Navigation Route Based Stable Clustering for Vehicular Ad Hoc Networks

Zhiwei Yang, Weigang Wu, Yishun Chen, Xiaola Lin, and Xiang Chen(✉)

School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
{yangzhw, wuweig, chenysh35, linxl, chenxiang}@mail.sysu.edu.cn

Abstract. Due to high mobility of vehicles, stability has been always one of the major concerns of vehicle clustering algorithms. In this paper, we propose a novel clustering algorithm based on the information of route planned by vehicular navigation systems. Including route information into cluster mechanism is not trivial due to two issues: (i) stability is a property of time rather than position, (ii) route diversity may cause high re-clustering overhead at road intersections. To address the first issue, we propose a function to quantitatively calculate the overlapping time among vehicles based on route information, with which a novel clusterhead selection metric is designed. To address the second issue, we design a mechanism of future-clusterhead, which can help avoid message exchanges at intersections. The simulation results show that, compared with similar works, our algorithm can cluster vehicles with higher stability and at the same time lower communication cost.

Keywords: VANET · Clustering · Mobile computing
Information dissemination · Ad hoc networks

1 Introduction

Vehicular Ad hoc NETwork (VANET) [1, 2] enables vehicles to communicate with roadside (V2R) or other vehicles (V2V) via wireless communications. VANETs can help drivers to acquire real-time information about road traffic status. One of major challenges in VANETs is the rapid change of network topology due to high mobility of vehicles. Establishing a cluster based hierarchy [3] is an effective and popular approach to cope with topology dynamics in ad hoc networks, including VANETs.

Most clustering algorithms use leadership metrics based on (relative) speed and distance. The SP-Clustering algorithm [4] is a typical example of this category, where the relative speed is calculated by the change of distance. More precisely, the distance of two vehicles is calculated using the signal strength of hello messages. Based on distance change, the algorithm can determine whether two nodes are closing to each other, and then delay cluster re-organization if possible. The affinity propagation (AP) algorithm [5, 6] is a distance-based clustering algorithm, which requires a lot of iterative loops that increase the delay time of cluster construction. Another type of leadership metric is

B. Li et al. (Eds.): ChinaCom 2017, LNICST 237, pp. 552–562, 2018.
https://doi.org/10.1007/978-3-319-78139-6_54

based on connection/link duration time. A vehicle needs to estimate the duration of the links with its neighbors and such a value is used to evaluate the priority of being clusterhead. The algorithms in [7, 8] choose the node with the longest connection duration as clusterhead, while the MA-Clustering algorithm in [9] takes the distance to destination points into consideration. The MA-Clustering algorithm takes into account the destination of vehicles, including the current location, speed, relative destination and final destination of vehicles, as parameter to arrange the clusters. The metric for clusterhead selection is the weighted sum of three parts: current distance of two vehicles, current speed difference of two vehicles and distance between their destination points.

Although navigation route has not been considered in clustering algorithms, it has been used in selecting data forwarding nodes in [10], where the term “trajectory” rather than “route” is adopted. Li et al. [11] propose a network coding with crowd sourcing-based trajectory estimation method to transmit data in vehicular networks. The estimation is completed by every node based on the pre-trajectory of GPS navigation. Network coding is used for data transmission according to the result of trajectory estimation. The STDFS algorithm proposed in [12] makes use of route information is a shared way. Such information is used to predict the encounters between vehicles, and a predicted encounter graph is constructed accordingly. Based on the encounter graph, STDFS optimizes the forwarding sequence to minimize delivery delay under a specific delivery ratio threshold. The TMC algorithm in [13] considers navigation route based multicasting. Route information is used to predict the chance of inter-vehicle encounter between two vehicles, and then the prediction result is further used to characterize the capability of a vehicle to forward a given message to destination nodes.

