This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A Unified Framework of Clustering Approach in Vehicular Ad Hoc Networks

Mengying Ren, Jun Zhang, Lyes Khoukhi, Houda Labiod, and Véronique Vèque, *Senior Member, IEEE*

***Abstract*— Effective clustering algorithms are indispensable in order to solve the scalability problem in vehicular ad hoc networks. Although current existing clustering algorithms show increased cluster stability under some certain traffic scenarios, it is still hard to address which clustering metric performs the best. In this paper, we propose a unified framework of clustering approach (UFC), composed of three important parts: 1) neighbor sampling; 2) backoff-based cluster head selection; and 3) backup cluster head based cluster maintenance. Three mobility-based clustering metrics, including vehicle relative position, relative velocity, and link lifetime, are considered in our approach under different traffic scenarios. Furthermore, a detailed analysis of UFC with parameters optimization is presented. Extensive comparison results among UFC, lowest-ID, and VMaSC algorithms demonstrate that our clustering approach performs high cluster stability, especially under high dynamic traffic scenarios.**

***Index Terms*— Vehicular ad hoc networks, clustering, 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], [2]. 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. In [3], a brief summarization of ITS applications and their requirements has been introduced.

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; thus, scalability problem becomes a big challenge in VANETs. To solve the network scalability problem, a hierarchical network structure has been proposed instead of a flat network structure, where vehicles are divided into several virtual groups, called clusters [4]. Vehicles in the same cluster can directly communicate with their leader vehicle via an intra-cluster communication, while inter-cluster communication can be achieved by leader vehicles.

Clustering algorithms have been originally proposed for Mobile Ad-hoc Networks (MANETs), such as Lowest-ID clustering algorithm [5], weighted clustering algorithms WCA [6], and highest connectivity based clustering algorithm HCC [7], which have been proven to effectively solve the network scalability problem. In recent years, some clustering algorithms, designed for ad hoc networks, were implemented in VANETs, however, due to the special characteristics of VANETs introduced in [1], such as predictable mobility and predefined road topology, some research works (e.g., [8]–[11]) find that mobility-based clustering metrics are more effective in order to improve vehicles' link connection quality in a cluster.

In a clustering algorithm, vehicles are usually located inside clusters; each cluster has at least one cluster head, and many cluster members. Generally, cluster stability is the average link connection time of a cluster; higher cluster stability indicates a better clustering algorithm [12]. Even though vehicle disconnections cannot be avoided in clustering, a suitable cluster maintenance method can assist to not only reduce vehicle disconnection frequency but also minimize the vehicle's re-affiliation delay.

In this paper, a unified framework of clustering (UFC) approach is proposed, aiming at improving cluster performance in terms of cluster formation efficiency, cluster changing rate, and cluster stability. In UFC, we use different mobility-based metrics and evaluate how these clustering metrics affect cluster performance. We also provide a detailed simulation context and a general discussion of the comparison between our clustering algorithm and a benchmark scheme. The main contributions of the paper are listed as follows:

- We propose a neighbor sampling (NS) scheme to filter out the unstable neighbors and to select the stable neighbor set. We assume that only vehicles in the stable neighbor set have the possibility to build connections with the cluster head.
- We propose a Backoff-based cluster head selection (BCS) scheme in order to reduce clustering management overhead. Each vehicle makes its own cluster head decision in a distributed manner by calculating its own backoff timer. The vehicle, with higher probability of being a cluster head, will set a smaller backoff timer. In our work, three mobility-based clustering metrics are implemented to compute the backoff timer, including link lifetime, relative speed, and relative distance.

Manuscript received December 12, 2016; revised April 18, 2017 and June 26, 2017; accepted July 2, 2017. The Associate Editor for this paper was L. Yang. *(Corresponding author: Mengying Ren.)*

M. Ren and L. Khoukhi are with the Environment and Autonomous Networks Laboratory, University of Technology of Troyes, 10010 Troyes, France (e-mail: mengying.ren@utt.fr; lyes.khoukhi@utt.fr).

J. Zhang and H. Labiod are with the INFRES, Télécom ParisTech, University of Paris-Saclay, 75013 Paris, France (e-mail: jun.zhang@telecom-paristech.fr; houda.labiod@telecom-paristech.fr).

V. Vèque is with the Laboratory of Signals and Systems, University Paris-Sud, University of Paris-Saclay, 91192 Gif-sur-Yvette, France (e-mail: veronique.veque@u-psud.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2727226

- We propose a BackUp Cluster Head based cluster maintenance (BUCH) scheme, in order to mitigate the influence of cluster intermittent. Since clusters may be overlapped, each vehicle may hear more than one cluster head. Thus, a Backup cluster head list can be easily created in each vehicle to cache several backup cluster heads, in the case of loss of connection to the main cluster head.
- A detailed analysis of the proposed clustering scheme is presented by adjusting the corresponding metrics under different typical traffic scenarios, including both relative stable and high dynamic traffic scenarios.
- Clustering performance metrics are categorized into macroscopic and microscopic levels. A detailed comparison between UFC scheme, Lowest-ID scheme, and VMaSC scheme is presented, in terms of both macroscopic and microscopic performance metrics.

The rest of the paper is organized as follows: Section II discusses the previous VANETs clustering works. Section III presents our proposed clustering algorithm from the aspects of cluster head selection, cluster formation and cluster maintenance. Section IV presents our simulation environment and analyzes UFC scheme from the view of parameter optimization, then, a fair comparison is given in this section. Section V concludes the paper and briefly presents our future work.

II. Related Work

A cluster is a virtual group which contains at least one cluster head (CH) and multiple cluster members (CM). Several research works have been presented in VANETs dealing with vehicle mobility metrics and cluster formation mechanisms. In this section, some recent clustering algorithms are reviewed.

A simple and direct way to choose a CH is to select the first vehicle moving in a certain direction. Cluster, like platooning in CONVOY [13] under highway scenario, selects the first vehicle as a CH. Vehicles, within the predefined maximum length to CH are grouped together, construct a multi-hop cluster. MC-DRIVE [14] proposed a direction-based clustering algorithm for intersection area. The first vehicle moving in a certain direction is selected as CH and clusters are formed in one-hop based on CHs' transmission range. Another clustering algorithm proposed in [15] and [16] aim to select the most front vehicle as the temporary CH, and then the temporary CH assigns the one that is mostly closest to the center of the cluster as the CH. In this work, the cluster members join the cluster in a one-by-one manner. However, these mechanisms are only suitable for simple road topology, like straight highways and intersections. In [17], a central vehicle, denoted as C-Vehicle, is selected for resource reusing.

Instead of simply choosing the first vehicle as a CH, most clustering mechanisms prefer calculating the stability of a node to its surroundings. MOBIC [18] was the first work proposing the relative mobility concept (originally proposed for MANETs). Each node calculates its relative mobility to all of its neighbors based on Received Signal Strength (RSS). The node with the lowest aggregate mobility is chosen as the CH. Similar to MOBIC, ALM [8] also chooses a node with less variance with respect to its surroundings as a CH. Instead of using RSS parameter, ALM calculates the relative distance between two nodes. APROVE [19] is based on a data clustering technique, named Affinity Propagation [20], where each vehicle sends hello messages periodically, including availability and responsibility messages. Vehicles' relative distance, current position and position prediction are used in APROVE. The vehicle with highest sum of availability and responsibility value is selected as a CH. In addition, SCRP [21], [22] is a cluster-based routing protocol based on Dominating Set (DS), which attempts to select a small number of mobile nodes as dominating nodes to form a stable backbone in a network. A graph theory algorithm has also been applied in a resource-sharing scheme proposed in [23].

Besides the clustering algorithms mentioned above, some algorithms select CH based on a sum of weighted values, named Weighted-based Clustering Algorithms (WCA). Mohammad and Michele [24] proposed a lane-based clustering algorithm using vehicles' relative speed, relative position and traffic flow. Each lane can be attributed with a certain weight value based on the traffic flow. VWCA [25] calculates the weighted clustering value based on several metrics values like vehicle distrust value, entropy value, number of neighbors and relative position. The vehicle with the minimum weighted sum value in the neighborhood is selected as a CH. To improve the network connectivity, VWCA introduced an adaptive allocation of transmission range algorithm (AATR) based on the intra-cluster communication standard, Dedicated Short-range Communication (DSRC) standard [26]. Vehicle can change the transmission range dynamically from 100m to 1000m according to vehicle density. Another weighted clustering mechanism AMACAD [11] was proposed based on vehicle's final destination, obtained by navigation system. In AMACAD, vehicles with similar destinations have high possibility to stay in the same cluster. The weighted sum is calculated based on vehicles' relative destinations, final destinations, relative speed and current position.

In recent years, researchers focus on building up multi-hop clusters to improve the cluster stability. The paper [27] proposed a N-hop clustering. N-hop relative mobility is based on the ratio of packet delivery delay of two consecutive packets. PPC [9] is also a multi-hop clustering mechanism which is based on vehicles' speed variations and predicted traveling time. Vehicle's relative stability value "Eligibility" decreases exponentially with the increased speed deviation. The VMaSC algorithm proposed in [28] claimed that it was the first multi-hop clustering scheme which is simulated under realistic traffic scenario. It aims to provide more stable clusters and to reduce the number of CHs in the network. The CH election is based on the calculated relative mobility with respect to its neighbors. The performance was compared with the N-hop scheme in [27] for 1-hop, 2-hops, 3-hops, and VMaSC shows a better clustering performance, especially when the cluster size is set to 3-hop. Similar to [28], Amoozadeh *et al.* [29] proposed a platoon management method, which can also be seen as a multi-hop cluster with a leader moving ahead. The platoon size is less than the predetermined optimal platoon size. DMCNF algorithm, introduced in [30], is another multi-hop based clustering algorithm proposing a neighborhood

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

following strategy, in which, each vehicle finds a stable target to follow. The vehicle only needs to know the information of its local one-hop neighbors; this can help to reduce the packet loss rate in the network.

