the indexing timestamp as shown in Equation 2, where $[\_]_2$ means the binary value.

$$[t_i]_2 = [(t_u - t_s)/(0.5\tau_u) \;\; mod \;\; 3]_2 \qquad (2)$$

The reason to have the component "mod 3" is because vehicles will send at least one update within the maximum update interval τ_u and hence there will be at most 3 indexing timestamps co-existing in the index. For example, the corresponding indexing timestamp for the time intervals $[t_s,\ t_s + 0.5\tau_u]$, $[t_s + 0.5\tau_u,\ t_s + \tau_u]$, $[t_s + \tau_u,\ t_s + 1.5\tau_u]$ are "00", "01" and "10", respectively. After the third time interval, all vehicles in the first time interval $[t_s,\ t_s + 0.5\tau_u]$ should have already been updated, which means no more information about the first time interval exists in the tree. The indexing timestamp "00" is then reused to indicate the fourth time interval. The final indexing key for a new vehicle is the concatenation of the indexing timestamp and the binary values of its future location, as shown in Equation 3.

$$Key = [t_i]_2 || [l_i]_2 \qquad (3)$$

The Leaving Event (LE) queue stores the values of estimated timestamps when member vehicles may be out of the communication range of the captain vehicle, in an ascending order. 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.

We now summarize the process at the captain vehicle side. Initially, the captain vehicle creates the CLV-tree that stores its own information. After it accepted the joining request from a new vehicle, the captain vehicle first computes the new vehicle's index key according to Equation 3. Then, it inserts the information of the new vehicle into the CLV-tree. The insertion process is similar to that in the B^+-tree. Moreover, the captain vehicle also estimates the leaving time of the new vehicle, and update the LE queue. If the leaving time is not recorded in the LE queue, a new entry with this leaving timestamp and the pointer to the leaf node of the vehicle will be inserted. With the aid of the CLV-tree and the LE queue, the captain vehicle can identify the next captain vehicle for the replacement when the captain vehicle is about to leave the current zone. The vehicle which is not leaving soon and is stored in the middle of the CLV-tree 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.

IV. The 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. The routing protocol consists of the following steps.

Step 1: Suppose that the sender vehicle has a piece of information (I) that it would like to share with vehicles around location $l(x, y)$. The sender vehicle sends a message in the 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 to find 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. The captain vehicle queries the CLV-tree to find the member vehicles closest to location l_e. The chosen vehicle is called the propagation vehicle (denoted as V_p). If the message is located in the current moving zone, the routing ends.

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 introduced in Section III), 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, 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 the case that a captain vehicle from a different moving zone receives the message from V_p, this captain vehicle start operations in Step 2. In the case that 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 operations in Step 2 as well.

V. Experimental Study

We used TraNS [22], an integration of SUMO and ns-2, for VANET mobility and network simulation. In particular, we simulate 800 vehicles moving on the Manhattan map where the average road segment length is 961 meters. The experiments were conducted on a linux based 64-bit Intel(R) Xeon(R) E5630 2.53GHz machine. We compare our work with the latest clustering-based routing protocol namely CBDRP in [21]. The CBDRP first partitions roads into multiple segments, and then group vehicles moving on the same road segment with the same direction into one cluster.

We evaluate the performance of the routing protocols by varying the distance between the message sender and the receiver. In each round of experiments, we fix the total number of vehicles to 800 and the routing distance between each pair of the sender and the receiver to a value selected from 600 to 4000 meters. We observe that the MOZO protocol has a delivery rate close to 100% when the distance is 600 meters and then is decreased to 50% when the distance is 4000 meters. The delivery rate of the MOZO protocol is much higher than CBDRP which drops to zero right after the distance is greater than 600 meters. 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 chance of the established route being invalid increases, which severely affects the message delivery rate. Another observation is that the message delivery rate decreases when the message routing distance increases. This is because the longer the distance, the higher the chance that the message being dropped in the middle of the route due to various reasons such as sparse distribution of vehicles on certain road.

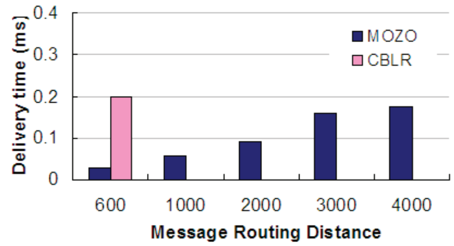

Fig. 2. Average Message Delivery Time

We also record the message delivery time which is measured from the sender vehicle sending out the message till the recipient vehicle received the message. Figure 2 shows the results. The time for CBDRP is not reported for distance greater than 600 meters because CBDRP has no message being delivered for long distances. We can observe that our MOZO routing protocol is faster than CBDRP. This is due to the fact that MOZO sends out the message directly when exploring the message route, while the CBDRP needs to establish route first and then send the actual message. Further, MOZO utilizes available routing information obtained from previous delivery. Since the clusters are relatively stable in MOZO, such prior information helps reduce the delivery time significantly. In addition, it is not surprising to see that the delivery time increases with the distance.

