International Journal of Computer Techniques -– Volume 6 Issue 1, Jan – Feb 2019

RESEARCH ARTICLE OPEN ACCESS

A High Performance Adaptive Clustering Scheme with a non-Handshake Size Control Mechanism in VANETs

Shiwen Huang[1], Xincheng Huang[1], Hongbin Huang[1,2], Weiping Liu[1]

[1](College of information science and technology, Jinan University, Guangzhou, China)
[2] (ZhongshanAiscent Technologies Ltd, Zhongshan, Guangdong, 528437, China)
* (Corresponding author: Shiwen Huang)

--************************----------------------------------

Abstract:

Vehicular ad hoc networks (VANETs), different from mobile ad hoc networks (MANETs), has characteristics including the high mobility of nodes (vehicles), the frequent changes of network topology especially under highway scenarios, the delay sensitivity and so on. Consequently, the clustering scheme proposed in MANETs is not always suitable for VANETs. In this paper, we proposed an adaptive clustering scheme for VANETs (ACSnH), focusing on three important aspects: 1) the exchange period of Hello packet; 2) the cluster head selection; 3) the size control of cluster member during cluster formation process. We adaptively adjust the period for the Hello packet according to vehicle state and link lifetime to reduce the overhead of clustering management and the probability of packet collision; we formulate the metric of cluster head selection by a weighted sum of contributions from the relative distance, relative velocity, and link lifetime; we optimize the formation process of clustering by a non-handshake size control mechanism which can directly control the size of cluster member without the help of cluster head. By comparing our ACSnH with other algorithms under different scenarios, we find that our ACSnH can realize an outstanding performance in terms of the cluster stability, re-clustering delay and clustering overhead.

Keywords **—Clustering, Vehicular ad hoc networks, non-handshake, NS-3, SUMO.**

--************************----------------------------------

I. INTRODUCTION

Vehicular ad hoc networks (VANETs), created by applying the principles of mobile ad hoc networks (MANETs) – a continuously self-configuring, infrastructure-less network of wirelessly connected mobile devices, are a vital components of the Intelligent Transportation System (ITS)[1]. With the rapid development of wireless communication technology and intelligent vehicles, vehicles presently, to some extent, have had the capacity to communicate with their neighbors in a certain range through a direct V2V (Vehicle-to-Vehicle) communication technology, or an indirect communication technology with the help of extra infrastructure for example roadside equipment called Road Side Units (RSUs)[2]. With these technologies, a variety of ITS applications can be implemented, such as the applications associated with traffic safety, traffic efficiency as well as infotainment. Numerous ITS applications and their associated requirements, along with challenges and solutions have been discussed and summed up in [3].

Although VANETs are born out of the MANETs, it is distinct from the traditional MANETs at least at the following aspects: the high mobility of nodes (vehicles), the fast change of network topology, high delay sensitivity, as well as the restricted and predictable movement trajectories of the nodes[4]. Besides, the issues of energy consuming and computing power of the nodes need not be considered in VANETs because of the huge volume of the vehicle, which are the important factors limiting the performance of the network in MANETs.

Due to the specific application scenario of VANETs mentioned above, the traditional clustering algorithms used in MANETs are not suitable for VANETs. In the case of VANETs, as the huge and the fast change of network topology and the high mobility of nodes, it is not easy to maintain a high enough transmission success rate of the packets between them. Therefore, a clustering algorithm, which can increase network capacity, realize the reuse of space resources, timely manage the information of nodes, improve communication efficiency and reliability, speed up routing lookup process, is must-be necessary for VANETs.

Clustering is a hierarchical network topology, which, according to specific rules, divides nodes into different states, such as cluster-head (CH), cluster-member (CM), cluster-gateway (CG) and so on. In a cluster structure, every cluster has and only has one CH which is responsible for managing and maintaining the information of CMs, as well as performing intra-cluster communication arrangement. CMis the common node of the network, which can communicate directly with the CMs or CH within the same cluster or indirectly with the nodes in other different clusters through its CH[5]. And the CG is a kind of inter-cluster link, proposed to forward the communication packet between neighbor clusters, such as in the works [6]–[10]. Generally, an excellent clustering algorithm can enhance cluster stability and extend cluster survival time without increasing clustering overhead.

In this paper, a new Vehicular Clustering Scheme is proposed. This scheme has the characters of a non-Handshake mechanism for CM size controlling and an adaptive adjustment of Hello packet transmission. The main contributions of this paper are listed as follows:

- We propose a Range Value Based non-Handshake (RVnH) mechanism to realize a passive CM size control in VANETs for the first time in the literature. Unlike traditional clustering formation mechanism, the RVnH uses a range value instead of a fixed value and a non-handshake mechanism instead of a handshake way to control cluster size and construct clusters, which can improve the cluster formation efficiency and reduce the clustering overhead and the CM re-clustering delay.
- We propose an Adaptive Adjustment of Hello Packet Period (LLTAAP) in order to reduce the clustering overhead. In LLTAAP, the period of Hello packet is adjusted by current state of vehicle and link lifetime between CM and its CH.
- We propose a Weight-Based Cluster Head Selection (WBCHS) method, which takes together the relative position, velocity and link lifetime of the vehicles into account, to promote cluster stability.
- We propose a detailed analysis of our proposed scheme and compare its performance with that of Lowest-ID (LID)[11] and VMaSC[12] under different scenarios, including the scenarios of vehicles continuing to keep in relative static or frequent changes.

The rest of this paper is organized as follows: Section II discusses the related work of clustering in VANETs. Section III presents the detail of our proposed clustering scheme. Section IV shows the performances of our proposed scheme under different simulation environments and their comparison with that of other algorithms. Finally, the conclusion and future work are given in Section V.

II. RELATED WORKS

In order to facilitate the management of vehicles and improve communication efficiency and reliability in the VANETs, different types of clustering algorithms are proposed. In this section, several relevant works about clustering mechanism in recent years are reviewed.

Clustering algorithm is originally well-known for MANETs, numerous research to discuss it has been proposed in the past. One of the most classical clustering algorithm is Lowest-ID[11], characteristics of nodes with unique ID and the periodically broadcasted Hello packets. However, it is not suitable for the dynamical networks with high mobility of the nodes since its selection method of CH without considering the mobility of nodes that selects always the CH with minimum ID. Thus, to tackle the situation with fast mobile nodes, a modified Lowest-ID algorithm called Mobility-based clustering algorithm (MOBIC) was proposed in [13]. In addition to these two classical algorithms, many other clustering algorithms are proposed for MANETs, such as the clustering based on mobility, topology, energy, weighted sum, neighbor nodes, and so on ([14]–[17]).

However, as the difference of the network topology between VANETs and MANETs, for example, in the aspects such as mobility velocity of node and channel conditions, those above-mentioned algorithms for MANETs are not necessarily applicable to VANETs. Therefore, a plenty of clustering schemes have been proposed specifically for the VANETs. These schemes can be classified into three different classes according to their individual clustering metric, differing in the number of parameters being considered into: one-parameter, multi-parameter and mixed(multi-weighted) parameter.

For the one-parameter scheme, ALM[18] presented a beacon-based clustering algorithm, which used the relative distance between two vehicles to select CH. Instead of using the relative distance, a multi-hop clustering scheme (K-hop) proposed in [19] by Zhang et al introduced a new mobility metric that calculated by the packet transmit delay. It is the first multi-hop clustering scheme for VANETs. VMaSC proposed in [12] is another multi-hop clustering algorithm for VANETs, which used the difference of velocity among neighbor vehicles to select cluster head. With using Simulation of Urban Mobility (SUMO) [20], VMaSC is known as the first multi-hop clustering algorithm to simulate under realistic traffic scenario.