In this paper, we consider to improve the stability of clusters by making use of navigation route information. The route of a vehicle can be planned in advance by the navigation system. Such a route obviously indicates the future movement path of the corresponding vehicle [13, 14]. To improve the stability of clusters, vehicles with similar routes should be grouped into one cluster.

To construct a better cluster, we design a residual route time function, which quantitatively calculates the time during which two vehicles may keep to be neighbors. With this function, we design a metric to evaluate the priority of leadership (being clusterhead). Our metric also includes the number of neighbors as input, to guarantee the effectiveness of the cluster structure in terms of topology control.

To reduce the cost of cluster maintenance, we design a mechanism of future-clusterhead. Considering that some nodes in a new cluster may previously come from the same old cluster and have known routes of each other, we let only one of them exchange route data with nodes from other clusters for possible merging. Such a vehicle is called a future-clusterhead. Future-clusterheads are also elected based on route information collected during cluster forming.

To examine the performance of our algorithm, we conduct simulations using ns-3. The results show that, our navigation route based clustering algorithm can achieve higher stability and at the same time reduce communication cost.

The rest of the paper is organized as follows. Section 2 presents the system model and assumptions involved in our design. The proposed clustering algorithm is presented in Sect. 3. Performance evaluation is reported in Sect. 4 and finally Sect. 5 concludes the paper with future directions.

2 System Model and Assumption

In our work, we assume that each participating vehicle is equipped with a navigation system. The system can provide information like position (via GPS) and speed of the vehicle. Each vehicle has a predefined start location and destination location. The route of a vehicle is planned by the navigation system. A route should be set at the starting and may be dynamically changed on the way according to traffic conditions, but such changes should be infrequent.

The route data of a vehicle is a sequence of road segments and turn directions (turn left, turn right, no turn) at intersection between two adjacent segments. In each route, there is no repeated segment, and also no loops. For the simplicity of presentation, we assume the intersection is the typical cross of two roads. The traffic light is placed at the intersection to control the passing of vehicles. Intersections of other types, i.e. crossing of more roads, can be handled similarly.

Each vehicle is also equipped with a wireless communication device. Two vehicles are connected in ad hoc way. The wireless link is assumed to be reliable and no packets will be lost. Two vehicles within transmission range of each other can communicate directly and they are neighbors of each other. The transmission range is less than the length of a road segment. Heartbeat messages are periodically exchanged among neighbors to probe neighbors, so each vehicle knows its neighbors and maintain a neighbor list. A heartbeat message also carries the speed and position information of the sender.

The route of a vehicle consists of a sequence of road segments, and each segment can be represented by the corresponding intersection ID (or number) and direction to go. Such information is with small size, so it can also be included into the heartbeat messages when it is necessary. On the other hand, if the route information is too large to be integrated into heartbeat message, vehicles can exchange route information via special messages. Since navigation route of a vehicle is seldom changed after selected, exchange of route information can be done with a much longer period than heartbeat, and the overhead of route information exchange would be quite small.

3 The Proposed Algorithm

Same as most existing clustering algorithms, our proposed algorithm consists of two phases: cluster formation and cluster maintenance. However, our algorithm has an additional mechanism of future-cluster. Future-clusters are formed in the cluster formation phase and used in cluster maintenance phase to save communication cost.

In the following, we first present the definition and calculation of *RRT*, and then describe operations of cluster formation, future-cluster formation and cluster maintenance.

The status of a node may be:

- *Undecided state (UN): the node is not in any cluster.*
- *Clusterhead (CH): the node is a clusterhead.*
- *Cluster member (CM): the node is a member of some cluster, but not a clusterhead.*
- *Future-clusterhead (FCH): the node is a future-clusterhead, which is in charge of coordinating the merge of clusters.*

3.1 The Metric of Residual Route Time

In our design, we define RRT, which is the representation of the “overall” time duration of neighborhood between a vehicle and its current neighbors. Table 1 lists the notations used in the definition of *RRT*.

