

Research Article **Double-Head Clustering for Resilient VANETs**

Ghada H. Alsuhli, Ahmed Khattab 💿, and Yasmine A. Fahmy

Electronics and Electrical Communications Engineering Department, Cairo University, Giza, Egypt

Correspondence should be addressed to Ahmed Khattab; akhattab@ieee.org

Received 27 September 2018; Revised 9 December 2018; Accepted 4 February 2019; Published 3 March 2019

Academic Editor: Nathalie Mitton

Copyright © 2019 Ghada H. Alsuhli et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Scalability and the highly dynamic topology of Vehicular Ad Hoc Networks (VANETs) are the biggest challenges that slow the roll-out of such a promising technology. Adopting an effective VANET clustering algorithm can tackle these issues in addition to benefiting routing, security and media access management. In this paper, we propose a general-purpose resilient double-head clustering (DHC) algorithm for VANET. Our proposed approach is a mobility-based clustering algorithm that exploits the most relevant mobility metrics such as vehicle speed, position, and direction, in addition to other metrics related to the communication link quality such as the link expiration time (LET) and the signal-to-noise ratio (SNR). The proposed algorithm has enhanced performance and stability features, especially during the cluster maintenance phase, through a set of procedures developed to achieve these objectives. An extensive evaluation methodology is followed to validate DHC and compare its performance with another algorithm using different existing and newly proposed evaluation metrics. These metrics are analyzed under various mobility scenarios, vehicle densities, and radio channel models such as log-normal shadowing and two-ray ground loss with and without Nakagami-m fading model. The proposed algorithm DHC has proven its ability to be more stable and efficient under different simulation scenarios.

1. Introduction

Vehicular Ad Hoc Networks (VANETs) are special types of Mobile Ad Hoc Networks (MANETs), in which the mobile nodes are vehicles moving on the road [1]. The progressive development of VANET technology paves the way for the implementation of Intelligent Transportation System (ITS) abundant applications. These applications may be safetyoriented applications to increase traffic safety and reduce road accident fatalities or nonsafety-oriented applications that aim to manage the traffic in addition to providing passengers with infotainment and comfort [2].

In VANETs, each vehicle can communicate with other vehicles as they are equipped with On-Board Units (OBUs). This kind of communications, which is called Vehicle-to-Vehicle (V2V), is expected to be essential for most ITS applications due to its low cost and availability [3]. At the same time, vehicles may be able to communicate with installed infrastructure, e.g., Road Side Units (RSUs), using Vehicle-to-Infrastructure (V2I) communications.

Although VANETs suffer from many challenging issues such as scalability, high node mobility, and limited spectral

bandwidth, it has special characteristics, like the constrained vehicles mobility patterns and unlimited power and calculation capabilities that can be exploited to mitigate the effect of these challenges [1]. One form of this exploitation is the use of VANET clustering.

Clustering in a VANET aims to convert the network structure from being flat into being hierarchical by dividing the network into virtual groups of vehicles called clusters [4]. Each cluster has a leader, usually donated by the cluster head (CH), and a number of cluster members (CMs). The CH acts as an infrastructure or an access point of the cluster. Then, instead of dealing with a pure ad hoc network, a virtualinfrastructure-based network is presented without the need of expensive physical infrastructure deployment. Moreover, when the clustering is done taking into consideration the mobility information of the vehicles, the topology inside the cluster becomes relatively static. Hence, the highly dynamic topology problem becomes less serious [5]. This kind of clustering is usually called mobility-based clustering [6].

A wide range of protocols and applications rely on VANET clustering to achieve their goals. For instance, clustering is exploited by many security applications to address certain security threats and detect intrusion [7, 8]. Furthermore, many VANET routing protocols are proposed based on clustering to mitigate scalability problems [9–11]. Since clustering allows a better utilization of the network resources and scheduling of medium access, various MAC protocols use the CH of each cluster to coordinate the medium access among the cluster members [12, 13]. Moreover, VANET safety applications benefit from clustering to disseminate safety messages by providing an effective broadcast mechanism [4, 14], in addition to many other applications such as topology discovery and QoS assurance [15].

Over the last three decades, a large number of clustering approaches have been proposed in the literatures. However, developing and validating a clustering algorithm suitable for all conditions and scenarios is one of the challenges that are still open to researchers in VANET field [15].

Our main contributions in this paper are as follows: First, we propose a new resilient clustering algorithm with a focus on increasing the stability of clusters and decreasing the number of clusters in the network under different conditions and scenarios. Our proposed technique has some distinct features as follows:

(i) Two functioning CHs are selected in each cluster to avoid unnecessary reclustering when a CM instantaneously loses the link with the primary CH.

(ii) A new multimetric CH selection scheme is presented. This scheme takes into consideration the different selection metrics that increase cluster stability and clustering efficiency.

