A Study of the Impact of Merging Schemes on Cluster Stability in VANETs

Mengying REN*, Jun ZHANG†, Lyes KHOUKHI*, Houda LABIOD† and Véronique VÈQUE‡
* University of Technology of Troyes, Troyes, France
Email: {mengying.ren, lyes.khoukhi}@utt.fr
†Telecom ParisTech, University of Paris-Saclay, France
Email: {mengying.ren, jun.zhang, houda.labiod}@telecom-paristech.fr
‡University of Paris-Sud, University of Paris-Saclay, France
Email: veronique.veque@u-psud.fr

***Abstract*—Effective clustering algorithms are indispensable in order to solve the scalability problem in Vehicular Ad-hoc Networks (VANETs). Due to the highly dynamic network topology, an effective cluster merging scheme is always required in clustering algorithms, aiming to prevent the collapse of clusters. In the literature, there is a lack of comparison of cluster merging schemes, which makes it hard to analyze the impact of this component on clustering performance. In this paper, we analyze the existing cluster merging schemes and propose a Leadership-based Cluster Merging (LCM) scheme. Then, a comprehensive comparison of different cluster merging schemes under various traffic scenarios is presented, and our scheme is shown to achieve better performance on cluster stability.**

I. Introduction

In the near future, vehicles will have the capability to communicate with each other directly in a Vehicle-to-Vehicle (V2V) manner or indirectly using the existing infrastructure alongside the road in a Vehicle-to-Infrastructure (V2I) way [1]. This will enable the implementation of numerous Intelligent Transportation Systems (ITS) applications, including road safety, traffic efficiency, and infotainment applications, assisting drivers in avoiding dangerous situations or provisioning of convenience applications for passengers.

Considering that ITS applications require information to be delivered in multiple hops away, VANETs is becoming one of the most widely distributed and largest scale ad hoc networks. Therefore, scalability problem becomes a big challenge in VANETs. To solve the network scalability problem, a hierarchical network structure has been proposed, where vehicles are divided into several virtual groups, called clusters [3]. Each cluster has at least one cluster head (CH), acting as a group leader, and one or more than one cluster members (CM). Vehicles in the same cluster can communicate directly with their leader vehicle via an intra-cluster communication, while inter-cluster communication can be achieved by CHs.

In recent years, many clustering algorithms for VANETs have been proposed. As been addressed in [4], most clustering schemes incline to form a small number clusters with large cluster size. Therefore, when two clusters approach one another, they intend to merge to form a single larger cluster. Then a new CH will be selected for the merging cluster. The process of combining two neighboring clusters into a single larger cluster is defined as a cluster merging process.

In this paper, we analyze the rationality of different cluster merging schemes for the clustering algorithms in VANETs and propose a Leadership-based Cluster Merging (LCM) scheme. The term "leadership" indicates the capability of coordination with CMs. We assign the vehicle with a better leadership to be the new cluster head of the newly merged cluster, so as to ensure that it can maintain stable connections with all members of the cluster. As far as we know, this is the first paper to study the impact of the component of cluster merging in the clustering algorithms for VANETs.

The rest of the paper is organized as follows: Section II discusses the related work of cluster merging schemes. Section III describes the proposed clustering framework [11]. Then, a Leadership-based cluster merging scheme is presented. Section IV introduces the simulation environment and provides a comprehensive comparison of the cluster performance of different cluster merging schemes. Section V concludes the paper and briefly presents our future work.

II. Related Work

According to the existing vehicular clustering schemes, cluster merging process is always triggered when two CHs approach one another and become one-hop vicinities. In order to guarantee the stability of the merged cluster and to decrease vehicle re-clustering frequency, most existing cluster merging schemes require two neighboring CHs to stay in the transmission

[1]This work is supported by the project V2V4D2D from the department STIC of University of Paris-Saclay, France.

978-1-5386-3531-5/17/$31.00 © 2017 European Union

range (TR) of each other for a short time period, which is defined as contention time or merge interval (MI), instead of starting cluster merging immediately.