Table 1. Notations used

Notations	Meaning	Notations	Meaning
R	The transmission range	l_i	The current position of node i
N_i	The set of neighbor nodes of i	v_i	The current speed of node i
n_i	Number of neighbors, i.e. $n_i = \lvert N_i \rvert$	v^k	The expected speed of vehicles at road segment k
Γ_i	The planned route of node i	p_{ijk}	The probability that nodes i and j being neighbors at road segment k
r_k	The length of road segment k		

For a pair of neighbor nodes i and j, we can estimate t_{ij}, the time they will keep to be neighbor in future trip:

$$t_{ij} = \frac{R \pm |l_j - l_i|}{|v_j - v_i|} + \sum_{k \in \Gamma_i \cap \Gamma_j} \left(\frac{r_k}{v^k} p^k_{ijk}\right) \tag{1}$$

The calculation of t_{ij} consists of two parts. The first part is the short-term estimation, which estimates the time that two vehicles will keep to be neighbors in the current road segment. It can be calculated using the current speed and position information. The second part is the long-term estimation, which is the time that two vehicles may be neighbors of each other in the future overlapping road segments. Since the vehicles have not entered these segments yet, such a time duration can only be calculated based on the expected speed obtained from historical data v_{ik}. We introduce the parameter p_{ijk} to denote the probability that two neighboring nodes become disconnected at road k. Such a value can be obtained by historic data analysis.

Then, considering all neighbor nodes of i, we have the total time of neighborhood, and the number of neighbor nodes. Therefore, we add two parameters in the final value of residual route time *RRT*, i.e.:

$$RRT = \theta_i t_i \text{ where : } \theta_i = \frac{1}{n_i} \sum_{j \in N_i} \frac{n_i}{n_j} = \sum_{j \in N_i} \frac{1}{n_j},\ t_i = \sum_{j \in N_i} t_{ij} \tag{2}$$

Finally, we have the complete definition of *RRT*:

$$RRT = \sum_{j \in N_i} \frac{1}{n_j} \sum_{j \in N_i} \left[\frac{R \pm |\ell_j - \ell_i|}{|v_i - v_j|} + \sum_{k \in \Gamma_i \cap \Gamma_j} (\frac{r_k}{v^k} p_{ijk}^k) \right] \quad (3)$$

The above calculation of RRT focuses on the effect of vehicle movement and the quality of wireless link between neighboring nodes.

3.2 Cluster Formation

Initially, there are no clusters and all the vehicles are in undecided (UN) state. The clustering algorithm is initiated by the upper layer application or other mechanisms. The first step is exchanging HELLO messages with its neighbors to collect information used in the calculation of *RRT*. More precisely, the HELLO message contains neighbor number and route data. Notice that in the beginning, the nodes do not know their neighbors and the neighbor number in the HELLO message is set to be zero. Later, the number of neighbors is changed according to the HELLO messages from neighbors.

Upon receiving HELLO messages from neighbors, a vehicle i will calculate its RRT accordingly, and then includes its *RRT* value into its HELLO message. To adapt the dynamic changes of network topology and navigation route, *RRT* value is updated upon the detection of such changes. Obviously, i will receive HELLO messages containing *RRT* value from its neighbors in the third and later rounds.

In the initial state, each node is in the UN state, so there is no clusterhead in the neighbor list of any node. When an UN node detects no clusterhead in its neighborhood, it will start the clusterhead election procedure. It will firstly include all UN neighbors into a CH election list, excluding those without sharing/overlapping navigation road segments. Notice that the CH election list contains only UN nodes, and a non-UN node will not be included, even though it has a greater *RRT*. If vehicle i has a greater *RRT* value than all its neighbors in the CH election list, i itself is selected as a clusterhead by switching to the CH.

In later rounds, when one or more HELLO messages from clusterhead are received, i will choose to join the cluster with the highest *RRT* value by sending a $JOIN(CH_{id}, UN_{id})$ message. The corresponding clusterhead will send an $ACK(CH_{id}, UN_{id})$ message to confirm the join. Vehicle i then switches to be in the CM state.