(iii) We introduce a cluster replacement (CR) scheme in which a CM can change the current cluster when the link with the corresponding CH is about to be disconnected.

(iv) We propose an All-Member-Interests-based Merging (AMIM) scheme in which the decision of merging is taken based on the benefits of all members not only the CH.

(v) We propose a solution of the faulty affiliation problem, other than affiliation handshaking which adds extra packet overhead.

Second, we propose an evaluation approach in which various evaluation metrics, vehicle densities, simulation environments, mobility traffic scenarios, and channel models are considered for more accurate assessment of a clustering algorithm.

The remainder of this paper is organized as follows. In Section 2, we describe the related literature of VANET clustering approaches and compare some of the used evaluation methodologies. The system model and assumptions are presented in Section 3. Section 4 explains the details of the proposed clustering algorithm. The performance evaluation approach and simulation results are described in Section 5. Finally, Section 6 provides our conclusions and future directions.

2. Related Work

2.1. VANET Clustering Algorithms. Despite the great diversity of clustering algorithms in the literature, they almost share the same procedural steps [15]. The fundamental clustering procedures are selecting a cluster leadership and adopting some schemes to maintain the formed clusters.

The existing clustering approaches differ in cluster leadership in terms of the number of CHs and the way they are selected. Most of the schemes suffice with one CH for each cluster such as in [14, 16, 17]. Despite the simplicity of this approach, reclustering will be an inevitable once the CH resigns or loses its suitability to continue as a CH. To tackle this, some algorithms, for example, AMACAD [18], SBCA [19], FLBA [20], and SCalE [21], suggested having a backup CH that takes the role of the current CH under certain circumstances. The common feature between these works is that the alternative CH has no role in the cluster unless the current CH leaves the cluster. In contrast, the second head used in DHC is always functioning to respond to any CM that temporary loses its connection with the primary CH.

In [3], each node in the network should be connected with two clusters (or CHs), which has a similar principle to doublehead clustering. However, although this approach enhances the connectivity too, the number of CHs is considerably large, and the cluster boundaries become unclear. In the same context, a multihead clustering was introduced in [22]. In this scheme, each cluster has a single master CH and several slave CHs that are uniformly distributed in the cluster area. It was claimed that the presence of multiple CHs in a cluster enhanced the stability and increased the lifetime of the cluster. In spite of partially sharing the concept of creating a cluster with more than one CH with [22], our algorithm differs in the way the secondary CH is selected and corporates with the primary CH.

Different CH selection methods were used in the literature. The popular selection methods can be classified into competition-based and comparison-based CH selection methods. UFC and TB [16, 23] are some examples of clustering algorithm that use competition to select the CH. A backoff timer is set by each vehicle separately. The timer value is usually calculated depending on some metrics [16, 23] or is set randomly as also proposed in [16]. The vehicle that firstly broadcasts a CH announcement wins the competition and becomes the CH. In comparison-based method, each node either compares the received suitability indexes from different one-hop neighbors as in SCalE [21] and APROVE [24] or calculates and compares neighbors' indexes by itself independently as in RMAC [25] and VMASC [14]. According to the comparison result, the node may report itself as the CH or sends a request to join the most suitable neighbor. Our approach uses a comparison-based selection due to its simplicity and efficiency to minimize the number of formed clusters with relatively smaller number of required control packets.

The metrics, on which the qualified CH selection depends, differ from an algorithm to another. For instance, the authors in [26] suggested evaluating the suitability of the CH using a weighted sum of the selection metrics such as speed, number of neighbors, and trust degree. In [27], speed, position, and acceleration were considered for CH selection to make the cluster more stable. A weighted sum of these normalized metrics was used to select the best CH. On the other hand, the CH is selected depending on certain flags in [28]. These flags are set based on the direction, speed, destination, and link quality with the neighbors. In [29], a

Algorithm	Network Simulator	Traffic Mobility		Simulation Environment			
		Different scenarios	Different densities	Realistic model	Synthetic	realistic	Loss Model
UFC [16]	NS2	\checkmark	×	SUMO	\checkmark	×	Two-Ray Ground
VMASC [14]	NS3	\checkmark	×	SUMO	\checkmark	×	Friis
CBL [32]	MATLAB, OPNET	\checkmark	\checkmark	SUMO	\checkmark	×	-
ADC [38]	NS2	×	×	SUMO	×	\checkmark	-
EnLOSC [36]	MATLAB	×	×	×	\checkmark	×	FSPL
ADAPTIVE [34]	OMNeT++	×	\checkmark	SUMO	\checkmark	×	-
ANTSC [33]	NS2	\checkmark	×	SUMO	\checkmark	×	-
Intersection-Based [17]	OMNET	×	×	SUMO	\checkmark	×	-
DMCNF [37]	NS2	×	×	VanetMobiSim	¥	×	Two-Ray Ground

TABLE 1: Verification methodologies of some of recent clustering algorithms.

nonlinear objective function, of position, node degree, and velocity, is calculated for each vehicle. This function is used by the serving road side unit to select the CH.