According to the recent related work that we have summarized above, all of them are beacon-based. Mobile vehicles exchange their basic information via beacon messages periodically, calculate their cluster head metric values, and exchange these values with respect to all of their neighbors. Thus, such this kind of metric exchanges causes a higher cluster management overhead, a waste of network resources, and more packet collisions. The proposed Backoff-based CH Selection (BCS) scheme in the algorithm, in which, every vehicle set a backoff timer to compete as CH, can reduce both the number of control packets and packet collisions. Moreover, another proposed Neighbor Sampling (NS) scheme can pick out stable neighbors. Vehicles only need to calculate cluster head metrics with respect to their stable neighbors, instead of all of their neighbors. Therefore, redundant packet transmissions can be avoided.

Furthermore, not all of the recent clustering algorithms have well defined the vehicle re-clustering process (e.g. [11], [29], [30]), even though it plays a vital role in increasing the cluster stability. The main objective of vehicle re-clustering is allowing the disconnected vehicles to reconnect to a new cluster head successfully with the minimum delay. In this study, a BackUp Cluster Head based cluster maintenance (BUCH) approach is introduced, in order to reduce vehicle re-clustering delay and to improve the probability of successful re-clustering.

III. Proposed Approach

In this paper, our work focuses on proposing a unified framework of clustering algorithm, denoted as UFC. UFC is only based on V2V communication type. All vehicles are assumed to be equipped with a GPS system which provides vehicle's basic information, including vehicle's current location, 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 their one-hop neighbors via *Beacon* messages at every Beacon Interval (BI). Information contained in *Beacon* message includes vehicle's identifier *ID*, vehicle's current state *R*, cluster identifier $ID\_cluster$, current position (x, y), current velocity v, and moving direction *Dir*.

Fig. 1 presents an example of cluster network topology. Two clusters are presented on the road, with single cluster head in each cluster. Clustering procedure will be described in the following sections: cluster head selection, cluster formation, and cluster maintenance. The notations used are presented in Table I.

A. Vehicle States

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

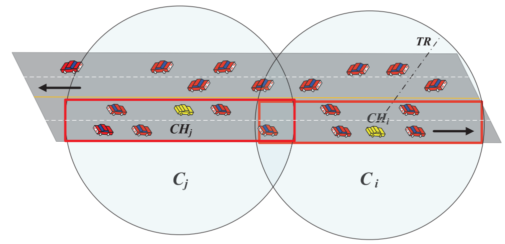

Fig. 1. Clusters C_i and C_j.

TABLE I
Notations

Notation	Description
TR	Transmission Range
BI	Beacon Interval
$T_{collect}$	System timer for *Beacon* exchange and collection
T_W	Backoff timer for CH selection
Δv_{th}	Speed difference threshold between two vehicles
MAX_CM	Maximum number of CMs in a cluster
V_i	Vehicle i,; i is the ID of vehicle
N_i	Number of stable neighbors of V_i
SN_i	Stable neighbor set of V_i
$SN_i(j)$	Neighbor list entry of SN_i in V_i
CH	Cluster head
CM	Cluster member
CCM	Candidate cluster member
CML_i	CM List of CH_i
LLT_{ij}	Link lifetime between V_i and V_j
$\overline{LLT_i}$	Average LLT between V_i and its stable neighbors
Δv_{ij}	Speed difference between V_i and V_j
ΔD_{ij}	Distance between V_i and V_j
C_i	Cluster i, i is the ID of CH
$\Delta\overline{V_{Cij}}$	Speed difference between C_i and C_j
$\Delta\overline{V_{Cth}}$	Speed difference threshold between two clusters
BCH	Backup cluster head
T_{cm}	Timer for CM to hear *Beacon* from its CH
T_{ch}	Timer for CH to hear *Beacon* from its CM
$BCHL_i$	Backup CH List in vehicle V_i
$BCHL_i(j)$	List entry of BCH_j, stored in BCH_i of V_i
MAX_BCH	Maximum number of BCHs in the BCHL
T_{bch}	Timer for CCM and CM to build BCHL

- Undecided Node (UN): Initial state of all vehicles, meaning that the vehicle does not belong to any cluster.
- Cluster Head (CH): The leader of the cluster, which can communicate with all of its members. Each cluster has only one CH.
- Cluster Member (CM): The vehicle which can directly be attached to an existing CH.
- Candidate Cluster Member (CCM): The vehicle which intends to be a CM of an existing cluster, but has not yet received a confirmation message.

The transition between two of these states are triggered by different events, presented through a state machine in Fig. 2. The state transition process will be described in the following subsections, through the presentation of the main procedures of our UFC algorithm, NS, BCS, as well as BUCH.

B. Neighbor Sampling (NS)

At the beginning of the clustering procedure, each node is in an initial state, indicated as UN node. The system starts a timer, called $T_{collect}$, during which vehicles exchange and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

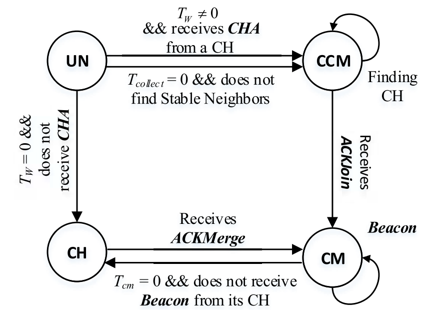

Fig. 2. State transition.

collect *Beacons* to discover their one-hop vicinities, called Potential Neighbor set (*PN*). However, not all vehicles in *PN* are ideally to be clustered. In order to reduce the redundant computation and useless message exchanges, the Neighbor Sampling process selects a set of stable neighbors from *PN*, denoted as Stable Neighbor set (*SN*), where $SN \subset PN$. The vehicle in SN is defined as the neighboring vehicle that presents a similar mobility pattern: (1) moving in the same direction; and (2) the speed difference Δv is smaller than the predetermined threshold Δv_{th}.

Every vehicle V_i maintains a set of stable neighbors SN_i, containing several entries, indicated as $SN_i(j)$. Every neighbor list entry $SN_i(j)$ contains the following information:

Term	Description
(x_j, y_j)	position of vehicle V_j
v_j	velocity of vehicle V_j
LLT_{ij}	Link lifetime between vehicle V_i and V_j
Δv_{ij}	relative speed between vehicle V_i and V_j
ΔD_{ij}	relative distance between vehicle V_i and V_j

Algorithm 1 Neighbor Sampling

while $T_{collect} > 0$ **do**
  if UN_i receives *Beacon* from UN_j **then**
    if $Dir(i) == Dir(j)$ && $\Delta v_{ij} \leq \Delta v_{th}$ **then**
      UN_i calculates LLT_{ij}
      if $UN_j \in SN_i$ **then**
        UN_i updates $SN_i(j)$
      else
        UN_i adds the entry $SN_i(j)$ to SN_i
        $N_i \leftarrow N_i + 1$
      end if
    end if
  end if
end while

It happens sometimes that some UN vehicles could not find any stable neighbors during $T_{collect}$ time period, where $SN_i = \emptyset$. Under these circumstances, such UN vehicles will directly change state to CCM, instead of participating in the Backoff-based CH Selection (BCS) process. Meanwhile, they will create a Backup CH List (BCHL) and try to find a suitable CH to follow, which will be explained in Algorithm 5 in Section III-E.

C. Backoff-Based CH Selection (BCS)

As we have mentioned in Section II, almost all cluster head selection schemes are based on mobility metric exchanging and broadcasting. This kind of scheme increases the packet collision probability, as well as the management overhead. Instead of broadcasting cluster metrics, our proposed CH selection scheme allows each vehicle to set up its own backoff timer, T_W, in a distributed manner, waiting to broadcast a CH Announcement message *CHA*. The first vehicles broadcasting *CHA* messages among their neighbors will become initial CHs.

This work proposes two methods for CH selection. The first method is a metric-based method, including the following metrics: average Link Lifetime $\overline{LLT_i}$, average relative distance $\overline{\Delta D_i}$, and average relative speed $\overline{\Delta v_i}$. The second one is a random-based method. The details are described as follows.

1) Metric-Based CH Selection Method: Link Lifetime (LLT), also called Link Expiration Time (LET) [31], describes the link sustainability, representing the duration of time when two vehicles remain connected. The work in [32] gives the definition of LLT, shown in Eq. (1), when two vehicles are moving in the same or opposite directions. Eq. (1) defines LLT calculation. Although vehicle position should be represented by x-coordinate and y-coordinate, this study assumes the trajectory of all vehicular nodes to be a straight line, as the lane width is small. Thus, the y-coordinate can be ignored. We denote the positions of V_i and V_j by x_i and x_j, respectively.

$$LLT_{ij} = \frac{-\Delta v_{ij} * \Delta D_{ij} + |\Delta v_{ij}| * TR}{(\Delta v_{ij})^2}. \quad (1)$$

Note that *TR* is the transmission range of a vehicle.