On the other hand, if the UN node i does not receive HELLO message from clusterhead (all neighbors with higher *RRT* join clusters of other vehicles), i will wait for more rounds until a suitable CH is found or it has the greatest RRT among all the UN neighbors.

Since at each round, at least one UN node changes its status to CH or CM, eventually each UN node will decide its status and the cluster formation procedure stops.

3.3 Future-Cluster

The vehicles in one cluster may turn to different directions at the next intersection. Accordingly, we divide one cluster into multiple future-clusters, each of which contains vehicles that will turn to the same direction. Then, after passing the intersection, vehicles in the same future-cluster can simply form a new, even without exchanging node status information. This can help reduce the overhead of cluster maintenance, in terms of both message and time.

Since each cluster is coordinated and managed by its clusterhead, the formation of future-clusters is also conducted by clusterhead. After a cluster is formed, the clusterhead must have collected route data from all members.

The future cluster formation is triggered each time the clusterhead detects that it has entered a new road. By checking the route of cluster members, the clusterhead can divide them into future-clusters according to their direction at the next intersection. At a typical intersection with two roads cross with each other (other intersection scenarios can be handled similarly), vehicles will take one of three directions: LT (left turn), RT (right turn) or NT (no turn). Then, at most three future-clusters may be formed within one cluster.

Accordingly, the clusterhead can compare *RRT* values of all members and assign the vehicle with the largest *RRT* in a future-cluster to be the future-clusterhead. To keep stable, the clusterhead will choose itself as the future-clusterhead of its own future-cluster. After the clusterhead of a cluster determines future-clusters and assigns future-clusterhead, it will broadcast the results to all members, via a *FCH(fch_list, fch_fcm_list)* message. And each member will learn about its own future-cluster and future-clusterhead.

Notice that a node assigned to be future-clusterhead will not start the cluster merging procedure until it passes the corresponding intersection. On the other hand, the cluster member maintenance mechanism of a future-cluster, which is similar to the mechanism of a cluster, will begin immediately after the future-clusterhead is assigned.

3.4 Cluster Maintenance

After clusters are already formed, the cluster members and clusterhead of a cluster monitor each other via HELLO messages. The HELLO message of a cluster member is different from common HELLO messages. It contains the ID of the cluster it belongs to, which may be in fact the ID of the clusterhead. When a cluster member is disconnected from its clusterhead due to speed difference or turning at intersections, it needs switch to another cluster. On the other hand, new clusterhead may be selected due to split and merging at intersections. Moreover, a cluster should be destroyed if there are too few members or a clusterhead with higher *RRT* value is in the neighborhood. Such cases are handled by cluster maintenance mechanism.

Cluster switch. When a cluster member *i* detects disconnection from its current clusterhead, it will switch to UN state, and try to switch to some cluster with the clusterhead that is currently in neighbor list by sending a $JOIN(CH_{id}, UN_{id})$ message.

The following operations are the same as in the cluster formation procedure. When the target clusterhead j receives the $JOIN(CH_{id}, UN_{id})$ message, it will reply with an $ACK(CH_{id}, UN_{id})$ message and i becomes a new member of cluster j.

Cluster merging. The case of cluster change at intersection is more complex. After passing an intersection, one cluster will be split into three or less future-clusters. Once a future-clusterhead passes the intersection, it will starts the procedure of cluster merging with other cluster/future-clusters that enter the same lane.

The future-clusterhead will first try to merge with other clusters. It will prepare the candidate clusterhead list by adding the clusterhead neighbors and excluding those without overlapping route and those not entered the current lane yet. Then, the future-clusterhead will select the node with the greatest *RRT*, say CH_{id}, from candidate clusterhead list, and send a $QUERY(CH_{id}, FCH_{id}, fcm\_list)$ message to CH_{id}, which carries future-clusterhead id and future-cluster member list. Upon receiving a $QUERY(CH_{id}, FCH_{id}, fcm\_list)$ message, the clusterhead will reply by broadcasting $RESPONSE(CH_{id}, FCH_{id})$. Corresponding to the $RESPONSE(CH_{id}, FCH_{id})$, a cluster member can join the clusterhead CH_{id}, along with future-clusterhead FCH_{id}. Notice that we let only clusterhead respond to a query to reduce message cost.