According to the research, there are two common strategies to select the new CH in the merging cluster. The first one is to select the CH that is attached with more CMs, denoted as "CM-based", as adopted by [5], [6], and [7]. In [5], cluster merging takes place when two CHs come within each other's transmission range and their speed difference is within the predefined threshold Δv_{th}. The CH that has a lower number of CMs simply gives up the CH role and becomes a CM in the new cluster. The rest CMs automatically join the neighboring cluster if they are in the transmission range of the CH and the speed difference is within the threshold. Similar to [5], in [6], when cluster merging happens, the cluster with fewer CMs is dismissed and these CMs try to join other clusters, launching a new CM re-clustering stage. The "CM-based" strategy aims to reduce cluster member disconnections. However, such a strategy cannot guarantee the stability of link connections between the new CH and its members.

The second strategy is to select the CH that has better stability within its original cluster, as adopted by [8], denoted as "VMaSC-based". During cluster merging, two CHs compare their averaged relative speed, called AVGREL_SPEED in their original clusters, respectively. The CH with higher average relative speed gives up its CH role and affiliates to the CH with lower average relative speed as a CM. Similar to [8], both the cluster merging schemes proposed in [9] and [10] select the new CH according to predefined vehicles' stability metrics (Aggregated Local Mobility (ALM) in [9] and Befit Factor (BF) in [10]) in the original clusters. The intuition of this strategy is assuming that the CH's stability in its original cluster is representative for its stability in the newly merged cluster. However, in reality, a higher stability in the original cluster cannot guarantee a better cluster performance in the merged cluster.

In this paper, the proposed cluster merging scheme will select a merging cluster head, which not only has stable links with members in the original cluster but also has stable links with other members in another cluster to be merged. The strategy is denoted as a leadership-based cluster merging scheme, as the selected vehicle provides better coordination of the merged cluster.

III. Algorithm Description

A Leadership-based Cluster Merging scheme is presented in this paper, based on our vehicle clustering framework, proposed in [11]. All vehicles are assumed to be equipped with a GPS which provides vehicle's basic information, including position, velocity, and moving direction. Moreover, each vehicle can both calculate speed difference and detect relative distance with respect to its vicinities.

Vehicles exchange their information periodically with its one-hop vicinities via *Beacon* messages at every Beacon Interval (BI). Information contained in *Beacon* message includes vehicle's identifier *ID*, current state *S*, cluster identifier *CHID*, current position *P*, current velocity *v*, and moving direction *Dir*. Fig. 1 presents an example of cluster network topology. Two clusters are presented on the road, with a single CH in each cluster, CH_f and CH_b.

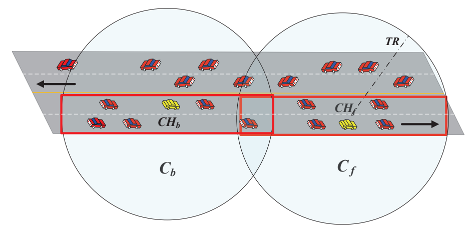


Fig. 1. Cluster C_f and C_b.

A. Vehicle states

In our proposed clustering scheme, each vehicle operates one of the following 4 states:

- Undecided Node (UN): The initial state of the vehicle, indicating that the vehicle does not belong to any cluster.
- Cluster head (CH): The leader vehicle of the cluster, being able to communicate with all of its members. Each cluster has only one CH.
- Cluster Member (CM): The member vehicle which is directly attached to an existing CH.
- Candidate Cluster Member (CCM): The vehicle that intends to be a CM of an existing cluster, but has not yet received a confirmation message from the corresponding CH.

The rest of this section will describe the clustering process from the aspects of cluster formation, cluster updating, and cluster merging.

B. Cluster formation

1) At the beginning of clustering, each vehicle is a UN node. The system starts a timer, called $T_{collect}$, during which vehicles exchange and collect beacon messages, in order to discover their one-hop vicinities. Every vehicle only maintains and updates the information of the vehicles which have the same direction and limited speed difference, $\Delta v \leq \Delta v_{th}$, where Δv_{th} is the speed difference threshold.
2) When the timer $T_{collect}$ expires, every vehicle (UN) calculates its own averaged relative mobility metric among its neighbors, and vehicles with the lowest metric values claim themselves as CHs. Instead of exchanging their relative mobility information, every vehicle starts a contention mode by setting an independent back-off timer T_W. Vehicle

with a lower relative mobility should set a smaller T_W, meaning the higher probability of becoming a CH.