Considering the influence of various factors on clustering comprehensively, multi-parameter clustering schemes are preferred to, which are better adaptable to the characteristics of VANETs. Ren et al[6] proposed a dynamic mobility-based clustering scheme, which is based on the mobility patterns of vehicles, including moving direction, relative velocity, relative distance, and link lifetime. In order to help cluster formation and control cluster size, the concepts of “temporary cluster head” (CHt) and “safe distance threshold” are proposed. To solve the scalability problem in VANETs, UFC[21] proposed a unified framework of clustering approach, which is composed of three parts: neighbor sampling, backoff-based cluster head selection, as well as cluster maintenance based on backup cluster head.With proposing two potential schemes to calculate backoff timer to select CH–one of which is considering the influence of relative velocity, relative distance and link lifetime of vehicles, the other one is using only a random value to calculate–UFC provides a framework that can work for both single or multi parameters metrics. However, UFC increases the average number of clusters and the role change rate, and relies more on synchronization.

For the mixed scheme, it characters with a parameter-weighting algorithm which weights the factors in the light of its importance. An efficient clustering algorithm proposed in [22] introduced three algorithms for VANETs, one of which called VWCA evaluated a weighted sum as the clustering metric that took into consideration the number of neighbors within dynamic transmission range, vehicles direction, the entropy and the distrust value parameters. Another algorithm used another different weighted sum as clustering metric is AMACAD proposed in [23]. It is an adaptable mobility-aware clustering algorithm based on destination positions, with the weighted sum being calculated according to relative distance, relative velocity and the final destination of vehicles.

In the process of clustering formation, some of the works use an affiliation handshake mechanism which refers to the vehicles joining process that vehicle sends a join request to a CH, and will join in the CH if it can receive a join response from the CH. This mechanism that is always used aims to control CM size per cluster or to help CH to complete the clustering joining authentication process. Both VMaSC[12] and UFC[21] mentioned above use this handshake mechanism to control the size of CM: when a vehicle is ready to join a cluster, it firstly send a join request to a CH; then CH will scan its CM list and judge whether the number of its CMs is less than the specified max size; if the answer is yes, CH will send a join response to the vehicle; finally after receiving this response, the vehicle will join this cluster. For another, AMACAD[23] is another clustering algorithm using the handshake mechanism. However, AMACAD do not use it to control CM size, but to find an existing neighbor CH to join in. Other cluster algorithms do not use a handshake mechanism, such as ALM[18] and K-hop[19] introduced above, which can reduce the management overhead for there is no round-trip affiliation packets. But they do not have an effective mechanism to control the size of CM, which has a certain effect on the performance of clusters: the larger the CM size is, the greater the overhead will be; while the smaller the CM size is, the more the roles change rate will be, and consequently the less of the cluster stability.

In other word, if we can control the cluster size without the handshake mechanism, not only the overhead can be reduced, but also the cluster stability can be enhanced. Therefore, we here propose a passive CM size control scheme based on rang value and non-handshake (RVnH) to enhance the performance of cluster. For another, the RVnH that joining cluster without waiting the response from CH can also obviously reduce the CM re-clustering delay.

Furthermore, most of the clustering scheme, such as all the clustering algorithm mentioned above, used the Hello packet (sometimes is called the beacon) to exchange vehicles information periodically. Generally, a too long-time interval of Hello packets sending-period will lead to vehicles not being able to update the information of their neighbors in time. In contrast, a too short time interval of sending-period will cause a huge overhead and high probability of packet collision. Consequently, an appropriate time interval of period should be explored. In both [24] and [25], several existing adaptive beacon approaches have been reviewed, analyzed and compared. In this paper, we also proposed a method (LLTAAP) that adjust the Hello packet period adaptively basing on the current states of vehicles and the links lifetime between CMs and their CHs to solve the problems mentioned above. The results show that the overhead is significantly reduced on the premise of clustering stability ensured.

In [26], a survey, comparison and summary of various clustering techniques in VANETs is discussed in detail. According to this summary, Lowest-ID[11] and MOBIC are the most classical benchmarks to be compared with when a new clustering scheme is proposed. In addition, with providing the detail simulation parameter settings, VMaSC[12], which is more and more as a benchmark for comparison, becoming one of the latest and the most cited clustering algorithms.

III. PROPOSED SCHEMES

We assume that every vehicle is equipped with a GPS system, in order to obtain its real-time information, such as the moving direction (Dir_i), current location (x_i, y_i) and velocity (v_i), which can be used as metrics to calculate the weighted sum for the determination of states and the CH selection. This basic motion information of vehicles is carried by Hello packets that are expected to be exchanged periodically among vehicles in one-hop. Besides carrying this motion information, Hello packets also contains the information of vehicle identifier (ID), mobility weighted sum (MW_i), CH ID (CH_id_i), number of the connected CM (CM_num_i), current state ($STATE_i$), and the number of vehicles in stable transition state within the range of transmission ($stableTS\_num_i$). The detailed notations used are in Table I.

TABLE I
NOTATIONS

Notation	Description	Notation	Description
IS	Initial State	T_{TS}	TS state timer
TS	Transition State	T_{CH}	CH state timer
CH	Cluster Head	T_{CM}	CM state timer
CM	Cluster Member	T_{merge}	Merge timer
V_i	Vehicle i, i is	STS_i	Stable

	the ID of vehicle		neighborhood vehicles in TS state set of V_i
Dir_i	Direction of V_i	Δv_{ij}	Velocity difference between V_i and V_j
HP_i	Hello Period of V_i	$\overline{\Delta v_i}$	Average velocity difference between V_i and its stable neighbors in TS state
NVL_i	Neighbor Vehicle List of V_i	ΔD_{ij}	Distance between V_i and V_j
CML_i	Cluster Member List of V_i	$\overline{\Delta D_i}$	Average distance difference between V_i and its stable neighbors in TS state
CH_id_i	CH id of V_i connection if it is a CM	LLT_{ij}	Link lifetime between V_i and V_j
CM_num_i	Number of CM has connected to V_i if it is a CH	$\overline{LLT_i}$	Average LLT between V_i and its stable neighbors in TS state
$[\mu, \psi]$	The range of CM max size	MW_i	Mobility weighted sum of V_i
TR	Transition Range	RD_i	A random number for V_i
$STATE_i$	Current state of V_i	TH_i	Join-in threshold of CH_i
$stableTS\_num_i$	The number of stable neighbor vehicles in TS state of V_i	ΔLLT_{th}	The threshold of LLT to judge stable TS
T_{IS}	IS state timer		

Through the periodic exchange of Hello packets, one vehicle can obtain the information of its surroundings vehicles and stores them in its Neighbor Vehicle List NVL_i. And to reduce the overhead of the cluster management, we proposed the exchange-period of Hello packets named Hello Period (HP) which can be adaptively adjusted by link lifetime and state of vehicle.

A. *States of Vehicles*

In our clustering scheme, each vehicle works in one of the following states shown in Fig. 1:

Initial State (IS): At the beginning of simulation, all vehicles are in this state. It is the initial state in ACSnH, which means that the vehicle does not belong to any cluster.

Transition State (TS): This is a temporary transition state. When a vehicle leaves its current cluster, it will enter this state until it belongs to a new cluster.

Cluster Head (CH): The leader in its cluster, which is responsible for managing and maintaining the information of its members and performing intra-cluster communication arrangement.

Cluster Member (CM): A common node in a cluster which can communicate with CMs or CH in its own cluster directly.