During $T_{collect}$ time period, for each received *Beacon* message from vehicle V_j, where $V_j \in SN_i$, vehicle V_i calculates the metric, LLT_{ij}, ΔD_{ij}, or Δv_{ij}, where $\Delta D_{ij} = x_i - x_j$, and $\Delta v_{ij} = v_i - v_j$. Then, vehicle V_i records the metric in its stable neighbor set SN_i.

When $T_{collect}$ is expired, V_i calculates the clustering metric $\overline{LLT_i}$, $\overline{\Delta D_i}$ or $\overline{\Delta v_i}$, according to Eq. (2), (3), or (4), respectively. The backoff time is calculated as in Eq. (5).

$$\overline{LLT_i} = \frac{\sum_{V_j \in SN_i} LLT_{ij}}{N_i} \quad (2)$$

$$\overline{\Delta D_i} = \frac{\sum_{V_j \in SN_i} |\Delta D_{ij}|}{N_i} \quad (3)$$

$$\overline{\Delta v_i} = \frac{\sum_{V_j \in SN_i} |\Delta v_{ij}|}{N_i}, \quad (4)$$

where N_i is the number of vehicles in SN_i of vehicle i.

$$T_{Wi} = T_{min} + (T_{max} - T_{min}) * M_i + \theta, \quad (5)$$

$$M_i = \begin{cases} 1 - \overline{LLT_i}/\Delta LLT_{max} \\ \overline{\Delta D_i}/\Delta D_{max} \\ \overline{\Delta v_i}/\Delta v_{max} \end{cases} \quad (6)$$

where M denotes the metric, T_{min} and T_{max} are set to 0s and 2s, respectively, LLT_{max} is given as the same time as the simulation time T_{sim}, ΔD_{max} is given as the same as *TR*, Δv_{max} is set to the same value as Δv_{th}, and $\theta \sim U(0, 0.1)$ follows a uniform distribution. Here, θ is added to the end of the equation to avoid the same T_W.

2) Random-Based CH Selection Method: Different from metric-based method, when $T_{colloct}$ expires, each vehicle chooses a random backoff time according to Eq. (7), where T_i follows a uniform distribution $T \sim U(0, 2.0)$, and $\theta \sim U(0, 0.1)$.

$$T_{Wi} = T_i + \theta_i \tag{7}$$

During T_{Wi}, if vehicle i receives a *CHA* message, it gives up CH competition process, cancels T_{Wi}, and changes its state from UN to CCM; otherwise, when T_{Wi} expires, vehicle i changes its state from UN to CH and broadcasts a *CHA* message to inform its vicinities of this state transition. The CH election process is described in Algorithm 2.

Algorithm 2 Contention-Based CH Election

When $T_{collect} == 0$
  UN_i calculates $\overline{M_i}$
  UN_i calculates T_{Wi} and starts timer T_{Wi}
  $flag\_T_W \leftarrow 1$
  while $T_{Wi} > 0$ **do**
    if UN_i receives A from CH_j **then**
      $flag\_T_W \leftarrow 0$
      $UN_i \rightarrow CCM_i$
      goto *Cluster formation*
    end if
  end while
  if $flag\_T_W == 1$ **then**
    $UN_i \rightarrow CH_i$
    CH_i broadcasts *CHA*
  end if

A *CHA* message should contain the following information:

- Message type, denoted by *CHA*;
- Vehicle ID: denoted by *ID*;
- Vehicle location: denoted by (x_i, y_i);
- Stable neighbor list: denoted by SN_i;
- Number of stable neighbors: denoted by N_i.

D. Cluster Formation

When a vehicle UN_j receives *CHA* message from CH_i, it immediately changes its state to a candidate cluster member CCM_j and sends a $ReqJoin$ message to CH_i. Meanwhile, CCM_j sets a timer T_{ack}, waiting for a confirmation notification from CH_i. The cluster formation process is described in Algorithm 3. After CH_i receives the *ReqJoin* message from CCM_j, it firstly checks the total number of existing CMs in the cluster. If the number of existing CMs is less than the cluster capacity, denoted as MAX_CM, CH_i adds CCM_j to its cluster member list CML_i and sends back a *ACKJoin* message; otherwise, CH_i ignores the $ReqJoin$ message. If CCM_j receives *ACKJoin* before T_{ack} expires, CCM_j changes its state to CM_j; otherwise, CCM_j tries to join another cluster, which will be described in the next section.

Algorithm 3 Cluster Formation

if UN_j receives *CHA* **then**
  $UN_j \rightarrow CCM_j$
  CCM_j sends *ReqJoin* to CH_i
  CCM_j starts timer T_{ack}
  goto *Waiting confirmation*
end if
if CH_i receives *ReqJoin* from CCM_j **then**
  if $N_i \leq MAX\_CM$ **then**
    $CML_i(j) \leftarrow CCM_j$
    sends *ACKJoin* to CCM_j
  end if
end if

Algorithm 4 Waiting Confirmation

while $T_{ack} \geq 0$ **do**
  if V_i receives *ACKJoin* from its CH_j **then**
    $V_i \rightarrow CM_i$
  end if
end while
if V_i does not receive *ACKJoin* **then**
  goto *BCH election*
end if

E. Backup CH Based Vehicle Re-Clustering

Due to the high dynamic nature of VANETs, vehicles keep joining and leaving clusters frequently. Indispensable vehicle re-clustering process guarantees that a CM can find a proper cluster to follow as long as it loses the contact with its current CH. However, a big delay in re-clustering process may lead to serious consequences, especially when delay-sensitive applications are implemented. To solve this problem, a caching cluster head scheme is proposed, aiming to reduce the vehicle re-clustering delay. A Backup CH List (*BCHL*) is created and updated in every CCM and CM node. Every time the vehicle loses the contact with its current CH, it starts to find the most qualified backup CH to follow. The backup CH based cluster maintenance procedure is introduced in the following parts.

1) Backup CH List (BCHL) Creation: Following the above description, every CCM node, for example CCM_i, which has not been clustered will set a timer T_{bch}, during which, CCM_i may hear *Beacon* messages from one or more than one CHs. For each detected CH_j, CCM_i only selects the qualified CHs and records their link lifetime LLT_{ij} and their relative speed Δv_{ij} in its Backup CH List $BCHL_i$ (Line 3-5 in Algorithm 5). The qualified CH should meet the following criteria: (1) moving in the same direction as CCM_i; (2) $\Delta v_{ij} \leq \Delta v_{th}$.

2) Backup CH List (BCHL) Ranking: In the $BCHL_i$ of vehicle V_i, BCHs are ordered according to their priorities: 1) BCH which has longer link lifetime LLT has higher priority; 2) if all *LLT* in the $BCHL_i$ are equal, BCH with

Algorithm 5 Backup CH List BCHL Creation

CCM_i starts timer T_{bch}
while $T_{bch} \geq 0$ **do**
  if CCM_i receives a *Beacon* from CH_j **then**
    if $Dir_j == Dir_i$ && $\Delta v_{ij} \leq \Delta v_{th}$ && $Size(BCHL_i) \leq MAX\_BCH$ **then**
      $BCHL_i(j) \leftarrow (CH_j, LLT_{ij}, \Delta v_{ij})$
      $BCHL_i$ *ranking*
    end if
  end if
end while

less Δv will be given a higher priority. Then, the vehicle V_i chooses the BCH, which has the highest priority, and sends a *ReqJoin* message as long as V_i loses the connection with its current CH. Algorithm 6 presents how a vehicle reconnects to a new CH.

If V_i fails to join the new cluster, it sends a *ReqJoin* message to another BCH, which has the second highest priority on the $BCHL_i$, and waits for the confirmation. The vehicle V_i repeats this process until it successfully reconnects to a new CH. Apparently, when the size of the backup CH list *BCHL* increases, the re-clustering delay increases. Thus, the size of *BCHL* should be limited by the predetermined value MAX_BCH, which will be discussed in Section IV.

Algorithm 6 BCH Selection

V_i checks $BCHL_i$
if $BCHL_i == \emptyset$ **then**
  if V_i is a CCM node **then**
    CCM_i restarts T_{bch}
  else
    if V_i is a CM node **then**
      $CM_i \rightarrow CH_i$
    end if
  end if
else
  if $Size(BCHL_i) \geq 1$ **then**
    V_i chooses CH with the highest priority
    V_i sends *ReqJoin* and starts T_{ack}
    goto ***Waiting confirmation***
  end if
end if

F. Cluster Maintenance

Clustering maintenance process in UFC can be divided into the following parts: *CML* updating, *BCHL* updating, leaving a cluster, and cluster merging.

1) CML Updating: Each CH maintains a dynamic CML. For each beacon interval, if CH hears a *Beacon* from one of its CMs, it updates the information in its *CML*. Each CH starts a control timer T_{ch} periodically, in order to monitor the *Beacons* of its CMs. Otherwise (i.e., if CH does not receive *Beacon* message from one of its CM during T_{ch} time period), CH removes this CM from its *CML*.