3) UN node broadcasts a CH Announcement message (CHA) to inform its neighboring nodes that it becomes a CH vehicle; otherwise, if it hears CHA from another node during its contention process, it quits the contention mode and cancels the T_W; meanwhile, it changes UN state to a Candidate Cluster Member (CCM) state, sends a *ReqJoin* message to the CH and sets a timer of waiting for confirmation, T_{ACK}, trying to join this cluster.
4) Upon receiving a *ReqJoin* message, the CH firstly checks whether the total number of CMs is less than its cluster capacity, the maximum number of vehicles in the cluster, denoted as *MAX_CM*. If yes, the CH adds CCM vehicle to its cluster member list and sends back an *ACKJoin* message; otherwise, it simply ignores the request.
5) As long as CCM vehicle receives the confirmation message *ACKJoin* before T_{ACK} expires, it changes the state from CCM to CM immediately and resets its cluster identifier *CHID*; otherwise, it waits to hear another CH.

C. Cluster merging

1) Merging condition check: With time increasing, more and more single CMs may change state to CHs, which makes clustering meaningless. Cluster merging process aims to combine two small clusters to form one larger cluster. When two neighboring CHs are moving in the same direction within the transmission range of each other, cluster merging detection process will be triggered. Assuming that there are two clusters on the road, cluster C_1 and cluster C_2 with two CHs respectively, CH_1 and CH_2 (1 and 2 are node identifiers, and the *ID* of cluster *CHID* is represented by the *ID* of its CH). As shown in Fig. 1, the CH moving in front is denoted as CH_f, and the CH moving in the back is denoted as CH_b (b and f indicate the relative position of CH). Cluster merging detection process is triggered as long as CH_b receives a *Beacon* message from CH_f. CH_b will start a contention timer, called Merge Interval (MI), in order to avoid frequent CM re-clustering. CH_b will check the merging condition only if it can receive the *Beacons* consecutively from CH_f during MI time period. Otherwise, these two clusters cannot be merged.

When MI expires, CH_b will check the merging conditions, listed as follows. If all of the conditions are satisfied, CH_b will send a *ReqMerge* message to CH_f.

- Two clusters are moving in the same direction;
- The number of CMs in the merged cluster is less than the predetermined cluster size *MAX_CM*, the value of *MAX_CM* will be addressed in Section IV;
- The difference between the mean relative speed of two clusters should satisfy $\Delta\overline{V_{Cfb}} \leq \Delta\overline{V_{Cth}}$. The mean relative speed of the cluster is described as the averaged speed difference between a CH and all of its CMs, shown in Eq. (1) and Eq. (2).

$$\Delta\overline{V_{Cfb}} = |\overline{V_{Cf}} - \overline{V_{Cb}}| \tag{1}$$

$$\overline{V_{Cf}} = \frac{\sum\limits_{k\in C_f} |\vec{v_f} - \vec{v_k}|}{N_f}, \tag{2}$$

where $\overline{V_{Cf}}$ is the average speed difference between CH_f and CMs of cluster C_f, $\vec{v_f}$ and $\vec{v_k}$ indicate the speed of CH_f and CM_k respectively, vehicle k is one of the CMs in cluster C_f, and N_f is the number of CMs in C_f.

Note that only clusters that have the same moving direction can be merged, in order to avoid frequent link disconnections between vehicles moving in the opposite directions. A predetermined cluster size *MAX_CM* is applied, indicating the maximum number of CMs in a cluster, to avoid overloading cluster's resource capacity, as well as to increase the cluster stability.

2) Leadership-based CH_{merge} selection: Upon the reception of *ReqMerge* message from CH_b, a leader in the newly merged cluster, denoted as CH_{merge}, should be selected by CH_f. The node with higher leadership is considered to have better link stability with neighboring members and will be selected as CH_{merge}. The definition of "leadership" is described as follows.

We define the stability factor $SF_{m,Cn}$ as the stability between vehicle m and all of the CMs in C_n, detailed in Eq. (3). It is represented by the averaged speed difference between a vehicle m and all of the CMs in cluster C_n (m and n are vehicles' identifiers and can be the same).

$$SF_{m,Cn} = \frac{\sum\limits_{k\in C_n} |\vec{v_m} - \vec{v_k}|}{N_n}, \tag{3}$$

where vehicle k is one of the CMs of cluster C_n and N_n is the number of CMs in C_n. A smaller $SF_{m,Cn}$ indicates that the mobility pattern of vehicle m is more similar to the vehicles in the cluster C_n. Then, we denote the leadership of a vehicle m in cluster C_n as $L_{m,Cn}$, normalized between 0 and 1, shown in Eq. (4). A higher leadership represents a better link stability between vehicle m and the CMs in cluster C_n.