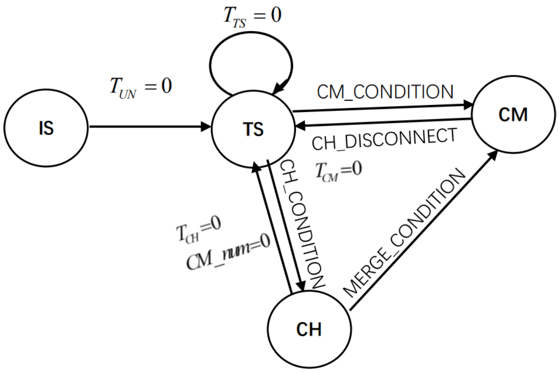

Fig.1 ACSnH state transition diagram

The transformation of the vehicles from one state to another is triggered by different events, which is schematically shown in Fig. 1 in detail. At the beginning, all the vehicles stay in IS state and exchange information with their neighbors until T_{IS} expires, then the vehicles will change their states to TS and start T_{TS} timer. If

CM_CONDITION is triggered, the vehicle at TS state will change its state to CM and the timer T_{CM} will be started meanwhile, where CM_CONDITION refers to the condition that a vehicle at TS state can join the cluster. The vehicle staying in the CM state will continue to keep its state until the T_{CM} expires and the condition of CH_DISCONNECT is satisfied which means that the vehicle at CM state fail to connect to its CH, then it will change its state to TS again. On the other hand, if the condition of CH_CONDITION is triggered, the vehicle in TS state will change its state to CH, where CH_CONDITION refers to the condition that a vehicle in TS state can be a CH. The vehicle in the CH state will turn back its state to TS if the number of CM is none and the timer T_{CH} expires at the same time. In order to minimize the number of clusters, the vehicle is assumed to try to join an existed cluster firstly, then if it cannot join an existed cluster, it will try to build a new cluster. The detail state transition process flow chart is shown in Fig. 2.

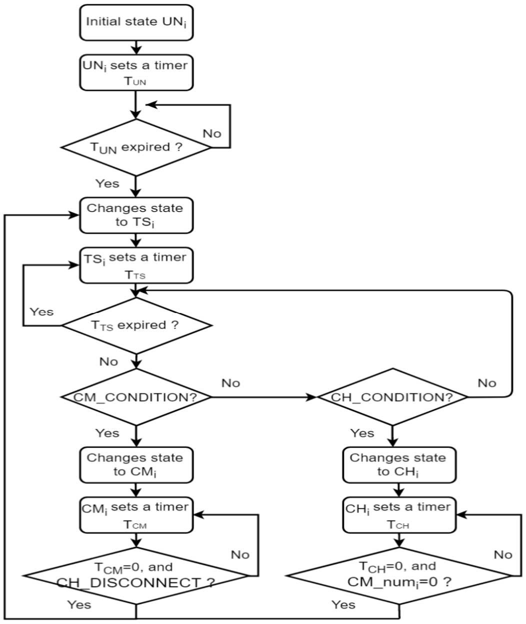

Fig.2 State change process (cluster formation)

If there are two CHs entering into the transition range of each other, the condition of MERGE_CONDITION will be satisfied, and then the CH with higher MW_i will give up its own CH status, and become a CM which belongs to the other CH.

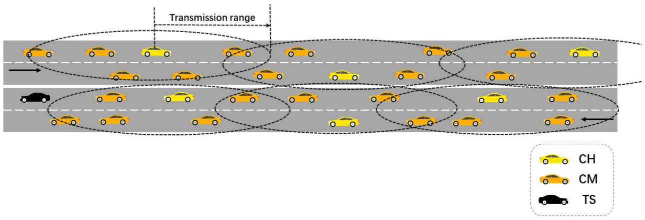

Fig.3 Example of cluster formation on highway

The example of cluster formation on highway is shown in Fig. 3, where six clusters are presented. The road topology is set to a two-way road with two lanes in each direction, and vehicles can only cluster with the vehicles that in the same direction as themselves. Vehicle with the minimum mobility weighted sum in the cluster become CH. While if a vehicle leaves the original cluster and does not belong to a new cluster, it enters the TS state.

B. Adaptive Adjustment of Hello Packet Period–LLTAAP

In the traditional clustering algorithm and most of the algorithms currently proposed, the distribution period of Hello packets (sometimes called beacons), which determines the timeliness of neighbor vehicles information stored by each vehicle, are fixed. How to select a suitable value of the period is essential for the performance of the clustering algorithm. A too large value of the period may seriously lag the received message of the state information of the neighbor vehicles. However, a too small value of the period will result in additional management overhead, leading to an excessive consumption of communication bandwidth which is not conducive to the effective transmission of data.

In this paper, we propose an adaptive adjustment period (LLTAAP) for the Hello packet, which based on states and link lifetime (LLT) [27], to solve the above problems. LLT, also called Link Expiration Time (LET)[28], describing the duration time of continuous communication between two vehicles, is defined as [27]:

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

where TR is the transmission range of a vehicle, v_i is the velocity of vehicle i, (x_i, y_i) is the position of vehicle V_i , $\Delta v_{ij} = v_i - v_j$ and $\Delta D_{ij} \approx x_i - x_j$ are the relative velocity and distance between V_i and V_j with ignoring the influence from y-coordinate on the LLT as the width of the road (y-coordinates) is far less than the vehicle communication range and driving safety distance.

Considering that in a real application scenario, most of the nodes stay in the CM state. For the CMs, as they do not need to manage other vehicles and transmit packets as CH does, the HP of them are theoretically permitted to be longer compared with that of CH and TS. Therefore, the management overhead guaranteeing that the vehicle information can be timely updated can be effectively reduced according to the HP of CMs which are dictated by the calculated LLT between CMs and their CHs. That is to say, the longer of the LLT is, the longer of HP or the less overhead will be, and vice versa.

For the CHs, their HP should not be too long to ensure the stability of the cluster due to the requirement that they should maintain the communication links with all the CMs, as well as the requirement that the new TS vehicles can be able to get the cluster head information and join the cluster in time. On the other hand, the HP of CHs can be small because of the small number of them compared with that of CMs in the VANETs and their negligent influence on the communication overhead.

And for the TSs, the HP of them should be also set to be small because of the requirement of updating the information about neighbor CHs in time to reduce the re-connection delay. For another, with the small number, TSs has little impact on total communication overhead.

Therefore, according to the above description, the HP of vehicles can be given as follow:

$$HP_i = \begin{cases} 1.0 & LLT_{ij} \in [20, +\infty), STATE_i \in \{CM\} \\ 0.5 & LLT_{ij} \in [5, 20), STATE_i \in \{CM\} \\ 0.2 & LLT_{ij} \in (0, 5), STATE_i \in \{CM\} \\ 0.2 & STATE_i \in \{CH, TS\} \end{cases} \qquad (2)$$

where $STATE_i$ is the current state of vehicle V_i, LLT_{ij} means the LLT between CM vehicle V_i and CH vehicle V_j. That is to say, if Vehicle V_i stays in CH or TS state, the HP of it is set to 0.2 s. Or if it is in CM state, the HP_i is determined by the length of LLT with its CH_j.

C. Weight-Based Cluster Head Selection (WBCHS)

For a TS vehicle V_i, if it receives the Hello packet of another TS vehicles V_j more than one time, the CH selection mechanism will be triggered. Then, it gets the information of its neighbors, such as the current velocity v_j, position (x_j, y_j) and so on, by the Hello packets it received. With these information, V_i can calculate its relative velocity Δv_{ij} and relative distance ΔD_{ij} with V_j, and the

LLT_{ij} according to Eq.(1). Then, V_i can calculate the metric $\overline{\Delta v_i}$, $\overline{\Delta D_i}$, and $\overline{LLT_i}$ as Eq. (3)–(5),

$$\overline{\Delta v_i} = \frac{\sum_{V_j \in STS_i} \left|\Delta v_{ij}\right|}{stableTS\_num_i} \tag{3}$$

$$\overline{\Delta D_i} = \frac{\sum_{V_j \in STS_i} \left|\Delta D_{ij}\right|}{stableTS\_num_i} \tag{4}$$

$$\overline{LLT_i} = \frac{\sum_{V_j \in STS_i} LLT_{ij}}{stableTS\_num_i} \tag{5}$$

where STS_i is the set of stable TS neighbor vehicles of V_i, and $stableTS\_num_i$ is the number of vehicles in STS_i of vehicle V_i. The stability criteria is described and calculated as Eq. (9).