TABLE II
LIST OF IMPORTANT MESSAGES

Message	Contents
Beacon	$< T, ID, R, IDcluster, L, v, Dir >$
CHA	$< T, ID, R, IDcluster, L, v, Dir, SN, N, D >$
ReqJoin	$< T, ID, R, IDcluster, L, v, Dir >$
ACKJoin	$< T, ID, R, IDcluster, L, v, Dir, SN >$
ReqMerge	$< T, ID, R, IDcluster, L, v, Dir, N, CML >$
ACKMerge	$< T, ID, R, IDcluster, L, v, Dir, CML >$

2) BCHL Updating: The backup CH list (*BCHL*) is stored in a CCM/CM vehicle. When a vehicle hears a *Beacon* from a qualified BCH, as we have defined above, it updates the entry of *BCHL*, as well as records the BCHs. Similar to T_{ch}, timer T_{bch} is used in CCM/CM node, in order to monitor the *BCHL*. If a CCM/CM vehicle discovers that it could not hear an existing BCH any more during T_{bch}, it removes this BCH from its *BCHL*. The *BCHL* is reordered as long as the entry is updated.

3) Leaving a Cluster: Each CM sets a timer T_{cm} to monitor its connection with CH. If CM does not receive *Beacon* from its CH during T_{cm}, it considers itself out of the communication range of its current CH. Then, CM selects a BCH from its *BCHL* and directly sends *ReqJoin* to this BCH. If *BCHL*$== \emptyset$, the CM claims itself as a new CH, and tries to create a new cluster, as described in Algorithm 6.

4) Cluster Merging: When two neighboring CHs, CH_i and CH_j, are moving in the same direction within the transmission range of each other, the cluster merging procedure will be triggered. Instead of starting merging immediately, we introduce a merge interval, called T_{merge}, to defer cluster merging. Cluster merging process starts only if two CHs can contact with each other consecutively during T_{merge}, and the average speed difference between two clusters $\Delta\overline{V_{Cij}}$, shown in Eq. (8), is smaller than the predefined threshold $\Delta\overline{V_{Cth}}$. The CH vehicle with less CMs will give up the leadership and another CH becomes the CH of the merged cluster, called CH_{merge}. CMs in the dismissed cluster will be automatically included in the merged cluster.

$$\Delta\overline{V_{Cij}} = |\overline{V_{Ci}} - \overline{V_{Cj}}| \tag{8}$$

$$\overline{V_{Ci}} = \frac{\sum\limits_{V_k \in CML_i} |\Delta v_{ik}|}{N_i}, \tag{9}$$

where $\overline{V_{Ci}}$ is the average speed difference of cluster C_i, $|\Delta v_{ik}|$ is the speed difference between CH_i and CM_k, and N_i is the number of CMs in the cluster C_i.

G. Main Messages

Table II presents the important messages transmitted during the clustering procedures. The essential contents included in these messages are described, where T represents the message type, R is the state of the vehicle, L indicates the location, v is the speed and Dir is the moving direction. Table III introduces the message dissemination types.

TABLE III
LIST OF IMPORTANT MESSAGES

Message	Source	Dissemination type
Beacon	all vehicles	broadcast
CHA	CH	broadcast
ReqJoin	CCM or CM	unicast towards CH
ACKJoin	CH	unicast towards CCM or CM
ReqMerge	CH	unicast towards CH
ACKMerge	CH	unicast towards CH

TABLE IV
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

IV. PERFORMANCE EVALUATION

In this section, we provide some insights into the operation of our proposed UFC approach. A detailed analysis of UFC approach will be presented by optimizing different parameters under various scenarios in the first part of the simulation. In the second part, we compare the performance of UFC to the simplest clustering algorithm Lowest-ID [5], as well as to the latest and most cited clustering algorithm VMaSC [28]. Since both the proposed algorithms and Lowest-ID are based on one-hop cluster, the one-hop VMaSC is implemented in our simulation. All of the schemes are implemented on NS2 [33]. The simulation configuration is described as follows.

A. Testing Scenarios

We consider four testing scenarios, which are all generated by Simulation of Urban MObility (SUMO) [34]. As far as we know, this is the first article in this field which evaluates the proposed algorithm in various traffic scenarios and provides the details of simulation setting. In all scenarios, there are 200 vehicles: 100 from east to west, and 100 from west to east. In each moving direction, there are two lanes. The length of the road is set to 10 km, which is equally divided into 8 segments. The traffic flow rate is set to 1500 vehicles per hour. The simulation runs for 600 seconds. The transmission range (TR) is 300 meters.

These four testing scenarios consist of two relative stable traffic scenarios and two highly dynamic traffic scenarios, 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 acceleration rate and deceleration rate are set according to the default values in SUMO. 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 IV, V, VI, and the setting of each scenario is shown in Table VII. The mobility pattern of vehicles in scenarios B.1 and B.2 is more unpredictable than that in scenarios A.1 and A.2.

For each scenario, the simulation runs for 600s. The clustering process starts at time T_{start}, the time when all vehicles have entered the road. Vehicles 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 $T_{start} = 300s$, and $T_{end} = 500s$. General simulation parameters are illustrated in Table VIII.

TABLE V
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 VI
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 VII
TESTING SCENARIO SETTINGS

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

TABLE VIII
DEFAULT SIMULATION PARAMETERS

Parameter	Value
Simulation time	200 s
T_{start}	300 s
T_{end}	500 s
Length of road	10 km
Number of vehicles	200
Transmission Range (TR)	300 m
Beacon Interval (BI)	1.0 s
Beacon size	66 bytes
MAX_CM	10
MAX_BCH	2
$T_{collect}$	3.0 s
T_{bch}	5.0 s
T_{ch}	5.0 s
T_{cm}	5.0 s
Δv_{th}	5.0 m/s
$\Delta \overline{V_{Cth}}$	10.0 m/s
T_{ack}	2.0 s
T_{merge}	5.0 s
Mobility model	Car-following model
Number of iterations	10
Propagation model	Two-Ray Ground
Frequency/Channel Bandwidth	5.9GHz/10MHz
MAC protocol	IEEE 802.11p

B. Performance Metrics

Cluster stability could be defined from various aspects according to the implemented upper layer applications. In our simulation, we try to provide a detailed analysis of cluster performance from both macroscopic and microscopic levels, listed as follows. Macroscopic performance presents the overall cluster stability on the road. Microscopic performance shows vehicles' behaviors during the clustering procedure.

1) Macroscopic Performance:

- *Cluster head duration* presents the cluster's lifetime. It is the average time from a vehicle becoming a CH to giving up its state.
- *Cluster member duration* defines the average time from a node taking a CM state until changing to another state.
- *Number of clusters* defines how many clusters have been formed during the simulation period. A single CH without CMs also represents an independent cluster. Generally, an efficient clustering scheme prefers less formed clusters and more CMs in a single cluster.
- *Clustering efficiency* is defined as the percentage of vehicles participating in clustering procedure during the simulation. A higher clustering efficiency means a better clustering performance.

2) Microscopic Performance:

- *Number of initial CHs* is the number of vehicles that are elected in the beginning of clustering procedure.
- *CM disconnection rate (per second)* illustrates the total number of link disconnections between CMs and their current CHs per unit time.
- *Average role change rate* presents the total number of state changes per second during the clustering procedure.
- *CM re-clustering delay* is defined as the time interval of a CM from losing connection to successfully joining another cluster.
- *CM re-clustering success ratio* is defined as the percentage of successful CM re-connections after disconnections. A higher CM re-clustering success ratio guarantees the stability of the cluster.

C. Performance Optimization of UFC

In Section III, a new clustering approach has been introduced from the aspects of CH selection, cluster formation, and cluster maintenance. In this section, we will present how each method, proposed in UFC, influences the clustering performance.

1) Backoff Timer T_W Calculation: A Backoff-based CH selection method BCS has been presented in Section 2. Two potential methods for backoff time calculation are mentioned: a metric-based method and a random-based method. Three metrics have been proposed in the metric-based method, including link lifetime, relative distance, and relative speed. All vehicles calculate their individual T_W as soon as a system timer $T_{collect}$ expires. We compare these three metric-based methods and random-based method under the same context and analyze how T_W calculation method affects the cluster performance.

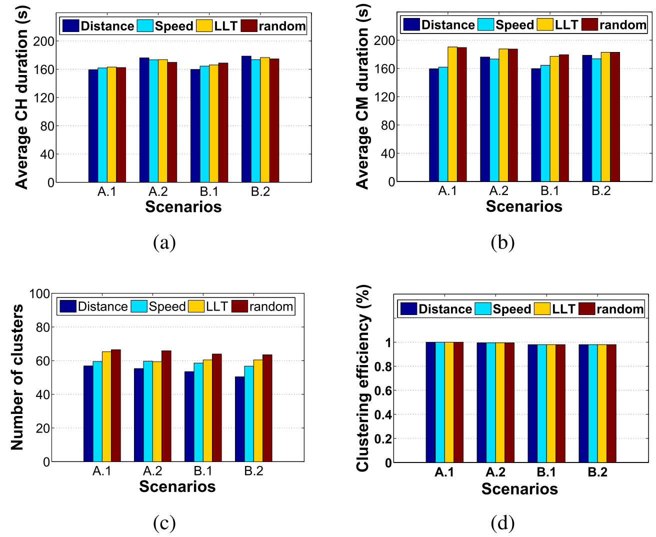

Fig. 3. Impact of T_W calculation method on the macroscopic performance of UFC. (a) Average CH duration. (b) Average CM duration. (c) Number of clusters. (d) Clustering efficiency.