$$L_{m,Cn} = \frac{1}{1 + SF_{m,Cn}}. \tag{4}$$

The details of the leadership-based merging CH selection process is described in Algorithm 1. Upon the reception of *ReqMerge* message from CH_b, the forward CH CH_f computes its leadership $L_{f,Cf}$ and the leadership of CH_b in cluster C_f, denoted as $L_{b,Cf}$, according to Eq. (3) and Eq. (4). If the leadership of CH_b in the cluster C_f is higher than that of CH_f, where

$L_{b,Cf} > L_{f,Cf}$, CH_b is more intuitive to be the leader becomes CH_{merge}. Otherwise, the forward CH_f keeps on being the leader and claims itself as CH_{merge}.

As long as the new CH is selected, CH_f sends back a *ACKMerge* message to CH_b, informing CH_b the information of CH_{merge} by $flag\_CH_{merge}$. If CH_f keeps on being the leader, $flag\_CH_{merge} = 0$, and CH_b broadcasts a message to inform its original CMs to change the identifier of CH, *CHID*. Meanwhile, CH_b becomes a CM of CH_f. On the contrary, if CH_b becomes the merging CH, where $flag\_CH_{merge} = 1$, CH_f broadcasts a message to inform its original CMs to change their *CHID*, and itself becomes a CM of the merging cluster.

Algorithm 1 Leadership-based merging CH selection

```
For cluster head moving in front CH_f, on receiving a ReqMerge message from CH_b:
CH_f calculates L_{f,Cf} and L_{b,Cf}, according to Eq. (3) and Eq. (4)
if L_{b,Cf} > L_{f,Cf} then
    CH_f sends back a ACKMerge message to CH_b, with flag_CH_merge = 1
    CH_f broadcasts a Change_CHID to inform its current CMs to change their corresponding CHID to the ID of CH_b
    CH_f changes its state to CM and changes its corresponding CHID to the ID of CH_b
else
    CH_f sends back a ACKMerge message to CH_b, with flag_CH_merge = 0
    CH_f becomes the CH_merge
end if

For cluster head moving back CH_b, on receiving a ACKMerge message from CH_f:
if flag_CH_merge = 1 then
    CH_b becomes the CH_merge
else
    if flag_CH_merge = 0 then
        CH_b broadcasts a Change_CHID message to inform its current CMs to change their corresponding CHID to the ID of CH_f
        CH_b changes its state to CM and changes its corresponding CHID to the ID of CH_f
    end if
end if

For cluster member CM_i, on receiving a Change_CHID message from its current CH, CH_f or CH_b:
CM_i extracts the new CH identifier and changes its corresponding CHID
```

D. Analysis

In this subsection, we will present a general model to analyze the rationality of both the existing and our proposed merging schemes. Without loss of generality, we consider the cluster C_1 is merged with the cluster C_2, while the merging condition is already satisfied. We assume that the new CH after merging is one of the two original CHs (1 or 2), and it associates with all CMs in C_1 and C_2, and another original CH.

We denote the current time when cluster merging happens as t_0. $D^-(m, C_m)$ indicates the average link duration between CMs in cluster C_m and their CH m before t_0, where $m = 1$ or 2. Furthermore, we denote $D^+(n, C_m)$ as the average link duration between CMs in cluster C_m and the external CH n after merging, counting from the moment t_0, where $(m, n) \in \{1, 2\} \times \{1, 2\}$. The superscript "-" and "+" represents the time before and after t_0, respectively.

We denote $\widetilde{D}$ as the average CM duration of vehicles in these two clusters around the moment t_0. Without loss of generality, we consider that the vehicle 1 is the new CH after merging. The CMs in C_1 originally have the association with the vehicle 1 uninterruptedly. Their link duration is $D^-(1, C_1) + D^+(1, C_1)$ on average. The CMs in C_2 originally break their association with the vehicle 2 at t_0, and associate with the vehicle 1 afterward. Therefore these CMs have two periods of CM durations, which are $D^-(2, C_2)$ and $D^+(1, C_2)$ on average respectively. The vehicle 2 associates with the vehicle 1 only after t_0 and its mean CM duration is $D^+(1, C_2)$. Since the total number of CM/CH connections around t_0 for vehicles in these two cluster is $N_1 + 2N_2 + 1$, the average CM duration of them, denoted as $\widetilde{D}$ in such a case is $\frac{[D^-(1,C_1)+D^+(1,C_1)]N_1+D^-(2,C_2)N_2+D^+(1,C_2)N_2+D^+(1,C_2)}{N_1+2N_2+1}$. Following the same procedure, we can compute $\widetilde{D}$ in the case that the vehicle 2 is the new CH. Let δ be an binary variable to indicate whether node 1 or node 2 is the new CH, which is defined as