Then the weighted sum of V_i can be calculated as following

$$MW_i = \alpha * \frac{\overline{\Delta v_i}}{\Delta v_{\max}} + \beta * \frac{\overline{\Delta D_i}}{\Delta D_{\max}} + \delta * \left(1 - \frac{\overline{LLT_i}}{LLT_{\max}}\right) \tag{6}$$

with

$$\alpha + \beta + \delta = 1 \tag{7}$$

where, in this paper, $\Delta v_{\max}$ is set to 20 m/s, $\Delta D_{\max}$ and $LLT_{\max}$ are assigned to be the same value with the TR and T_{sim}.

Algorithm 1 :Weight-based CH Selection (WBCHS)

```
$TS_i$ starts timer $T_{TS}$
$TS_i$ calculates $MW_i$ and $HP_i$
while $T_{TS} > 0 \&\& flag\_TS == 1$ do
    if $TS_i$ receives Hello packet from $V_j$ and $Dir_i == Dir_j$ then
        $TS_i$ updates $NVL_i$
    end if
    if $NVL_i$ includes $CH_j$ then
        sort $CH_j$ according to $MW_j$
        for all $CH_j \in NVL_i$ do
            goto Cluster Formation: RVnH
        end for
    end if
    if $NVL_i$ includes $TS_j$ and $flag\_CM == 0$ then
        if $MW_i$ is the smallest in neighbor $TS_j$ then
            $flag\_CH \leftarrow 1$
            $flag\_TS \leftarrow 0$
            $TS_i \rightarrow CH_i$
            $CH_i$ starts timer $T_{CH}$
            $CH_i$ calculates $HP_i$
            $CH_i$ broadcasts Hello packet
        end if
    end if
end while
```

It can be seen from Eq. (6) that the MW_i value of the vehicle is proportional to the average relative velocity and relative distance and inversely proportional to the link survival time. Therefore, the vehicle with the minimum MW value in all stable TS neighbors is more preferred for being considered to be CH as it is beneficial to the stability of the cluster. Finally, the newly generated CH also broadcast its Hello packets to inform other vehicles to join in, and then the vehicles in TS state which receive the Hello packets sent by this CH will try to join it. The CH election process is in detail described in Algorithm 1.

D. Cluster Formation: Passive CM Size Control Based On Range Value and Non-handshake (RVnH)

If a vehicle joins a cluster with a traditional mechanism[12] – a handshake mechanism, it will send a join request to a certain CH first, then the CH determines whether the direction of the vehicle is the same, the cluster metric is within a certain threshold, and the number of its CMs has reached the maximum, and so on. If all the conditions are satisfied, then the CH will send out a join response to the vehicle and affirm that the vehicle is a CM of its own. On the other side, after receiving a join response from CH, the vehicle then changes its state to CM and joins in the cluster.

In fact, an important goal of the handshake process is to control the CM size, which always leads to the large CM disconnection rate and high management overhead. So, a question may be raised that can this process be cut out. The answer is yes if we control the CM size not by CH of the cluster. Actually, we can control the CM size passively by the TSs which decide whether to join a cluster or not by themselves.

However, if without the help of CH, the CM size of the cluster may suddenly grow too large in the case that several vehicles join the same cluster at the same time, as the result of the delay in packet transmission and the possible failure of transmission. To avoid the appearance of the above situation, in this paper, we introduce a CM size control mechanism by assigning a value range $[\mu, \psi]$ instead of a fixed value for the max size of CM and a join-in threshold for CH.

The lower limit of the range value μ allows whether the CH accepts new CMs and the upper limit ψ is related with the join-in threshold that dictates the probability of the CH that can be successfully joined. These two parameters are to be optimized in experiment.

The join-in threshold of CH (TH) can be specified by the current number of the existed CM, the stable TS around CH, and the upper limit of the range value. The detail description of TH is as follows.

Whenever a vehicle V_i receives a Hello packet sent from another, it updates its NVL_i. In the cluster formation process, TS_i firstly scans the neighboring CHs in the increasing MW_{ij} order in its NVL_i, $CH_j \in NVL_i$. Subsequently, TS_i gets the number of CM(CM_num_j) that the CH_j has had and judges whether CM_num_j is smaller than μ. If that, CH_j allows a new TS vehicle join in it, then TS_i will acquire a random number RD_i ($RD_i \in (0,1)$) and calculate the probability of a vehicle to join CH_j, called join-in threshold TH_j, which is given as

$$TH_j = \begin{cases} \dfrac{\psi - CM\_num_j}{stableTS\_num_j} & stableTS\_num_j \neq 0 \\ 1.0 & otherwise \end{cases} \qquad (8)$$

where $stableTS\_num_j$ is the number of stable $TS_{i'}$ neighboring vehicles of CH_j, determining by a criteria as

$$LLT_{ji'} \geq \Delta LLT_{th} \qquad (9)$$

where ΔLLT_{th} is experimentally set to 10 seconds.

Algorithm 2 :Cluster Formation: RVnH

```
if $CM\_num_j < \mu$ then
$TS_i$ calculates $RD_i$ and $TH_j$
if $RD_i < TH_j$ then
  $flag\_CM \leftarrow 1$
  $flag\_TS \leftarrow 0$
 $TS_i \rightarrow CM_i$
 $CH\_id_i \leftarrow V_j$
 $CH_j$ updates $CML_j$
 $CM_i$ starts timer $T_{CM}$
 $CM_i$ calculates $HP_i$
end if
```

```
end if
```

The purpose of calculating the threshold TH_i is to avoid a large number of TSs joining the CH_j at the same time, so as to achieve the purpose of controlling the size of CM based on the number of stable TSs around CH_j, which is also the number of vehicles that may join CH_j. When $RD_i < TH_j$ is satisfied, TS_i joins CH_j and changes its state to CM_i, otherwise tries to join next CH. If it cannot join any cluster and has other TS vehicles around, it will restart the cluster head selection mechanism. The cluster formation (RVnH) is described in Algorithm 2.

For the whole process of cluster formation mentioned above is completed in a distributed way, without waiting for a response from CH, TSs can join in a CH as soon as possible.

E. Cluster Merge

Cluster merge is triggered when there are two CHs with the same motion direction traveling to communication range of each other and enduring over T_{merge}. With the information got from Hello packets, CH_i and CH_j can know CM_num of each other. If $CM\_num_i + CM\_num_j < \psi$ is satisfied, the vehicle, for example CH_i who is with the higher MW_i, will send a merge request to CH_j. On the other hand, when receiving merge request from CH_i, CH_j will update its CML_j and send also a merge response to CH_i. After receiving merge response, CH_i will give up its CH role, change its state to be CM_i that belongs to CH_j and send a broadcast to inform its CMs to join CH_j. If the CM receiving the broadcast belongs to CH_j is in the communication range of CH_j, it will become a CM of CH_j, or it will change its state to TS.

IV. PERFORMANCE EVALUATION

In this section, a deep analysis of our proposed adaptive clustering scheme (ACSnH) is provided, and the simulation results are compared with the most classical benchmarks LID[11] and the latest and most cited clustering algorithm VMaSC[12] in different scenarios. As the ACSnH is a one-hop clustering algorithm, the VMaSC algorithm is implemented in the one-hop way in our simulation for comparison. The simulations are performed in the Network Simulation ns-3 (Release 3.25) [29], with the testing scenarios and the realistic mobility of the vehicles generated by Simulation of Urban Mobility(SUMO)[3]. Detailed simulation configurations and test scenarios are described below.