On the other hand, if no clusterhead is in the candidate clusterhead list, clusterhead re-election is unavoidable. To reduce role changes and simplify operations, clusterhead re-election is done among only future-clusterheads. The future-clusterhead with the greatest RRT is elected as the new clusterhead, and other future-clusterheads will join the new cluster together with their members.

Cluster destruction. A cluster may be destroyed if there are too few members. A clusterhead keeps monitoring its members and neighbor *RRT* value via HELLO messages. If the number of members decreases and becomes less than a predefined threshold for a time long enough, the clusterhead will destroy the cluster by switching to be a future-clusterhead and try to join another cluster. The rest operations are the same as cluster merging.

4 Performance Evaluation

To evaluate the performance of our algorithm, we conduct simulations using ns-3. The mobility of vehicles is simulated via SUMO. To accurately examine the effect of our proposed future-clusterhead mechanism, we simulate our algorithm under two different variants: RT-Clustering is the variant without future-clusterhead and RTF-Clustering is the variant with the future-clusterhead mechanism. For comparison purpose, we also simulate three representative existing algorithms: ID-Clustering [15], SP-Clustering [4], and MA-Clustering [9]. We set a road network of a grid topology, with 4 horizontal roads and 4 vertical roads. Each road has eight lanes, four for each direction. The crossing point of two roads is viewed as an intersection. Each road segment between two crossing points is set to be 1 km. The maximum speed of vehicles is varied from 10 m/s to 35 m/s. The value of p_{ijk} plays a significant role in our algorithm. In our simulation, we adopt the value 0.8 in the discussion of simulation results.

We adopt four metrics to measure the performance of clustering algorithms.

- *Average clusterhead lifetime (CHT)*: the average consecutive time a node acts as a clusterhead.
- *Average cluster member lifetime (CMT)*: the average consecutive time a node acts as a cluster member. This metric is similar to *CHT*.
- *Average cluster size (ACS)*: the average number of nodes in a cluster.
- *Number of messages sent per node (NMS)*: the average number of messages of a node sent to form and maintain the cluster architecture.

We now present and discuss simulation results according to metrics. The confidence interval with confidence level 90% is shown in the result figures.

(1) *Lifetime of Clusters*

The stability of clusters is indicated by the lifetime of clusters, i.e. *CHT* and *CMT*, as shown in Figs. 1 and 2 respectively.