As has been addressed in Section III-C, T_W is set to a value in $[T_{min}, T_{max}]$. We have tested the impact of T_{max} on our cluster performance, in which T_{max} is set to 0.5, 1.0, 2.0, 5.0, and 10.0s, respectively. According to the experimental results, we found that a larger T_{max}, for example when $T_{max} = 10.0s$, will cause longer cluster formation time, and a small T_{max}, for example when $T_{max} = 0.5$, will cause more packet collisions. Therefore, the value of T_{max} is fixed as 2 seconds, which shows the best performance in our simulation.

Fig. 3 compares the cluster macroscopic performance when applying different CH selection metrics. The comparison results of CH duration in Fig. 3(a) shows small difference when implementing these four methods, and CM duration is higher when implementing LLT-based and random-based methods. However, LLT-based and random-based methods create more clusters, comparing to distance-based method, shown in Fig. 3(c). In Fig. 3(d), clustering efficiency of these four methods are almost the same under each scenario. It means that the T_W calculation methods have no impact on the clustering efficiency. In addition, clustering efficiency decreases when the traffic scenario becomes more dynamic (from scenario A.1 to B.2).

To further analyze the impacts of CH selection methods, we also compare the cluster microscopic performance, shown in Fig. 4. Fig. 4(a) shows that random-based method selects less initial CHs than other three methods. The results in Fig. 4(b) illustrate that the CH selection method does not have big impact on vehicle state change rate. In UFC, vehicle changes state from UN to CH or to CCM at beginning. Therefore, each vehicle changes state at least once during the simulation. Fig. 4(c) presents CM disconnection rate. When traffic scenario becomes more dynamic, the CM disconnection rate decreases, and the differences of the CM disconnection rate among these four methods reduce. For example, under scenario B.2, the results are almost the same. We also notice that the value of random-based and speed-based method

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

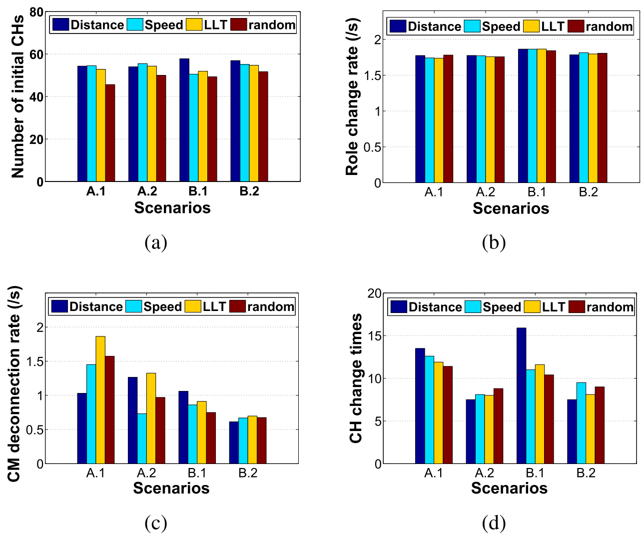

Fig. 4. Impact of T_W calculation method on the microscopic performance of UFC. (a) Number of initial CHs. (b) Role change rate (per second). (c) CM disconnection rate (per second). (d) Total CH change times.

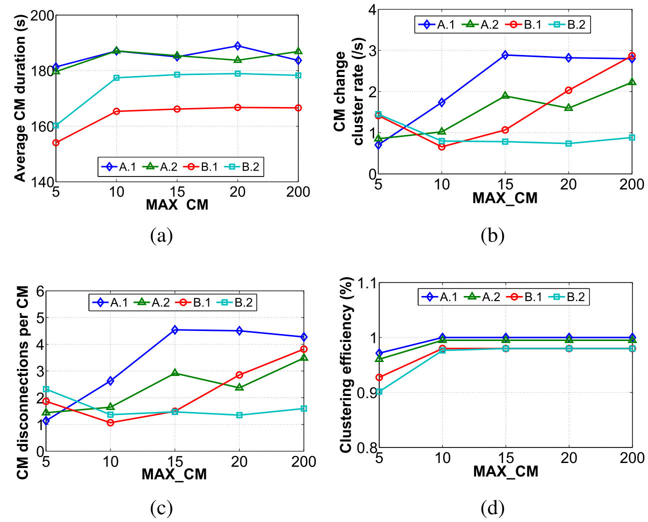

Fig. 5. Impacts of cluster capacity (MAX_CM) on cluster performance of UFC. (a) Average CM duration. (b) CM change cluster rate (per second). (c) CM disconnections per CM. (d) Clustering efficiency (%).

is always lower than that of LLT-based method under all scenarios. The comparison results of total CH change times in Fig. 4(d) show that the performance of distance-based method is more sensitive to the setting of traffic scenarios, compared with other three methods.

Therefore, from the results in Fig. 3, we can conclude that LLT-based and random-based CH selection methods perform better in terms of the overall cluster stability with the price of creating more clusters.

2) Cluster Capacity MAX_CM: Generally, cluster size is determined by vehicle's communication range in a single-hop clustering scheme. The increased vehicle density may cause high packet collision rate. Therefore, cluster size in UFC is controlled by a predetermined cluster capacity, i.e., the maximum number of CMs in the cluster, denoted by MAX_CM. In this section, we analyze the influence of cluster capacity on cluster performance by modifying MAX_CM under different scenarios. MAX_CM is set to 5, 10, 15, 20, and 200 vehicles per cluster, respectively. When MAX_CM is 200, it means that there is no limit on cluster capacity.

As shown in Fig. 5(a), and Fig. 5(d), CM duration and clustering efficiency have a sharp increase when MAX_CM changes from 5 to 10, and remain nearly stable with the increase of MAX_CM when MAX_CM is at least 10. It is because that when MAX_CM is too small, some original CMs are excluded from the new cluster due to the cluster capacity limit and become CHs during cluster merging. Such this phenomenon reduces the mean CM duration. In addition, when MAX_CM is too small, some vehicles remain in the CCM state because their neighboring clusters reach the capacity limits. It reduces the clustering efficiency. When MAX_CM is larger than the average number of neighbors, the above phenomenon seldom happens.

As shown in Fig. 5(b), and Fig. 5(c), CM change rate and the CM disconnection rate increase with the increase of MAX_CM for relative stable traffic scenarios (A.1 and A.2), and have a local optimal around MAX_CM=10 for dynamic traffic scenarios (B.1 and B.2). This is because, a cluster with more CMs has a higher probability to lose its CMs when compared to a cluster with less CMs. When the traffic is dynamic, cluster merging occurs frequently and causes frequent CM disconnections. In this case, when MAX_CM is too small, more CMs members will be excluded due to the cluster capacity limit during cluster merging, causing frequent CM disconnection.

In summary, we observe that, due to the comparison of different performance metrics, there is no single MAX_CM value that can optimize all performance metrics at the same time for different traffic scenarios. Nevertheless, when MAX_CM is 10, there is a good trade-off between these performance metrics. Therefore, in the rest part of the simulation, MAX_CM is fixed as 10.

3) Backup CH Number MAX_BCH: Vehicle clusters are considered as a backbone structure during information dissemination, data aggregation, packet delivering, and etc. An unexpected vehicle's disconnection may result in losing an emergency message. In the proposed BUCH approach, our objective is to ensure that the disconnected vehicles can successfully join another existing cluster as soon as possible. Considering the management overhead caused by the Backup CH maintenance procedure in each CM and CCM vehicle, the number of BCHs maintained in backup CH list *BCHL* should be carefully chosen to not only reduce re-clustering delay, but also avoid big overhead. The size of *BCHL*, denoted by MAX_BCH, is set to 1, 2, 3, 4, and 5, respectively. We evaluate the clustering performance in terms of CM re-clustering delay and CM re-clustering success ratio.

In Fig. 6(a), CM re-clustering delay is always far less than 1.0s except when $MAX\_BCH = 1$. When $MAX\_BCH = 1$, it means that CM/CCM vehicle only caches the first qualified BCH it hears to its *BCHL* without any prioritization.

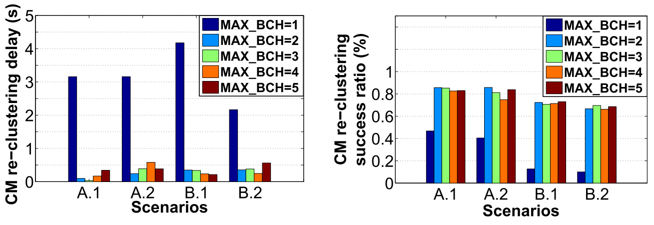

Fig. 6. Impacts of MAX_BCH on CM re-clustering performance. (a) CM re-clustering delay (s). (b) CM re-clustering success ratio (%).

Therefore, this BCH may be unreliable and could not be the best choice for the disconnected vehicles, resulting in a lower CM re-clustering success ratio, revealed in Fig. 6(b). When MAX_BCH is larger than 1, CM/CCM can prioritize the neighboring CHs it hears, and chooses the most reliable BCH as the target CH. Therefore, we can conclude that at least 2 BCH vehicles are required in order to both reduce the re-clustering delay and improve the re-clustering success ratio. Because there is no big difference in both the results in Fig. 6(a) and in Fig. 6(b) when MAX_BCH is larger than 1, and because recording redundant BCHs may cause a high $BCHL$ maintenance overhead and increase the CM re-clustering delay, MAX_BCH is fixed to 2 in the rest part of the simulation.