A. Testing Scenarios

Referring to the simulation scenarios presented in UFC[21], this paper also adopts four simulation scenarios generated by SUMO. In all simulation scenarios, the road topology is set to a two-way road with two lanes in each direction. The total length of the road is 10 km, which is divided into 10 segments on average. The velocity limit of each segment is different in order to imitate the real road scenarios. At the beginning of the simulation, there are two vehicles injected into the road per second until a total of vehicles driving on the road are up to 200. The transmission range (TR) of each vehicle is set to 200 meters. The simulation runs for 355 seconds (T_{end}), and the clustering process starts at 155 seconds (T_{start}) when all the vehicles have entered road. Therefore, all of the cluster performance metrics are evaluated for the remaining 200 seconds ($T_{sim} = T_{end} - T_{start}$). The detail of simulation settings is shown in TableII.

TABLE II
SIMULATION PARAMETERS FOR VANETS

Notation	Value
Simulation Time (T_{sim})	200 s
T_{start}	155 s
T_{end}	355 s
Length of Road	10 km
Number of Vehicles	200
Transmission Range	200 m
Hello Package Size	64 bytes
$\Delta v_{\max}$	20 m/s
ΔLLT_{th}	10 s
Max Size of CM($[\mu, \psi]$)	[9, 11]
T_{IS}	2.0 s
T_{TS}	2.0 s
T_{CH}	5.0 s
T_{CM}	10.0 s
T_{merge}	2.0 s
MAC protocol	IEEE 802.11p
Channel Model	YANS[30]

The four testing scenarios are divided into the relatively stable scenarios ST.1 and ST.2, and the relatively dynamical scenarios DY.1 and DY.2. In ST.1 scenario, there are only one type vehicles on the road, and the velocity limit of all the segments on the road are set to the same value. The road settings of ST.2 scenario is the same as that of ST.1 scenario, but with four different types of vehicles on the road. In DY.1 and DY.2 situations, the velocity limits of the ten road segments are totally different, and the types of the vehicles are set to be four. The detail settings of vehicles in each scenario is shown in Table.III, IV and V, and the settings of road segments in each scenario is shown in Table.VI.

In addition, as the dynamic parameters that dynamically assigned according to each scenario, the weights of cluster head selection in Eq. (6) (α, β, δ) are set according to the outcome of the set of simulations. We find that the weights with best results according to CH duration are $\alpha = 0.3$, $\beta = 0.3$ and $\delta = 0.4$ for ST.1 and ST.2, $\alpha = 0.7$, $\beta = 0.01$ and $\delta = 0.29$ for DY.1, $\alpha = 0.01$, $\beta = 0.7$ and $\delta = 0.29$ for DY.2 respectively. Therefore, we use these values to set and run our simulation respectively.

B. Performance Analysis and Comparisons

In this section, the performance of ACSnH is analyzed and compared with LID and VMaSC. To make a fair comparison, the max size of CM in LID and VMaSC are set to 10, which is the median of the range interval $[\mu, \psi]$ in ACSnH, and T_{SE} is set to 2 seconds, the same as T_{TS} in ACSnH. Other timers, such as T_{IS}, T_{CM}, T_{CH} and T_{merge} are set as the same as ACSnH, and the timer to update the clustering in LID is set to the same as T_{CM} in ACSnH. In order to better contrast and verify the performance of LLTAAP proposed in this paper, the HP is set to a fix value (such as 0.2 s, 0.5 s and 1.0 s) in ACSnH and VMaSC, making a version of ACSnH(0.2), ACSnH(0.5), ACSnH(1.0), VMaSC(0.2) and VMaSC(1.0) respectively. The HP is set to 1.0 s in LID. And the ACSnH in the picture refers to the method that uses the LLTAAP proposed in Section III. On the other hand, vehicles both in VMaSC and LID are set to only can cluster with vehicles in the same direction, and the id of vehicles in LID are randomly set.

The performance metrics used for analyzing and comparing include average CM duration, average CH duration, average number of CHs and the temporary state in algorithms (such as TS in ACSnH or SE in VMaSC), CM disconnections rate, role change rate, clustering overhead, and CM re-clustering delay.

1) ***Average Duration of CH and CM:*** CH duration, which presents the lifetime of cluster, is defined as the average time from a vehicle becoming an CH to becoming another state. CM duration refers to the average time of a vehicle from joining a cluster to becoming an CM until it leaves its own cluster. Generally, the longer the duration of CH and CM is, the better the stability of the cluster will be.

TABLE III
SETTINGS OF VEHICLES IN ST.1

Type	Max Speed	Length	Acceleration	Deceleration	Sigma	Speed Deviation
Car A	20 m/s	3 m	2.0 m/s^2	6.5 m/s^2	0.5	0.1

TABLE IV
SETTINGS OF VEHICLES IN ST.2

Type	Max Speed	Length	Acceleration	Deceleration	Sigma	Speed Deviation
Car A	20 m/s	3 m	2.9 m/s^2	7.5 m/s^2	0.5	0.7
Car B	20 m/s	3 m	2.9 m/s^2	7.5 m/s^2	0.5	0.3
Car C	20 m/s	3 m	2.0 m/s^2	6.5 m/s^2	0.5	0.1
Car D	20 m/s	3 m	1.5 m/s^2	5.5 m/s^2	0.5	0.3

TABLE V
SETTINGS OF VEHICLES IN DY.1 AND DY.2

Type	Max Speed	Length	Acceleration	Deceleration	Sigma	Speed Deviation
Car A	35 m/s	3 m	2.9 m/s^2	7.5 m/s^2	0.5	0.7
Car B	25 m/s	3 m	2.9 m/s^2	7.5 m/s^2	0.5	0.3
Car C	20 m/s	3 m	2.0 m/s^2	6.5 m/s^2	0.5	0.1
Car D	10 m/s	3 m	1.5 m/s^2	5.5 m/s^2	0.5	0.3

TABLE VI
SETTINGS OF SEGMENTS OF ROAD (M/S)

Segments	1	2	3	4	5	6	7	8	9	10
ST.1	20	20	20	20	20	20	20	20	20	20
ST.2	20	20	20	20	20	20	20	20	20	20
DY.1	20	30	20	25	15	20	15	10	30	20
DY.2	20	10	15	30	25	20	15	30	15	20

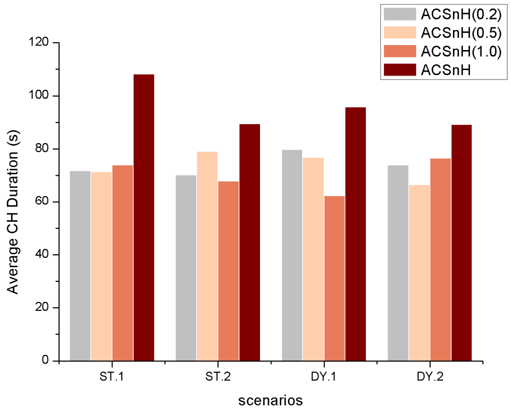

Fig.4 Average CH duration comparison with different HP

Fig. 4 and Fig. 5 compares the cluster performance on CH duration and CM duration under the above mentioned four scenarios when the HP is set in the fixed way or adaptive way. The comparison results show that ACSnH with

LLTAAP plays better performance than ACSnH with fixed HP value, especially for the CM duration – two or three times that of ACSnH(0.2), ACSnH(0.5) or ACSnH(1.0), for it can adjust the HP adaptively by cluster and traffic condition in real time.