VI. Conclusion

This paper presents a novel moving-zone based architecture and a suitable routing protocol for message dissemination in VANETs. To the best of our knowledge, this is the first study that applies moving object techniques to the unique characteristics of VANET environments. The moving object modeling and indexing techniques have been leveraged in various tasks including zone construction, maintenance as well as information dissemination. The proposed approach achieves high delivery rate and short delivery time.

References

[1] C. Sommer, O. Tonguz, and F. Dressler, "Traffic Information Systems: Efficient Message Dissemination via Adaptive Beaconing," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 173–179, 2011.

[2] O. K. Tonguz, "Biologically Inspired Solutions to Fundamental Transportation Problems," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 106–115, 2011.

[3] F. Dressler, F. Kargl, J. Ott, O. Tonguz, and L. Wischoff, "Research Challenges in Intervehicular Communication—Lessons of the 2010 Dagstuhl Seminar," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 158–164, 2011.

[4] M. Aoki and H. Fujii, "Inter-vehicle communication: Technical issues on vehicle control application," *IEEE Communications Magazine*, vol. 34, no. 10, pp. 90–93, 2002.

[5] O. K. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar, "Broadcasting in VANET," in *INFOCOM Mobile Networking for Vehicular Environments*, 2007, pp. 7 – 12.

[6] V. Naumov and T. Gross, "Connectivity-aware routing (CAR) in vehicular ad-hoc networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2007, pp. 1919–1927.

[7] W. Viriyasitavat, O. K. Tonguz, and F. Bai, "UV-CAST: An Urban Vehicular Broadcast Communication Protocol," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 116–124, 2010.

[8] O. K. Tonguz, N. Wisitpongphan, and F. Bai, "DV-CAST: a distributed vehicular broadcast protocol for vehicular ad hoc networks," *IEEE Wireless Communnications Magazine*, vol. 17, pp. 47–56, 2010.

[9] P. Samar, M. R. Pearlman, and Z. J. Haas, "Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 595–608, 2004.

[10] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, pp. 69–72, 2005.

[11] C. Shea, B. Hassanabadi, and S. Valaee, "Mobility-based clustering in VANETs using affinity propagation," in *IEEE Global Telecommunications Conference*, 2010, pp. 1–6.

[12] Y. Peng, Z. Abichar, and J. M. Chang, "Roadside-aided routing (RAR) in vehicular networks," in *IEEE International Conference on Communications*, 2006, pp. 3602–3607.

[13] O. Kayis and T. Acarman, "Clustering formation for inter-vehicle communication," in *IEEE Intelligent Transportation Systems Conference*, 2007, pp. 636–641.

[14] J. Chen, C. Lai, X. Meng, J. Xu, and H. Hu, "Clustering moving objects in spatial networks," in *International conference on database systems for advanced applications*, 2007, pp. 611–623.

[15] W. Liu, J. Feng, Z. Wang, and H. Shan, "Multi-thresholds Clustering Objects in a road network," in *International Conference on Computer Science and Software Engineering*, 2008, pp. 686–689.

[16] K. Ibrahim and M. Weigle, "CASCADE: Cluster-based accurate syntactic compression of aggregated data in VANETs," in *IEEE GLOBECOM Workshops*, 2009, pp. 1–10.

[17] A. Squicciarini, D. Lin, and A. Mancarella, "Paim: Peer-based automobile identity management in vehicular ad-hoc network," in *Proc. of the IEEE Computer Software and Applications Conference (COMPSAC)*, 2011.

[18] T. Little and A. Agrawal, "An information propagation scheme for vanets," in *International IEEE Conference on Intelligent Transportation Systems*, 2005, pp. 155–160.

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

[20] Y. Luo, W. Zhang, and Y. Hu, "A new cluster based routing protocol for vanet," in *International Conference on Networks Security Wireless Communications and Trusted Computing*, 2010, pp. 176 – 180.

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

[22] M. Piórkowski, M. R., A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "Trans: realistic joint traffic and network simulator for vanets," *Mobile Computing and Communications Review*, vol. 12, no. 1, pp. 31–33, 2008.