D. Performance Comparison With VMaSC and Lowest-ID

In this section, we compare the cluster performance of the proposed UFC approach to Lowest-ID [5], and VMaSC [28] from both macroscopic and microscopic levels, under four traffic scenarios. Lowest-ID is one of the most famous clustering algorithms. According to the summary in [35], most proposed clustering algorithms in VANETs have chosen Lowest-ID as a benchmark to compare with. In addition, VMaSC is one of the latest and the most cited clustering algorithms, which provides detailed simulation parameter settings.

According to the definition of Lowest-ID [5], we consider the awareness of moving direction. “LID (all)” indicates the original Lowest-ID scheme which can cluster vehicles moving in all directions, while “LID (same dir)” indicates the optimized Lowest-ID scheme, in which only vehicles moving in the same direction can be clustered. In addition, the original VMaSC [28] clustering algorithm is a multi-hop based approach. Since both Lowest-ID and UFC are one-hop based clustering algorithm, we implement the one-hop VMaSC scheme in this study, denoted as “VMaSC_1hop”.

We compare 2 sub-schemes of UFC, with or without Neighbor Sampling (NS) method, denoted as “UFC (w/o NS)” and “UFC (w/ NS)”, respectively. When implementing NS method, vehicle only recruits stable neighbors, meeting the condition $\Delta v \leq \Delta v_{th}$. On the basis of the previous simulation results, the default MAX_CM and MAX_BCH are set to 10 and 2, respectively. To make a fair comparison, we set the cluster updating interval to 5.0s in Lowest-ID scheme, which is same as T_{cm} in UFC and *CH_TIMER* in VMaSC scheme. The merging intervals are set to 5.0s both in UFC and VMaSC

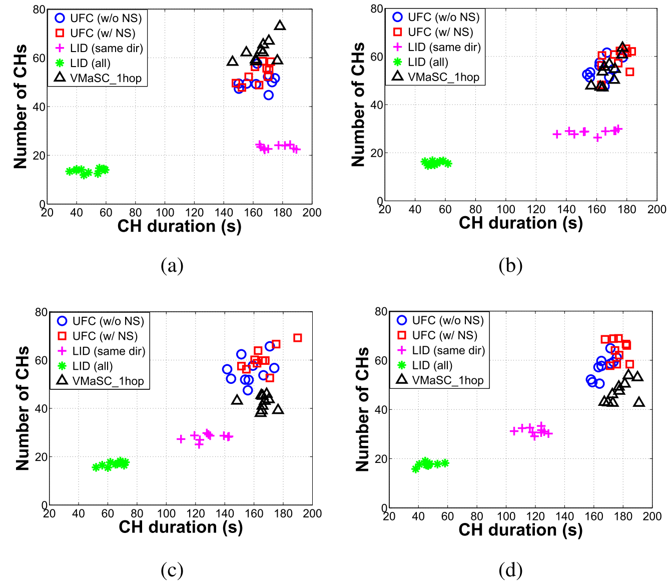

Fig. 7. CH lifetime comparison between UCF, LID, and VMaSC. (a) Scenario A.1. (b) Scenario A.2. (c) Scenario B.1. (c) Scenario B.2.

schemes. We repeat each simulation for 10 times. The nodes IDs in Lowest-ID are assigned randomly in each simulation.

In Fig. 7, we observe that LID (all) presents the minimum number of clusters and the smallest CH lifetime under all traffic scenarios. Lowest-ID scheme is a passive approach, in which, CM changes its CH as long as it hears a vehicle with a lower ID than its current CH; meanwhile, CH could become a CM as long as it hears a vehicle with a lower ID than itself. Therefore, LID (all) provides more chances for a CM to change its CH and for a CH to become a CM vehicle, since vehicles moving in opposite directions could also stay in the same cluster when they meet each other on the road. The frequent CH changing reduces CH lifetime, and the unlimited cluster size reduces the number of the created clusters. We also notice that under low dynamic scenario A.1, shown in Fig. 7(a), LID (same dir) performs the best in terms of CH duration and number of clusters. However, when the traffic becomes more dynamic, especially in Fig. 7(c) and Fig. 7(d), the average CH duration of LID (same dir) decreases significantly and becomes smaller than that of two UFC sub-schemes and VMaSC_1hop scheme.

Furthermore, the results in Fig. 7 presents the similar CH lifetime and number of clusters when implementing two UFC sub-schemes and VMaSC_1hop. Overall, UFC (w/ NS) performs better CH lifetime than UFC (w/o NS) scheme, especially in scenarios A.2, B.1, and B.2, which means that NS method is effective in increasing CH lifetime. Table IX and Table X calculate the averaged CH lifetime and averaged number of clusters according to the results in Fig. 7. In Table IX, we observe that the average CH lifetime of LID (all) is always the lowest one, about 65% lower than that of UFC schemes under all scenarios. Moreover, the average CH lifetime of UFC (w/ NS) remains the highest under scenario A.2 and B.1. Although VMaSC_1hop shows the highest average CH lifetime under scenario B.2, 178.243s,

TABLE IX
AVERAGE CH LIFETIME (S)

Scenarios	Schemes				
	UFC		LID		VMaSC
	w/o NS	w/ NS	same dir	all	1hop
A.1	163.943	163.112	**175.556**	47.92	164.875
A.2	163.814	**173.645**	156.969	53.164	167.801
B.1	157.386	**166.112**	128.589	64.2	165.779
B.2	166.626	176.57	119.717	46.762	**178.243**

TABLE X
AVERAGE NUMBER OF CLUSTERS

Scenarios	Schemes				
	UFC		LID		VMaSC
	w/o NS	w/ NS	same dir	all	1hop
A.1	50.41	53.47	23.32	**13.59**	62.71
A.2	54.56	58.56	28.52	**15.72**	53.72
B.1	55.59	60.44	28.12	**16.92**	42.35
B.2	57.23	63.95	31.27	**17.68**	47.1

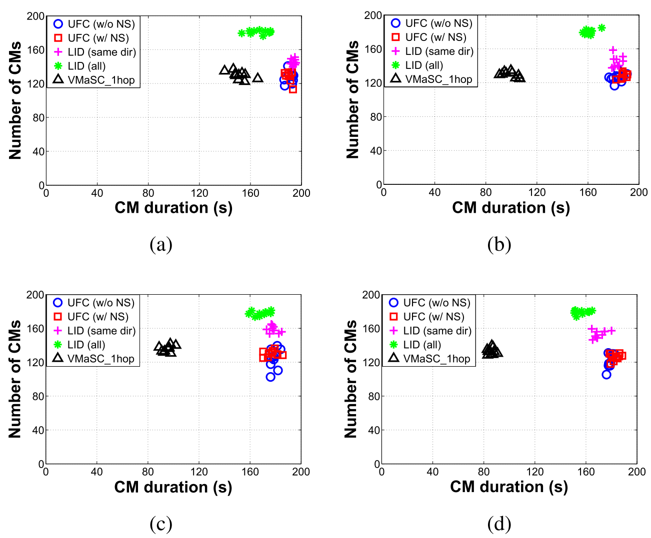

Fig. 8. CM lifetime comparison between UFC, LID, and VMaSC. (a) Scenario A.1. (b) Scenario A.2. (c) Scenario B.1. (d) Scenario B.2.

the value of UFC (w/ NS) is still the second highest one with a slight difference of 1.673s. However, the improvement of cluster stability in VMaSC and UFC always cause the increase of the number of created clusters.

Fig. 8 illustrates the CM lifetime and number of CMs. Apparently, LID (all) presents the maximum number of CMs in all scenarios. The reason is that vehicles moving in the opposite directions are allowed to stay in the same cluster and the cluster capacity is not limited in LID. Since the link connections between vehicles moving in the opposite directions are not stable, the CM lifetime of LID (all) is smaller than LID (same dir). LID (same dir) presents similar performance with two UFC sub-schemes under scenario A.1, A.2, and B.1,

TABLE XI
AVERAGE CM LIFETIME (S)

Scenarios	Schemes				
	UFC		LID		VMaSC
	w/o NS	w/ NS	same dir	all	1hop
A.1	191.065	190.395	**192.032**	166.986	151.436
A.2	182.959	**187.582**	182.484	161.418	99.604
B.1	177.129	**178.963**	178.107	168.706	95.073
B.2	179.313	**182.796**	170.514	156.194	85.897

TABLE XII
AVERAGE NUMBER OF CMS

Scenarios	Schemes				
	UFC		LID		VMaSC
	w/o NS	w/ NS	same dir	all	1hop
A.1	127.89	128.73	145.01	**180.63**	129.72
A.2	126.33	128.74	144.65	**179.82**	129.46
B.1	124.3	130.06	159.43	**177.68**	135.81
B.2	121.96	126.09	153.2	**179.05**	132.74

in terms of CM lifetime; and presents higher performance than UFC sub-schemes in terms of number of CMs under all scenarios. This is because the cluster capacity is not limited in LID. Furthermore, we are surprised by the performance of VMaSC_1hop in Fig. 8, which shows good performance in Fig. 7, but presents the lowest CM lifetime under all scenarios. We conjecture that the transitions between CM and unstable state node (denoted as SE in [28]) reduces the CM lifetime.

Table XI and Table XII calculates the averaged CM lifetime and averaged number of CMs based on the simulation results in Fig. 8. Apparently, UFC (w/ NS) performs the highest cluster stability under scenario A.2, B.1, and B.2, as shown in Table XI. Even though the performance is worse than that of LID (same dir) and UFC (w/o NS) under scenario A.1, the tiny differences may be ignored, 0.67s with UFC (w/o NS) and 1.637s with LID (same dir). Besides, we observe that when the traffic scenario becomes more dynamic, the average CM lifetime decreases quickly in all schemes, except two UFC sub-schemes. We can conclude that UFC scheme is more robust to the change of traffic scenarios. Table XII presents similar average number of CMs, except LID. LID (all) always performs the highest number of CMs under all scenarios.