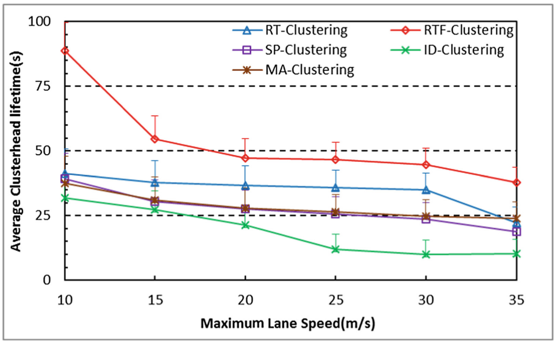

Fig. 1. Average clusterhead lifetime

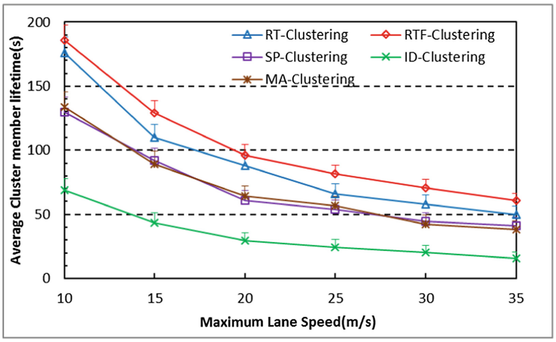

Fig. 2. Average cluster member lifetime

From both figures, we can easily see that, with the increase of lane speed, the lifetime of both clusterhead and cluster member decreases significantly. Among the three algorithms, ID-Clustering achieves the shortest lifetime. This is because ID-Clustering does not consider the movement of vehicles at all. Our proposed algorithm outperforms the others in both CHT and CMT. SP-Clustering and MA-Clustering considers the speed and direction of vehicles, but only the current movement status is considered. With the help of navigation route information, our algorithm can select clusterheads with more stable links, where the lifetime in CHT/CMT can be as high as twice of that of SP/MA-clustering. The different performance of RTF and RT shows clearly the benefit of our future-cluster mechanism.

Moreover, compared with RT/RTF-Clustering and ID-Clustering, SP-Clustering is more sensitive to lane speed, which is shown in both CHT and CMT. That is, with land speed increases, the lifetime of SP-Clustering decreases faster than the other two algorithms. This can be explained as follows. Since SP-Clustering considers only the current speed and direction status of vehicles, under a higher lane speed, such information is valid for a shorter time, then more cluster switches will occur. In our RT-Clustering algorithm, however, we consider future segment and speed, and the effect of current speed and direction will be reduced.

(2) *Average Cluster Size*

Figure 3 shows the average cluster size (*ACS*) of different clustering algorithms. The cluster size of ID-Clustering is the smallest, with a value always less than 2.0. The cluster in RT-Clustering has about 4.0 members in average, and SP-Clustering's cluster size is about 3.0. With the consideration of route information, our algorithm constructs larger clusters than other algorithms do. This may be the benefit of future cluster merging at intersections, which try to merge future clusters into a large one.

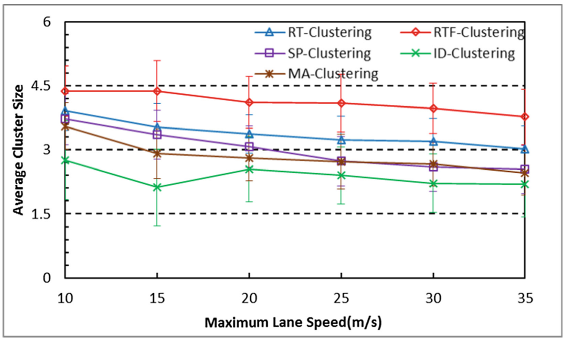

Fig. 3. Average cluster size

(3) *Number of messages sent per node*

NMS measures the communication cost for constructing and maintaining clusters. Figure 4 shows the cost of all the algorithms. Our algorithm performs better than all

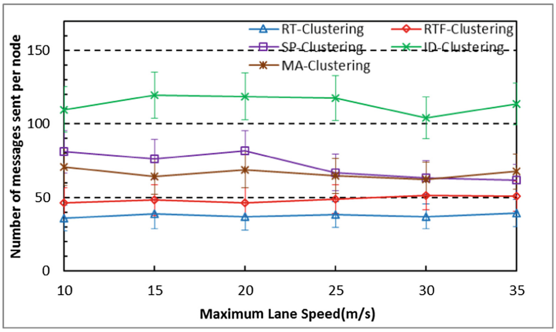

Fig. 4. Number of messages sent per node

others in all cases. This is certainly the benefit of high stability of clusters constructed. Since our algorithm can construct clusters with high stability, the message cost of cluster construction and maintenance should be much less than SP/MA-Clustering.

It is more interesting to compare RT-Clustering and RTF-Clustering. RTF-Clustering, the variant with Future-cluster mechanism, causes more communication cost than RT-Clustering, although the difference is not very large. We explain the additional communication cost of RTF as below. Future-clusterhead mechanism needs to exchange messages for merging future-clusters. On the other hand, RT-Clustering may simply keep future-clusters run individually to avoid re-clustering overhead. This is consistent with the results of cluster size in Fig. 3.