$$\delta = \begin{cases} 1, & \text{if vehicle 1 is the new CH} \\ 0, & \text{if vehicle 2 is the new CH.} \end{cases} \tag{5}$$

Then $\widetilde{D}(\delta)$ can be generalized as a function of the CH selection decision, which is

$$\widetilde{D}(\delta) = \frac{\alpha + \beta}{\gamma}, \tag{6}$$

where α and β are the sum of CM duration before and after t_0, and γ is the total number of CM/CH connections around t_0 for vehicles in these two clusters. They are:

$$\begin{aligned} \alpha &= N_1 D^-(1, C_1) + N_2 D^-(2, C_2), \\ \beta &= \delta N_1 D^+(1, C_1) + (1-\delta)(N_1+1) D^+(2, C_1) \\ &\quad + \delta(N_2+1) D^+(1, C_2) + (1-\delta) N_2 D^+(2, C_2), \\ \gamma &= N_1 + N_2 + (1-\delta) N_1 + \delta N_2 + 1. \end{aligned}$$

When $\widetilde{D}(1) - \widetilde{D}(0) > 0$ (or ≤ 0), the vehicle 1 (or the vehicle 2) should be selected as the new CH, as that leads to a larger mean CM duration. We will show in the following that, different cluster merging schemes follows such rationality under different assumptions.

Under the assumption that $D^-(m, C_m)$ and $D^+(n, C_m)$ are equal to a constant for $m \in \{1, 2\}$

and $n \in \{1, 2\}$, the sign of $\widetilde{D}(1) - \widetilde{D}(0)$ is identical with $N_1 - N_2$. This is the intuition of the "CM-based" cluster merging schemes, such as [5], [6], and [7].

Under the assumption that N_1 is similar to N_2, and $D^+(2, C_1)$ is similar to $D^+(1, C_2)$, the sign of $\widetilde{D}(1) - \widetilde{D}(0)$ is identical with $D^+(1, C_1) - D^+(2, C_2)$. This is the intuition of the "VMaSC-based" scheme in [8].

Our proposed LCM scheme is under the assumption that N_1 is similar to N_2, and the sign of $D^+(2, C_1) - D^+(1, C_1)$ is identical with $D^+(1, C_2) - D^+(2, C_2)$. The intuition is that there is an exact one-one matching between CHs and clusters on the similarity in mobility pattern. Without loss of generality, we consider the vehicle 1 is the CH f in the front, and the vehicle 2 is the CH b in the back. Then the sign of $\widetilde{D}(1) - \widetilde{D}(0)$ is identical with $D^+(f, C_f) - D^+(b, C_f)$. Notice that the leadership $L_{m,Cn}$ is positively correlated with $D^+(m, C_n)$. Then the sign of $\widetilde{D}(1) - \widetilde{D}(0)$ is identical with $L_{f,C_f} - L_{b,C_f}$ calculated at the front vehicle f. Compared with "CM-based" and "VMaSC-based" schemes, our scheme does not require the assumption of equality between CM durations, which is more flexible.

IV. Performance Evaluation

In this section, we provide some insights into three cluster merging schemes: the CM-based scheme; the VMaSC-based scheme; and the proposed leadership-based cluster merging (LCM) scheme, denoted as "Leadership-based". In the simulation, three cluster merging schemes are implemented on a cluster framework independently, and a detailed cluster performance comparison is presented.

All of the schemes are implemented on the Network Simulator NS2 [12]. Cluster Merge Interval (MI) is set to 2, 5, 10, 20, and 50 seconds respectively. Each simulation is repeated for 10 times and calculates the average value. The simulation configuration is described as follows.

A. Testing scenarios

In the simulation, four testing scenarios are generated by Simulation of Urban Mobility (SUMO) [13], aiming to observe the cluster performance under different traffic scenarios. In all scenarios, there are 200 vehicles: 100 from East to West, and 100 from West to East. Each moving direction has two lanes. The length of the road is set to 10 km, which is equally divided into 8 segments. Vehicles are generated with a certain traffic generation rate, 1500 vehicles per hour. The simulation runs for 600 seconds. The transmission range is *TR=300* meters.