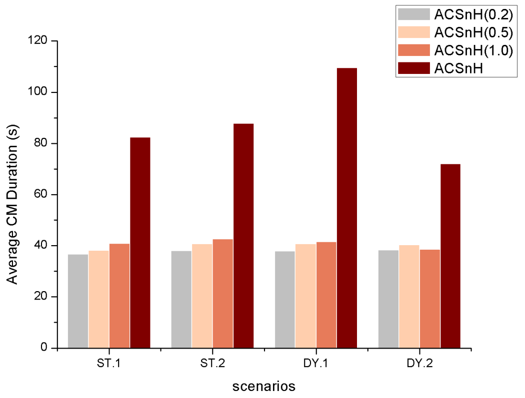

Fig.5 Average CM duration comparison with different HP

Fig. 6 and Fig. 7 shows the comparison of the average CH and CM duration in LID, VMaSC (0.2), VMaSC (1.0) and ACSnH respectively under the four scenarios. In Fig. 6, we can observe that the average CH duration of ACSnH is slightly less than that of VMaSC in scenario ST.1 and DY.2. However, the average CM duration of ACSnH is much longer, two to three times that of VMaSC (0.2) and VMaSC (1.0) in all scenarios, as shown in Fig. 7. It can be also seen that from Fig. 7 the average CM duration of VMaSC (1.0) is longer than that of VMaSC (0.2). The reason lies in its large delay of updating vehicle information caused by its larger HP (1.0). On the other hand, the average CH duration and CM duration both in VMaSC and in ACSnH are always far longer than that in LID, for LID do not select CHs with considering vehicles' mobility as well as do not have the delay mechanism, which means that many vehicles may change their states in a short time once two CHs are close to each other and then away immediately. In summary, the stability of cluster in ACSnH is better than that of cluster both in LID and in VMaSC.

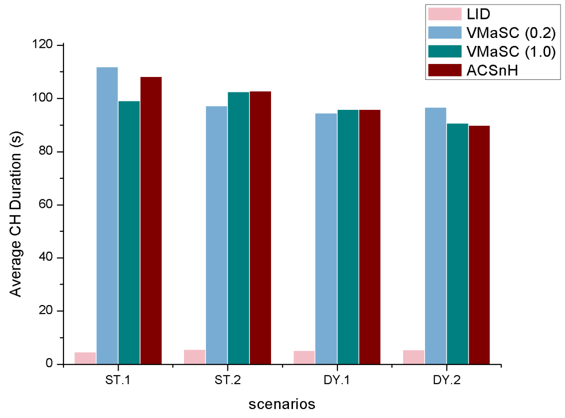

Fig.6 Average CH duration comparison

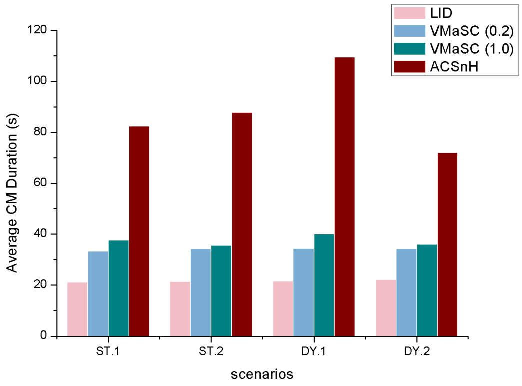

Fig.7 Average CM duration comparison

2) ***Average Number of Cluster:***As every cluster has one and only one CH, the average number of CHs is equal to the average number of clusters. In general, with the premise of guaranteeing the stability of the cluster, constructing fewer clusters means a higher efficiency of the clustering scheme and the data-packet transmission. Fig. 8 and Fig. 9 shows the comparison of the number of clusters under four simulation scenarios. It is obvious that

the number of clusters in ACSnH is always the lowest one, about 20% - 43% lower than that of ACSnH with fixed HP. On the other hand, ACSnH is also always the lowest one when compring with other algorithms showing in Fig. 9, about 24%-44% lower than that of LID, about 44%-55% lower than that of VMaSC (0.2) and about 41% - 64% lower than that of VMaSC (1.0).

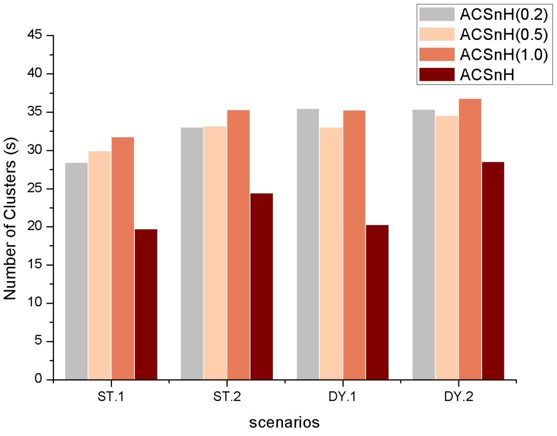

Fig.8 Number of clusters comparison with different HP

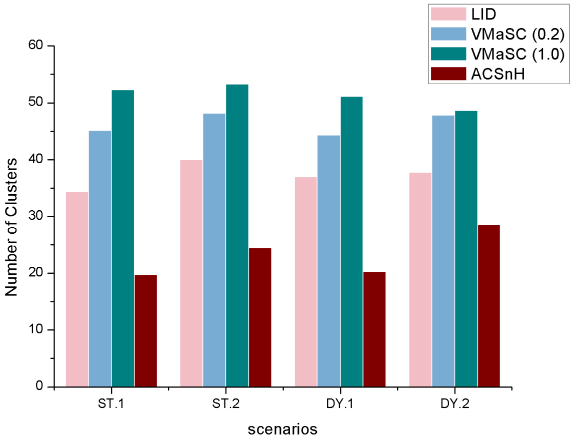

Fig.9 Number of clusters (CHs)

3) ***Average Number of Temporary States:*** Temporary state refers to the state in which vehicle does not belong to any cluster, which is unstable, such as the SE state in VMaSC, and the TS state in ACSnH. LID do not have temporary state because once a CM cannot connect to its CH, it changes its

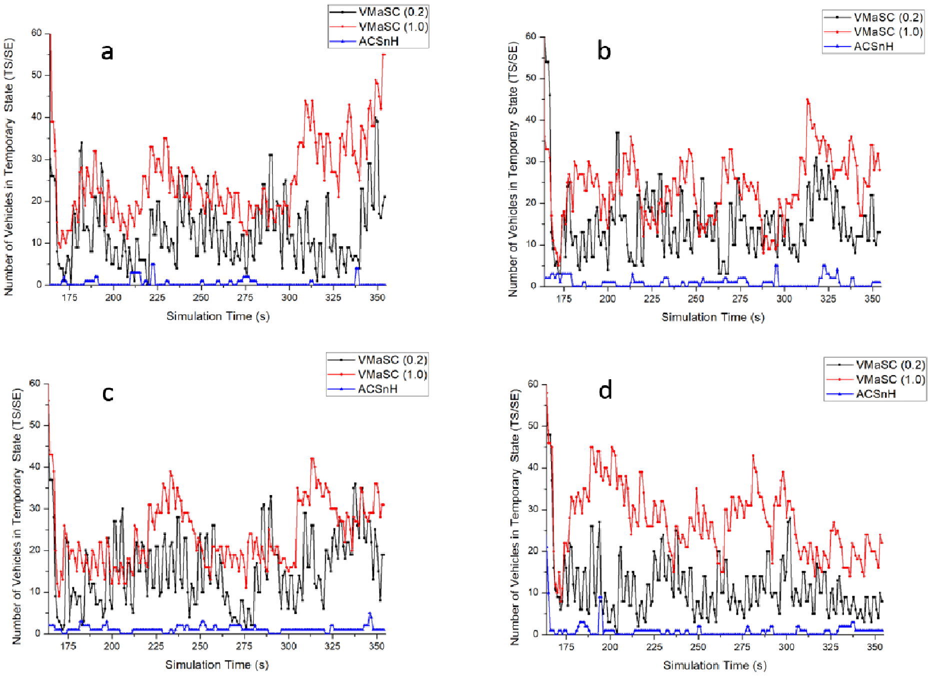

Fig.10 Number of vehicles in the temporary state for VMaSC (0.2), VMaSC (1.0) and ACSnH under

(a) scenario ST.1. (b) scenario ST.2. (c) scenario DY.1. (d) scenario DY.2

state to a new CH. Fig. 10 shows the number of vehicles in the temporary state for VMaSC (0.2), VMaSC (1.0) and ACSnH as a function of simulation time under the respective scenario ST.1, ST.2, DY.1, DY.2. We observe that the number of vehicles in SE state for VMaSC (1.0) is the largest in all cases, resulting from the delay of vehicle information caused by its HP that is largest.The number of vehicles in temporary state of VMaSC is very large, sometimes up to 55 at the same time, far more than ACSnH, whereas the number of vehicles in temporary state of ACSnH is generally less than 2 for most of the time, indicating that the stability of the cluster in ACSnH is much better.