5 Conclusion and Future Work

In this paper, we propose a novel clustering algorithm for VANETs by considering navigation route of vehicles. Based on the overlapping road segments of routes from different vehicles, we design a function to estimate the time that two vehicles may keep to be neighbors in future trip. Clusterheads are elected based on the overall time that a vehicle can keep its neighborhood in future. Compared with existing clustering algorithms, our solution can improve cluster stability, in terms of various performance metrics.

Further study is certainly necessary. Possible directions include constructing models to estimate the probability of being neighbor in future road, optimizing the cluster size with respect to the upper layer applications.

References

1. Jaiswal, P.K., Jaidhar, C.D.: Location prediction algorithm for a nonlinear vehicular movement in VANET using extended Kalman filter. Wirel. Netw. **23**(7), 2021–2036 (2017)
2. Nasr, M.M., Abdelgader, A.M., Wang, Z., et al.: VANET clustering based routing protocol suitable for deserts. Sensors **16**(4), 1–23 (2016)

3. Caballerogil, C., Caballerogil, P., Molinagil, J., et al.: Self-organized clustering architecture for vehicular ad hoc networks. Int. J. Distrib. Sens. Netw. **11**(8), 1–12 (2015)
4. Souza, E.D., Nikolaidis, I., Gburzynski, P.: A new aggregate local mobility (ALM) clustering algorithm for VANETs. In: International Conference on Communications, pp. 1–5 (2010)
5. Shea, C., Hassanabadi, B., Valaee, S., et al.: Mobility-based clustering in VANETs using affinity propagation. In: Global Communications Conference, pp. 1–6 (2009)
6. Hassanabadi, B., Shea, C., Zhang, L., et al.: Clustering in vehicular ad hoc networks using affinity propagation. Ad Hoc Netw. **13**, 535–548 (2014)
7. Bononi, L., Felice, M.D.: A cross layered MAC and clustering scheme for efficient broadcast in VANETs. In: International Conference on Mobile Adhoc and Sensor Systems, pp. 1–8 (2007)
8. Ni, M., Zhong, Z., Zhao, D., et al.: MPBC: a mobility prediction-based clustering scheme for ad hoc networks. IEEE Trans. Veh. Technol. **60**(9), 4549–4559 (2011)
9. Morales, M.M., Hong, C.S., Bang, Y., et al.: An adaptable mobility-aware clustering algorithm in vehicular networks. In: Asia-Pacific Network Operations and Management Symposium, pp. 1–6 (2011)
10. Cunha, F.D., Villas, L.A., Boukerche, A., et al.: Data communication in VANETs: protocols, applications and challenges. Ad Hoc Netw. **44**, 90–103 (2016)
11. Li, L., Yang, Z., Wang, J., et al.: Network coding with crowdsourcing-based trajectory estimation for vehicular networks. J. Netw. Comput. Appl. **64**, 204–215 (2016)
12. Xu, F., Guo, S., Jeong, J., et al.: Utilizing shared vehicle trajectories for data forwarding in vehicular networks. In: International Conference on Computer Communications, pp. 441–445 (2011)
13. Jiang, R., Zhu, Y., Wang, X., et al.: TMC: exploiting trajectories for multicast in sparse vehicular networks. IEEE Trans. Parallel Distrib. Syst. **26**(1), 262–271 (2015)
14. Jeong, J., Guo, S., Gu, Y., et al.: TBD: trajectory-based data forwarding for light-traffic vehicular networks. In: International Conference on Distributed Computing Systems, pp. 231–238 (2009)
15. Lin, C., Gerla, M.: Adaptive clustering for mobile wireless networks. IEEE J. Sel. Areas Commun. **15**(7), 1265–1275 (1997)