The simulation results in Fig. 9 present the comparison between UFC (w/ NS), LID (same dir), LID (all), and VMaSC_1hop, in terms of CH change rate, role change rate, and clustering efficiency. In Fig. 9(a), we observe that CH changes most frequently in LID (all) under all traffic scenarios, especially under B.2. Even though CH change rate of LID (same dir) is the lowest under scenario A.1, the value increases significantly and becomes larger than that of UFC (w/ NS) and VMaSC_1hop when traffic scenario becomes more dynamic. The results are consistent compared to the

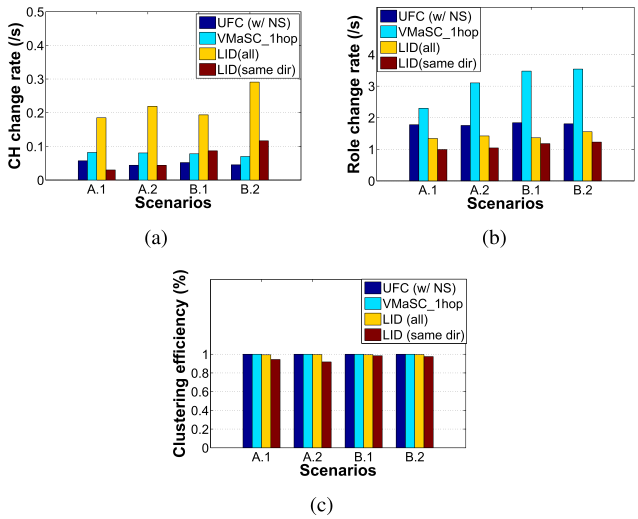

Fig. 9. Cluster performance comparison between UFC and LID scheme. (a) CH change rate (per second). (b) Role change rate (per second). (c) Clustering efficiency (%).

previous comparisons. On the contrary, the CH change rate of UFC (w/ NS) and VMaSC_1hop are both considered insensitive to the change of traffic scenarios, and UFC (w/ NS) always performs better than VMaSC_1hop.

In Fig. 9(b), we observe that vehicles change the state more frequently when implementing UFC (w/ NS) and VMaSC_1hop, compared to both LID (all) and LID (same dir). Since LID scheme is a passive scheme and vehicles only change state based on their neighbors' identifiers, its role change rate is not accurate and not comparable with UFC (w/ NS) and VMaSC_1hop. In addition, the role change rate of VMaSC_1hop remains the highest under all scenarios, because many CMs change to unstable state as long as they lose the connections with their CHs. On the contrary, the role change rate of UFC (w/ NS) scheme remains stable while traffic scenario changes. This is because the proposed NS scheme guarantees more stable vehicle link connections and BUCH scheme allows CMs to find backup CHs instead of changing their state immediately.

Fig. 9(c) illustrates the comparison results in terms of clustering efficiency. The clustering efficiency is always 100% with both UFC (w/ NS) and VMaSC_1hop approaches, under all traffic scenarios. This indicates that all vehicles on the road have participated the clustering process. However, only part of vehicles are able to participate in clustering. LID (same dir) and LID (all) always miss some vehicles, which may lead to the clustering inaccuracy.

E. Summary of Observations

From the results in Sections IV-C and IV-D, we summarize the main observations as follows:

1) When there is no backup CH, the relative mobility based metric (speed based) does provide more stability, as reflected by CM disconnection rate in Fig. 4(c). However, in a situation when there are at least 2 BCHs, it does not perform well compared to the link duration based metric (LLT based), and sometimes even to the random-based metric (reflected in Fig. 3(a) and Fig. 3(b), respectively). Nevertheless, the difference in CH duration and CM duration in such scenarios is just marginal. This illustrates the importance of backup CHs on the stability of clusters.
2) The UFC clustering scheme performs better than both VMaSC scheme and the passive Lowest-ID clustering scheme, especially in terms of CH and CM lifetime, CM re-clustering delay, and vehicle state change rate. When the traffic scenario becomes dynamic, only the performance of UFC scheme shows stability, compared with other schemes. Therefore, we can conclude that our proposed UFC approach is robust enough and presents higher cluster stability, especially under high dynamic traffic scenarios, even though the trade-off is a higher number of created clusters.

F. UFC Algorithm Analysis

1) Overhead Analysis: Clustering overhead describes the total number of clustering related packets. In UFC, during the NS process ($T_{collect}$ time period), only *Beacon* messages are exchanged. When $T_{collect}$ expires, every vehicle calculates their own mobility metric and sets a timer T_W. The proposed Backoff-based CH Selection (BCS) scheme avoids broadcasting mobility metrics. Instead, only the selected CHs broadcast *CHA* messages (around 50 *CHA* messages according to the simulation results in Fig. 4(a)). In the rest of the simulation, similar to one-hop VMaSC, a vehicle sends a *ReqJoin* message to the CH when it wants to join a cluster. Once a CH confirms the join of this vehicle, it sends back an *ACKJoin*. In addition, during cluster merging, two kinds of messages have to be transmitted, *ReqMerge* and *ACKMerge*.

In one-hop VMaSC, besides the control messages (*ReqJoin*, *ACKJoin*, *ReqMerge* and *ACKMerge*), a CH advertisement message should be broadcast as long as a SE (State Election) [28] vehicle transits to CH. Thus, with time increasing, the total number of the broadcasted CH advertisement messages increases. We assume that, all vehicles broadcast *Beacon* messages for the same Beacon Interval (1.0s) in both UFC and one-hop VMaSC, and the total number of the control messages are almost the same. In this case, although the overhead of UFC is larger than one-hop VMaSC at the beginning of the simulation (*CHA* broadcasting), the overhead of one-hop VMaSC increases more rapidly than UFC when the simulation time increases. Moreover, since one-hop VMaSC presents a higher CH change rate and role change rate than UFC according to the simulation results in Fig. 9 (a) and (b) (around 1.5 times of UFC), the number of control messages will be higher than UFC.

2) UFC Performance Analysis: According the above simulation results, in general, the proposed UFC clustering algorithm with Neighbor Sampling performs better than other clustering algorithms, especially in terms of CH and CM lifetime, CM re-clustering delay, and vehicle state change rate.

- During the cluster formation process in UFC, only the Stable Neighbors which are moving in the same direction can be added in the same group. CH may change the state only when cluster merging happens. However, the original Lowest-ID, LID (all), can combine vehicles moving in the opposite directions, which greatly reduces the CH lifetime and increases the CH change rate. Moreover, since LID is a passive clustering algorithm, CH changes state as long as it hears another vehicle with lower ID.

- In UFC, instead of changing to an unstable state immediately, CM will remain in the state for a tiny period of time (0-0.4s, shown in Fig. 6 (a)) to find BCH when it loses connection with the current CH. The BUCH re-clustering scheme allows the disconnected CM to build up-link connection with an appropriate BCH as soon as possible, aiming at reducing the CM re-clustering delay and increasing CM lifetime. On the contrary, in VMaSC, the disconnected CM will change state to State Election (SE) (unstable state) immediately, which decreases the CM lifetime.

In this work, the proposed UFC clustering algorithm has been tested under 4 traffic scenarios that are generated by SUMO, through modifying road conditions, vehicle velocity, speed acceleration, and speed deviation on one dimensional highway. The simulation results reveal that UFC shows superior performance comparing with LID and VMaSC algorithms. In the future work, we will verify the UFC performance under more practical settings, such as propagation model with fading and urban traffic scenarios with intersections, and re-design the CH selection metric accordingly.

3) Further Deployment of UFC: The proposed UFC algorithm can further solve the information dissemination in vehicular networks. UFC forms one-hop clusters, and CHs form the backbone of the vehicular networks. Instead of pure broadcast, CMs send information directly to the associated CHs. The CHs aggregate information and forward it to the destined vehicles or the related geographic regions. The transmission over the CH backbone can be in a carry-and-forward manner, so as to avoid expensive overhead on maintaining end-to-end path. To reduce the information dissemination latency, it is also possible to make use of fixed infrastructures, such as Road Side Units (RSUs). The CH can contact RSUs to disseminate information in the backbone, formed by RSUs, and let the RSU, the closest one to the destined vehicles or regions, to forward the information to the corresponding CH. For data with different delay requirements, the CH can decide whether to aggregate or not the received information.

Moreover, the further deployment of UFC should take the cluster size optimization into consideration. Vehicles can pre-compute a table of optimal cluster size subject to different combinations of road condition, requirement of applications, packet sizes, etc., then the cluster head will decide on-line about the suitable cluster size based on its location condition by table checking.

V. Conclusion

In this paper, we proposed a Unified Framework of Clustering approach (UFC) in vehicular ad hoc networks. UFC includes Neighbor Sampling (NS), Backoff-based Cluster head Selection (BCS), and BackUp Cluster Head based cluster maintenance (BUCH) schemes. NS scheme can filter out unstable neighbors in order to increase vehicle link stability. BCS scheme allows vehicles to make their own cluster head decisions in a distributed manner, which can reduce the clustering management overhead. Moreover, BUCH scheme guarantees that the disconnected CMs could rebuild connections with other CHs as soon as possible, thus, effectively reduces the CM re-clustering delay. We compared UFC's cluster performance with Lowest-ID and one-hop VMaSC algorithms under four different traffic scenarios, and observed that UFC scheme performs better cluster stability than Lowest-ID and one-hop VMaSC, especially under high dynamic traffic scenarios. Meanwhile, UFC scheme shows steady performance under different traffic scenarios.