This is because VMaSC uses the handshake system, so a vehicle at the SE state needs to wait for a join response from CH before joining a cluster. If CH does not send a join response (such as the number of CMs of the CH is full, or the join request send from SE is lost which causes CH cannot receive the request), the vehicle will continue to stay in SE state and need to wait for a new period of time to continue to the next action (such as sending a join request to another CH or declaring itself to be an CH). However, in ACSnH, with the RVnH proposed in Section III, the vehicle in TS state does not have to wait for a long time for a feedback from CHs, instead it can immediately determine and decide whether to join an CH or not. If it cannot join a certain CH, the next action will be started immediately, rather than staying for a long time in an unstable temporary state.

4) ***CM Disconnection Rate and Role Change Rate:***CM disconnection rate is the total number of link disconnections between CMs and their CHs per unit time, and role change rate refers to the average number of the changes of states during simulation process per vehicle. Fig. 11 shows the comparison of the CM disconnection rate among LID, VMaSC (0.2), VMaSC (1.0) and ACSnH under the same four scenarios. We can observe that the CM disconnection rate of ACSnH is only about 1/10 to 1/4 of that of LID, is only about 1/5 to 1/2 of that of VMaSC (0.2), and is about 1/4 to 3/5 of that of VMaSC (1.0), which means that ACSnH has the better cluster stability. Fig. 12 shows the comparison of role change rate per vehicle. The role change rate of ACSnH in all cases is always the lowest, showing that the stability of the vehicle states is more better in ACSnH.

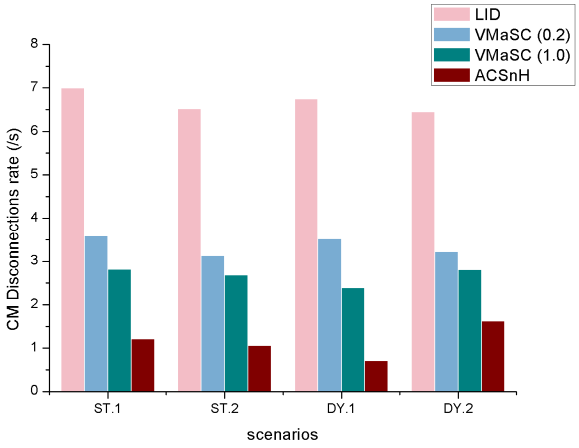

Fig.11 CM disconnection rate (per second)

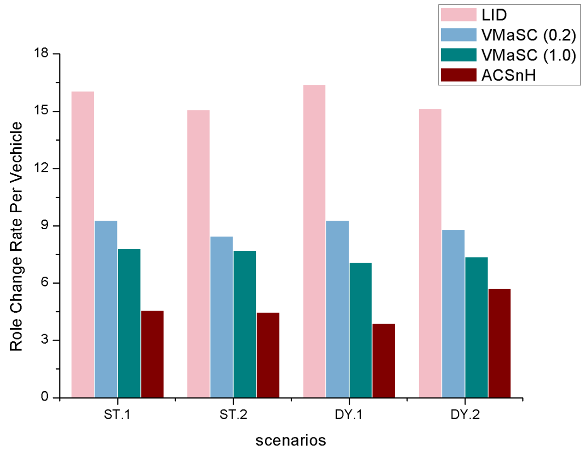

Fig.12 Role change rate (per vehicle)

5) ***Clustering Overhead:***Clustering overhead is defined as the total number of clustering related packets generated in VANETs, including Hello packet, join request, join response, merge request and so on. A good clustering algorithm should be able to minimize the clustering overhead under the premise of ensuring the stability of cluster. Fig. 13 shows the comparison results of clustering overhead between ACSnH used LLTAAP and ACSnH used fixed HP value (0.2 s, 0.5 s and 1.0 s). In LLTAAP, the HP of vehicles in CH or TS is set to 0.2 s, or in CM is set adaptively among 0.2 s, 0.5 s and 1.0 s according to real-time traffic condition. Therefore, the overhead of ACSnH must be between that of ACSnH(1.0) and ACSnH(0.2). It is obvious that the overhead of ACsnH is less than ACSnH(0.5) and much less than ACSnH(0.2). So the goal of reducing clustering overhead is achieved on the premise of guaranteeing cluster stability according to the simulation results that showing better performance on cluster stability showed above. Fig. 14 shows the clustering overhead with the algorithm of LID, VMaSC (0.2), VMaSC (0.5), VMaSC (1.0) and ACSnH. We can observe that the clustering overhead of ACSnH is lower than that of LID, VMaSC (0.2) and VMaSC (0.5). Though comparing with ACSnH(1.0) and VMaSC (1.0), ACSnH has a relative larger overhead, it performs much better on the clustering stability according to the analysis in Section III, and the simulation results in Fig. 4 to Fig. 12 and Table. VII.

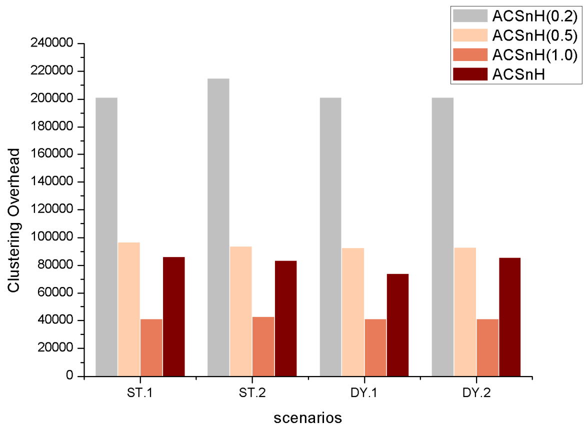

Fig.13 Clustering overhead comparison with different HP

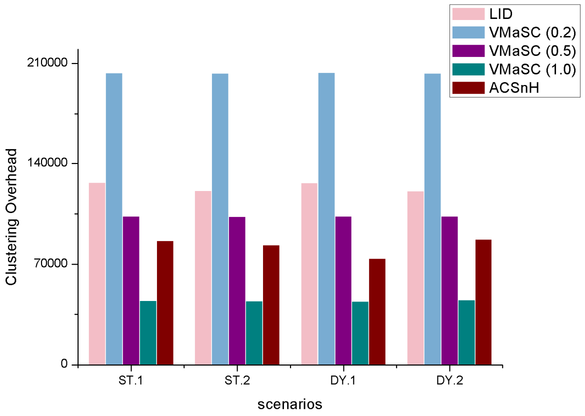

Fig.14 Clustering overhead

6) ***CM re-Clustering Delay:***CM re-clustering delay represents the time interval of a CM from disconnecting with its current CH to successfully joining another cluster. Table. VII shows the average CM re-clustering delay of three clustering scheme under different scenarios. As shown in table, the re-clustering delay of VMaSC (0.2) is about 3 to 4 seconds, and the delay of VMaSC (1.0) is about 7 to 8 seconds, while the delay of ACSnH

is always lower than 0.5 seconds in these scenarios, which is far less than VMaSC.