These four testing scenarios consist of two relative static traffic models and two highly dynamic traffic models, which are named as scenarios A.1, A.2, B.1, and B.2 respectively. There is only one vehicle type in scenario A.1, and there are four vehicle types in scenario A.2. The maximal speed limit of the road is a constant in scenarios A.1 and A.2. In scenarios B.2 and B.2, there are four types of vehicles, and the maximal speed limit of each segment is different. The setting of vehicles in each scenario is shown in Tables I, II, III, and the setting of each scenario is shown in Table IV. The mobility pattern of vehicles in scenarios B.1 and B.2 is more unpredictable than that in scenario A.1 and A.2.

TABLE I
Vehicle setting for scenario A.1

Max speed	Acceleration	Deceleration	Speed deviation
20 m/s	2.0 m/s^2	6.5 m/s^2	0.1

TABLE II
Vehicle setting for scenario A.2

Type	Max speed	Acceleration	Deceleration	Speed deviation
1	20 m/s	2.9 m/s^2	7.5 m/s^2	0.7
2	20 m/s	2.9 m/s^2	7.5 m/s^2	0.3
3	20 m/s	2.0 m/s^2	6.5 m/s^2	0.1
4	20 m/s	1.5 m/s^2	5.5 m/s^2	0.3

TABLE III
Vehicle setting for scenario B.1 and B.2

Type	Max speed	Acceleration	Deceleration	Speed deviation
1	35 m/s	2.9 m/s^2	7.5 m/s^2	0.7
2	25 m/s	2.9 m/s^2	7.5 m/s^2	0.3
3	20 m/s	2.0 m/s^2	6.5 m/s^2	0.1
4	10 m/s	1.5 m/s^2	5.5 m/s^2	0.3

TABLE IV
Testing scenario settings

Scenario	Vehicle type	Lane segment Max speed (m/s)
A.1	Table I	8 segments (for each: 20)
A.2	Table II	8 segments (for each: 20)
B.1	Table III	8 segments (20,30,20,30,10,20,15,20)
B.2	Table III	8 segments (20,15,25,30,25,20,15,20)

The clustering process starts at time T_{start}, the time when all vehicles have entered the road. Vehicles are possible to establish CH/CM connections according to the clustering scheme. After the time T_{end}, all connections between CH/CM are automatically disconnected. T_{end} is the time which guarantees that $T_{end} - T_{start}$ is large enough, and most of vehicles are still on the road before T_{end}. In our simulation, we set $T_{start} = 350s$, and $T_{end} = 550s$.

In [14], according to the Canton of Zurich scenario, about 50% of vehicles have no more than 10 neighboring vehicles, and nearly 95% vehicles have less than 60 neighboring vehicles. Therefore, in this simulation, the maximum number of vehicles in each cluster is set to 10, where *MAX_CM* $= 10$. Other simulation parameters are illustrated in Table V.

TABLE V
DEFAULT SIMULATION PARAMETERS

Parameter	Value
Simulation time	200 s
T_{start}	350 s
T_{end}	550 s
Length of road	10 km
Number of vehicles	200
Transmission Range (TR)	300 m
Beacon Interval (BI)	1.0 s
Merge Interval (MI)	2, 5, 10, 20, 50 s
MAX_CM	10
T_W	2.0 s
$T_{collect}$	3.0 s
Δv_{th}	5.0 m/s
$\Delta\overline{V_{Cth}}$	10.0 m/s
T_{ACK}	2.0 s
Mobility model	Car-following model
Propagation model	Two-Ray Ground
MAC protocol	IEEE 802.11p
Frequency/Channel Bandwidth	5.9GHz/10MHz
Number of iterations	10

B. Performance metrics

We study the performance of clustering schemes from both macroscopic and microscopic aspects through the simulation. The macroscopic performance presents the overall cluster stability, and the microscopic performance shows vehicles' behaviors during the clustering procedure. The detail of the definition of performance in these two aspects is as follows:

1) Macroscopic performance:

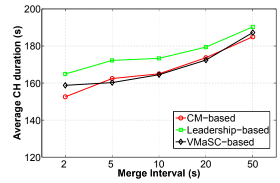
- *Cluster head duration* presents the cluster's lifetime. It is the average time interval from a vehicle becoming a CH to giving up its state. In general, a longer CH lifetime means a more stable cluster.
- *Cluster member duration* defines the average time interval from a node joining an existing cluster as a member in CM state to becoming another state.