As future work, we aim to investigate the use of UFC approach in complex urban traffic scenarios and extend it for data aggregation and data dissemination for recent VANETs applications.

References

[1] H. Moustafa and Y. Zhang, *Vehicular Networks: Techniques, Standards, and Applications*, 1st ed. Boston, MA, USA: Auerbach Publications, 2009.

[2] X. Cheng, L. Yang, and X. Shen, "D2D for intelligent transportation systems: A feasibility study," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1784–1793, Aug. 2015.

[3] G. Karagiannis *et al.*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 584–616, Nov. 2011.

[4] R. S. Bali, N. Kumar, and J. J. Rodrigues, "Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions," *Veh. Commun.*, vol. 1, no. 3, pp. 134–152, 2014.

[5] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, Sep. 1997.

[6] M. Chatterjee, S. K. Das, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 3. Nov. 2000, pp. 1697–1701.

[7] M. Gerla and J. T.-C. Tsai, "Multicluster, Mobile, multimedia radio network," *Wireless Netw.*, vol. 1, no. 3, pp. 255–265, Aug. 1995.

[8] E. Souza, I. Nikolaidis, and P. Gburzynski, "A new aggregate local mobility (ALM) clustering algorithm for VANETs," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.

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

[10] L. Khoukhi, A. E. Masri, A. Sardouk, A. Hafid, and D. Gaiti, "Toward fuzzy traffic adaptation solution in wireless mesh networks," *IEEE Trans. Comput.*, vol. 63, no. 5, pp. 1296–1308, May 2014.

[11] M. M. C. Morales, C. S. Hong, and Y. C. Bang, "An adaptable mobility-aware clustering algorithm in vehicular networks," in *Proc. 13th Asia–Pacific Netw. Oper. Manage. Symp.*, Sep. 2011, pp. 1–6.

[12] J. Zhang, M. Y. Ren, and H. Labiod, "Performance evaluation of link metrics in vehicle networks: A study from the cologne case," in *Proc. 6th ACM Symp. Develop. Anal. Intell. Veh. Netw. Appl. (DIVANet)*, New York, NY, USA, 2016, pp. 123–130.

[13] V. Véque, F. Kaisser, C. Johnen, and A. Busson, "CONVOY: A new cluster-based routing protocol for vehicular networks," in *Models and Algorithms for Vehicular Networks*, H. Labiod and A. Beylot, Eds. Hoboken, NJ, USA: Wiley, May 2013, ch. 4, pp. 91–102.

[14] N. Maslekar, J. Mouzna, H. Labiod, M. Devisetty, and M. Pai, "Modified C-DRIVE: Clustering based on direction in vehicular environment," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2011, pp. 845–850.

[15] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Veque, "A new mobility-based clustering algorithm for vehicular ad hoc networks (VANETs)," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2016, pp. 1203–1208.

[16] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Véque, "A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (VANETs)," *Veh. Commun.*, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.vehcom.2016.12.003

[17] R. Zhang, X. Cheng, L. Yang, X. Shen, and B. Jiao, "A novel centralized TDMA-based scheduling protocol for vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 411–416, Feb. 2015.

[18] P. Basu, N. Khan, and T. D. C. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proc. 21st Int. Conf. Distrib. Comput. Syst. Workshops*, Apr. 2001, pp. 413–418.

[19] B. Hassanabadi, C. Shea, L. Zhang, and S. Valaee, "Clustering in vehicular ad hoc networks using affinity propagation," *Ad Hoc Netw.*, vol. 13, pp. 535–548, Sep. 2014.

[20] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[21] M. A. Togou, A. Hafid, and L. Khoukhi, "SCRP: Stable CDS-based routing protocol for urban vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1298–1307, May 2016.

[22] M. A. Togou, A. Hafid, and L. Khoukhi, "A novel CDS-based routing protocol for vehicular ad hoc networks in urban environments," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[23] R. Zhang, X. Cheng, Q. Yao, C. X. Wang, Y. Yang, and B. Jiao, "Interference graph-based resource-sharing schemes for vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 8, pp. 4028–4039, Oct. 2013.

[24] S. A. Mohammad and C. W. Michele, "Using traffic flow for cluster formation in vehicular ad-hoc networks," in *Proc. IEEE Local Comput. Netw. Conf.*, Oct. 2010, pp. 631–636.

[25] A. Daeinabi, A. G. P. Rahbar, and A. Khademzadeh, "VWCA: An efficient clustering algorithm in vehicular ad hoc networks," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 207–222, Jan. 2011.

[26] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[27] Z. Zhang, A. Boukerche, and R. Pazzi, "A novel multi-hop clustering scheme for vehicular ad-hoc networks," in *Proc. 9th ACM Int. Symp. Mobility Manage. Wireless Access*, New York, NY, USA, 2011, pp. 19–26.

[28] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-cluster-based IEEE 802.11p and LTE hybrid architecture for VANET safety message dissemination," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2621–2636, Apr. 2016.

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

[30] Y. Chen, M. Fang, S. Shi, W. Guo, and X. Zheng, "Distributed multi-hop clustering algorithm for VANETs based on neighborhood follow," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, p. 98, 2015, doi: 10.1186/s13638-015-0327-0.

[31] W. Li, A. Tizghadam, and A. Leon-Garcia, "Robust clustering for connected vehicles using local network criticality," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 7157–7161.

[32] S.-S. Wang and Y.-S. Lin, "PassCAR: A passive clustering aided routing protocol for vehicular ad hoc networks," *Comput. Commun.*, vol. 36, no. 2, pp. 170–179, 2013.

[33] *The Network Simulator—ns-2*. Accessed on Jun. 30, 2016. [Online]. Available: http://isi.edu/nsnam/ns/

[34] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "SUMO—Simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, nos. 3–4, pp. 128–138, Dec. 2012. [Online]. Available: http://elib.dlr.de/80483/

[35] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A comparative survey of VANET clustering techniques," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 657–681, 1st Quart., 2017.

Mengying Ren received the B.Eng. degree in communications engineering from Northwestern Polytechnical University, China, in 2012, and the master's degree in computer science from the University of Technology of Troyes, France, in 2014. She is currently pursuing the joint Ph.D. degree from the University of Technology of Troyes and Telecom ParisTech. Her thesis focuses on vehicular networks. Her current research interests include wireless LAN and vehicular networks.

Jun Zhang received the B.Eng. degree in computer science and technology from Shanghai Jiaotong University in 2002, and the Ph.D. degree in computer science from The Hong Kong University of Science and Technology in 2007. He was a Research Fellow with the City University of Hong Kong from 2007 to 2009, The Hong Kong Polytechnic University from 2009 to 2011, a Post-Doctoral Fellow with The Hong Kong University of Science and Technology from 2011 to 2013, and currently a Post-Doctoral Researcher with Telecom ParisTech. His current research interests include wireless multi-hop networks (ad hoc/sensor/mesh networks), wireless LAN, and vehicular networks.

Lyes Khoukhi received the Ph.D. degree in electrical and computer engineering from the University of Sherbrooke, Sherbrooke, QC, Canada, in 2006. In 2008, he was a Researcher with the Department of Computer Science and Operations Research, Université de Montréal. Since 2009, he has been an Assistant Professor with the University of Technology of Troyes, France. His current research interests include vehicular networks, M2M and Internet of Things, cloud, performance evaluation, and security.

Houda Labiod received the Ph.D. degree from the University of Versailles Saint-Quentin-en-Yvelines, France, in 1998, and the HDR (Habilitation diriger les recherches) in 2015. She has been a Professor with the Department INFRES (Computer Science and Network Department), Telecom ParisTech, Paris, France, since 2001. Prior to this, she held a research position with the Eurecom Institute, Sophia Antipolis, France. She has authored or co-authored six books and many research papers in his research areas. Her current research interests include wireless networks and autonomous networks (WLANs/MANETs/WSN/NEMO/Mesh/vehicular), QoS, performance evaluation, and security and link adaptation mechanisms. She is a Founder of the IFIP NTMS Conference, on New Technologies, Mobility and Security (NTMS2007). She served as an Associate Editor and a member on the Editorial Board for several journals.

Véronique Vèque (SM'09) received the Ph.D. degree in communication networks from the Université Pierre et Marie Curie, France, in 1989. In 1990, she was an Associate Professor with the University of Paris-Sud (Paris 11), where she has been a Full Professor with the University of Paris-Sud, since 2000. She is currently a Research Member of the Laboratory of Signals and Systems. She has authored or co-authored over 100 papers in international journals or conferences. Her research interests lie in the field of both wireless, mobile and high speed communication networks with emphasis on vehicular networks, resource allocation, quality of service techniques, ad hoc routing, and performance evaluation.

She is a Senior Member of the IEEE Communications Society and the IEEE Vehicular Society. She has been involved on the technical program committee of the main conferences in networking, including GLOBECOM, ICC, PIMRC, ISCC, WLN, LCN, and WiMOB, and regularly invited to chair some of their sessions. She was the co-editor of a special issue on cognitive radio of the *IEEE Communication Magazine*.