TABLE VII
AVERAGE CM RE-CLUSTERING DELAY (S)

Scenarios	VMaSC(0.2)	VMaSC(1.0)	ACSnH
ST.1	3.27939	7.91442	0.29587
ST.2	3.36345	7.29846	0.33473
DY.1	3.85878	7.62122	0.21058
DY.2	3.01378	8.08982	0.24817

V. CONCLUSIONS

In this paper, we proposed an adaptive clustering scheme with a non-handshake size control mechanism in VANETs. In our proposed ACSnH, the Hello Packet Period (LLTAAP) is assigned to be adaptable, which can greatly reduce the overhead of clustering management and the probability of packet collision by adaptively adjusting Hello packet period according to vehicle state and link lifetime. And the cluster head of a vehicle is selected by comprehensively weighting the contributions from the relative distance, relative velocity and link lifetime, with which the stability of the clustering is greatly enhanced. Furthermore, the formation process of clustering is optimized by a non-handshake size control mechanism which can directly control the size of CM per cluster without the help of cluster head, which highly improves the cluster stability and formation efficiency, and obviously reduces the clustering overhead and re-clustering delay. We compare our ACSnH with LID and VMaSC under different scenarios and find that our ACSnH can realize an outstanding performance in terms of the aspects mentioned above.

In future work, we aim to carry out more experiments and improve ACSnH to better adapt to complex urban traffic scenarios, as well as extend it for data dissemination for applications in VANETs.

REFERENCES

[1] M. M. Zanjireh, H. Larijani, A survey on centralised and distributed clustering routing algorithms for wsns, in: Vehicular Technology Conference(VTC Spring), 2015 IEEE 81st, IEEE, 2015, pp. 1-6.

[2] H. Moustafa, Y. Zhang, Vehicular networks: techniques, standards, and applications, Auerbach publications, 2009.

[3] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, T. Weil, Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions, IEEE communications surveys & tutorials 13 (4) (2011) 584-616.

[4] S. Tayal, M. R. Triphathi, Vanet-challenges in selection of vehicular mobility model, in: 2012 Second International Conference on Advanced Computing & Communication Technologies, IEEE, 2012, pp. 231-235.

[5] R. S. Bali, N. Kumar, J. J. Rodrigues, Clustering in vehicular ad hoc networks: taxonomy, challenges and solutions, Vehicular communications 1 (3) (2014) 134-152.

[6] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, V. V`eque, A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (vanets), Vehicular Communications 9 (2017) 233-241.

[7] F. Chiti, R. Fantacci, G. Rigazzi, A mobility driven joint clustering and relay selection for ieee 802.11 p/wave vehicular networks, in: Communications (ICC), 2014 IEEE International Conference on, IEEE, 2014, pp. 348-353.

[8] Y. Guenter, B. Wiegel, H. P. Gro?mann, Medium access concept for vanets based on clustering, in: Vehicular Technology Conference, 2007. VTC 2007 Fall. 2007 IEEE 66th, IEEE, 2007, pp. 2189-2193.

[9] O. Kayis, T. Acarman, Clustering formation for inter-vehicle communication, in: Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, 2007, pp. 636-641.

[10] R. Santos, R. Edwards, A. Edwards, Cluster-based location routing algorithm for inter-vehicle communication, in: Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th, Vol. 2, IEEE, 2004, pp. 914-918.

[11] A. Ephremides, J. E. Wieselthier, D. J. Baker, A design concept for reliable mobile radio networks with frequency hopping signaling, Proceedings of the IEEE 75 (1) (1987) 56-73.

[12] S. Ucar, S. C. Ergen, O. Ozkasap, Multihop-cluster-based ieee 802.11 p and lte hybrid architecture for vanet safety message dissemination, IEEE Transactions on Vehicular Technology 65 (4) (2016) 2621-2636.

[13] P. Basu, N. Khan, T. D. Little, A mobility based metric for clustering in mobile ad hoc networks, in: Distributed computing systems workshop, 2001 international conference on, IEEE, 2001, pp. 413-418.

[14] M. Ni, Z. Zhong, D. Zhao, Mpbc: A mobility prediction-based clustering scheme for ad hoc networks, IEEE Transactions on Vehicular Technology 60 (9) (2011) 4549-4559.

[15] M. Chatterjee, S. K. Das, D. Turgut, Wca: A weighted clustering algorithm for mobile ad hoc networks, Cluster computing 5 (2) (2002) 193 204.

[16] R. Purtoosi, H. Taheri, A. Mohammadi, F. Foroozan, A light-weight contention-based clustering algorithm for wireless ad hoc networks, in: Computer and Information Technology, 2004. CIT’04. The Fourth International Conference on, IEEE, 2004, pp. 627-632.

[17] A. Akbari, M. Soruri, S. V. Jalali, Survey of stable clustering for mobile ad hoc networks, in: Machine Vision, 2009. ICMV’09. Second International Conference on, IEEE, 2009, pp. 3-7.

[18] E. Souza, I. Nikolaidis, P. Gburzynski, A new aggregate local mobility (alm) clustering algorithm for vanets, in: Communications (ICC), 2010 IEEE International Conference on, IEEE, 2010, pp. 1-5.

[19] Z. Zhang, A. Boukerche, R. Pazzi, A novel multi-hop clustering scheme for vehicular ad-hoc networks, in: Proceedings of the 9th ACM international symposium on Mobility management and wireless access, ACM, 2011, pp. 19-26.

[20] E. D.Krajzewicz, M. Behrisch, Sumo - simulation of urban mobility, http://sumo.sourceforge.net/ (2011).

[21] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, V. V`eque, A unified framework of clustering approach in vehicular ad hoc networks, IEEE Transactions on Intelligent Transportation Systems 19 (5) (2018) 1401-1414.

[22] A. Daeinabi, A. G. P. Rahbar, A. Khademzadeh, Vwca: An efficient clustering algorithm in vehicular ad hoc networks, Journal of Network and Computer Applications 34 (1) (2011) 207-222.

ACKNOWLEDGMENT

This work was financially supported by NationalHigh Technology Research and DevelopmentProgram of China (863Program) (Grant No.2015AA015501), andApplied ScienceTechnology Development and Major ScientificTechnological Achievements TransformationProject of GuangDong Province, China.

[23] M. M. C. Morales, C. S. Hong, Y.-C. Bang, An adaptable mobility-aware clustering algorithm in vehicular networks, in: Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific, IEEE, 2011, pp. 1-6.

[24] B. K. A. e. a. Ghafoor K Z, Lloret J, Beaconing approaches in vehicular ad hoc networks: A survey, Wireless Personal Communications 73 (3) (2013) 885-912.

[25] F. X. A. K. M. S. Syed Adeel Ali Shah, Ejaz Ahmed, R. M. Noor, Adaptive beaconing approaches for vehicular ad hoc networks: A survey, IEEE Systems Journal 12 (2) (2018) 1263-1277.

[26] C. Cooper, D. Franklin, M. Ros, F. Safaei, M. Abolhasan, A comparative survey of vanet clustering techniques, IEEE Communications Surveys & Tutorials 19 (1) (2017) 657-681.

[27] S.-S. Wang, Y.-S. Lin, Passcar: A passive clustering aided routing protocol for vehicular ad hoc networks, Computer Communications 36 (2 (2013) 170-179.

[28] W. Li, A. Tizghadam, A. Leon-Garcia, Robust clustering for connected vehicles using local network criticality, in: Communications (ICC), 2012 IEEE International Conference on, IEEE, 2012, pp. 7157-7161.

[29] Network simulator-ns-3, http://www.nsnam.org/ (2017).

[30] M. Lacage, T. R. Henderson, Yet another network simulator, in: Proceeding from the 2006 workshop on ns-2: the IP network simulator, ACM, 2006, p. 12.