2) Microscopic performance:

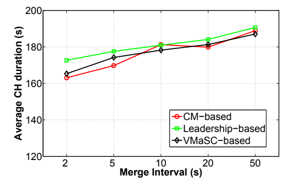
- *Average role change rate (per second)* presents the total number of vehicles' state changes in one second.
- *Cluster member disconnection rate (per second)* describes the total number of disconnections between CMs and their CHs per second.

C. Performance analysis

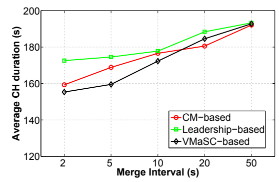
The macroscopic performance, i.e., the mean CH lifetime and mean CM lifetime, versus cluster Merge Interval (MI), is shown in Fig. 2 and Fig. 3. In the four testing scenarios, the maximal speed of lanes and vehicles are equal in the A series scenarios, while that is not a constant in the B series scenarios. Therefore, the cluster size is similar in the A series scenarios as there is no traffic jamming. Since the traffic mobility is quite unpredictable in the B series scenarios, the mean association duration between CMs and different neighboring CHs can be regarded as more or less similar. According



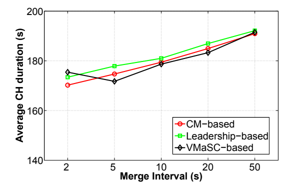
(a) Scenario A.1



(b) Scenario A.2



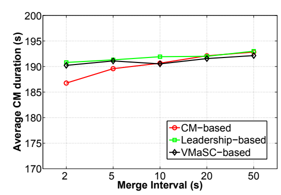
(c) Scenario B.1



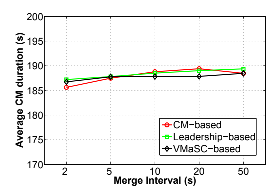
(d) Scenario B.2

Fig. 2. Impacts of MI on CH lifetime.

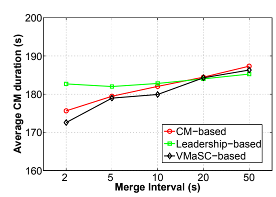
to our analysis in Section III-D, the CM-based scheme is more suitable in the A series scenarios, and the VMaSC-based scheme is more suitable for the B series scenarios, which is confirmed by the simulation results. When MI increases, the mobility pattern of the two clusters that pass the merging condition check becomes more and more similar. Therefore the CM-based scheme outperforms the VMaSC-based scheme when MI is sufficiently large. Comparing with the two benchmarks, the Leadership-based scheme overall achieves the best performance, in regardless of traffic scenarios and the setting of MI, as it has a better prediction on the stability between CMs and potential CHs.



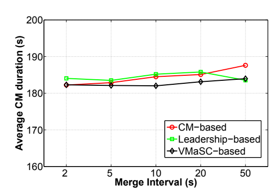
(a) Scenario A.1



(b) Scenario A.2



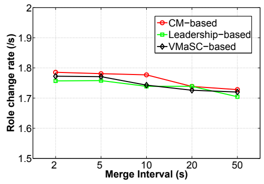
(c) Scenario B.1



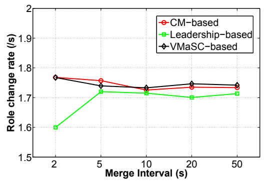
(d) Scenario B.2

Fig. 3. Impacts of MI on CM lifetime.

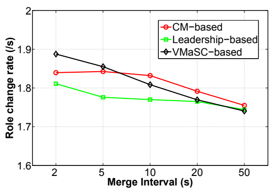
The microscopic performance, i.e. the role change rate, and the CM disconnection rate, is shown in Fig. 4 and Fig. 5. Since the Leadership-based scheme targets to ensure a stable cluster after cluster merging, it achieves the lowest role change rate and lowest CM disconnection rate by avoiding frequent cluster splitting and CM/CH de-association. Its advantage over VMaSc-based scheme shows that the stability between CMs and different CHs provide more information than the stability in the current cluster. The CM-based scheme, on the contrary, shows poor performance especially for the CM disconnection rate, as it does not take into consideration of the stability of the merged cluster.



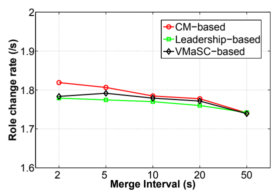
(a) Scenario A.1



(b) Scenario A.2

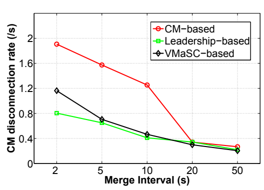


(c) Scenario B.1

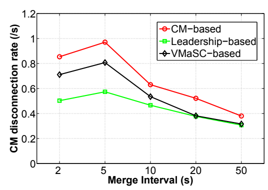


(d) Scenario B.2

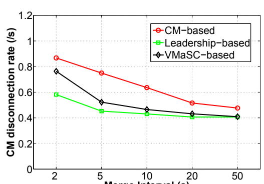
Fig. 4. Impacts of MI on the average role change rate.



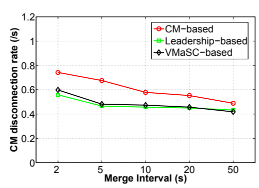
(a) Scenario A.1



(b) Scenario A.2



(c) Scenario B.1



(d) Scenario B.2

Fig. 5. Impacts of MI on CM disconnection rate.

We can observe that both macroscopic and microscopic performance are improved with the increase of MI. Nevertheless, it is not desirable to always utilize a large MI, as this may result in many small sized clusters, which degrades the efficiency of clustering in VANETs for data communication. The simulation results reveal that the proposed Leadership-based scheme is less sensitive to the change of MI. Therefore, LCM is able to ensure both good macroscopic and microscopic performance and high clustering efficiency at the same time.

V. Conclusion

In this paper, we investigate the impact of cluster merging component inside clustering for vehicles in VANETs. A Leadership-based Cluster Merging (LCM) scheme is proposed and is compared with other cluster merging schemes under the same framework. The LCM scheme assigns a vehicle which has better link stability with all members in the new cluster, to be the cluster head of the merged cluster. The intuition of LCM is to have a better perception of novel events and better cluster stability, so as to be compatible with popular applications such as platooning in vehicular networks. The simulation results conducted by NS2 and SUMO show that LCM achieves better cluster stability, compared with VMaSC-based, and CM-based schemes.

In the future work, we will investigate a more detailed definition of leadership in real traffic scenarios, and study the impact of cluster merging schemes for applications such as data dissemination and data aggregation.

References

[1] Y. Z. Hassnaa Moustafa, *Vehicular Networks: Techniques, Standards, and Applications*, 2nd ed. Auerbach Publications, 2009.

[2] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 584–616, Fourth 2011.

[3] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of network and computer applications*, vol. 37, pp. 380–392, 2014.

[4] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A Comparative Survey of VANET Clustering Techniques," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.

[5] Z. Y. Rawashdeh and S. M. Mahmud, "A novel algorithm to form stable clusters in vehicular ad hoc networks on highways," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 15, 2012.

[6] Z. Wang, L. Liu, M. Zhou, and N. Ansari, "A position-based clustering technique for ad hoc intervehicle communication," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 2, pp. 201–208, 2008.

[7] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by VANET," *Vehicular Communications*, vol. 2, no. 2, pp. 110 – 123, 2015.

[8] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2621–2636, April 2016.

[9] E. Souza, I. Nikolaidis, and P. Gburzynski, "A new aggregate local mobility clustering algorithm for vanets," in *2010 IEEE International Conference on Communications*, May 2010, pp. 1–5.

[10] H. R. Arkian, R. E. Atani, A. Pourkhalili, and S. Kamali, "Cluster-based traffic information generalization in Vehicular Ad-hoc Networks," *Vehicular Communications*, vol. 1, no. 4, pp. 197 – 207, 2014.

[11] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, "A Unified Framework of Clustering Approach in Vehicular Ad hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, 2017, doi:10.1109/TITS.2017.2727226, to be published.

[12] The Network Simulator - ns-2. [Online]. Available: http://isi.edu/nsnam/ns/

[13] Recent development and applications of SUMO - Simulation of Urban MObility. [Online]. Available: http://elib.dlr.de/80483/

[14] O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Worm Epidemics in Vehicular Networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2173–2187, Oct 2015.