To prolong the life of the formed clusters, despite the dynamic topology of VANET, many maintenance procedures were proposed. The most commonly used maintenance strategies included cluster information updating and cluster merging [15]. Most of the clustering algorithms share the same information updating methods during the maintenance phase by exchanging and storing up-to-date information periodically between the CH and its CMs.

Since minimizing the number of clusters in the system is an aim, many cluster merging schemes were proposed. The decision of merging is usually taken by a CH when (1) another stable CH stays in its range for a minimum period of time as in [16, 30], (2) the candidate CH is more eligible than the current CH as in [14, 31], or (3) the target stable CH has more members as in [23]. Unlike such works, the DHC merging decision is taken via our AMIM scheme that is based on all members interest not only the benefit of CH. In addition to the previously mentioned maintenance procedures, our algorithm also suggests adding a cluster replacement scheme that will be discussed later.

2.2. Performance Evaluation Methodologies. To speed up the roll-out of VANET technology, performance evaluation methodologies should be paid a great attention. As long as the practical experiment of VANET protocols is very difficult in real environments, the validation of any clustering algorithm must be done using an analogous simulation framework. This framework should include suitable network and mobility simulators, different simulation environments, various channel models, and proper performance evaluation metrics.

Table 1 shows a comparison between the validation methodologies used to evaluate some of the recent clustering algorithms. One of the VANET simulation key components is using realistic mobility models, different vehicle densities, and mobility scenarios. This is to ensure that the insights driven from the simulation results will be applicable to a great extent after the real deployment. Most of the algorithms listed in Table 1 are evaluated using realistic mobility model (provided by SUMO). In addition, many of the algorithms simulated different mobility scenarios such as [14, 16, 32, 33] and different vehicle densities as in [32, 34]. On the other hand, Table 1 shows that simulating realistic roads, taken from a real map, is rare despite its simplicity. To avoid modeling the vehicular channel which is very challenging [35], the packet loss rate may be used to evaluate the performance of the cluster algorithms. Nevertheless, [15] emphasized the importance of considering a proper channel model before validating any clustering algorithm since the performance of clustering is very sensitive to signal propagation properties. For instance, [14, 36] used the Free Space Propagation Loss (FSPL) simplistic channel model in their verification, while [16, 37] used the two-ray ground model. However, none of the existing works neither considered different channel models nor investigated the impact of the channel model on the algorithm behavior.

In our work, we aim to evaluate DHC using a realistic mobility model with different mobility scenarios and vehicles densities. In addition, our simulation will be performed on both realistic and synthetic roads. Moreover, to the best of our knowledge, none of the clustering algorithms was verified under different propagation loss and fading models as we shall do.

3. System Model and Assumptions

Our proposed algorithm considers V2V communication. Each vehicle communicates and exchanges the information with other vehicles, by either broadcasting the message to all one-hop neighbors or unicasting it to a specific destination within its transmission range. Any additional infrastructure such as RSUs is not needed for our approach. We also assume that each vehicle is equipped with a Dedicated

FIGURE 1: Clustering in VANET topology.

Short-Range Communication (DSRC) wireless unit, a Global Positioning System (GPS), and a Processing unit (PU). The DSRC transceiver allows the vehicle to communicate with other vehicles in the network and exchange the information. The messages that are required to perform the clustering process are sent using the DSRC control channel (CCH) that is more likely to have wider range than the service channels SCHs [23]. This assumption imposed additional complications on our clustering algorithm, since the vehicles may receive from other vehicles on the CCH but cannot connect to the SCHs. Then, only vehicles that can communicate using SCHs can be in the same cluster. The vehicle uses the GPS, or any other positioning system, to obtain its real-time position and velocity which are necessary to be measured and exchanged periodically. The processing unit is needed to run the clustering algorithm on each vehicle separately in a distributed manner. Furthermore, a unique identifier (ID) is allocated to each vehicle in the network. Our approach can be used for both highway and urban environments. The roads, on which the vehicles move, may have multiple lanes and two directions. It is also not necessarily to be straight roads, but with no intersections as the intersection turning prediction is not presented yet in our scheme.

4. Proposed Double-Head Clustering

Our proposed double-head clustering (DHC) scheme is a distributed algorithm. Each vehicle runs the algorithm independent of other vehicles in the network. The only information that a vehicle knows about other vehicles is that conveyed by the exchanged messages. These messages will be discussed in detail while we are describing the algorithm in this section.

Figure 1 shows an example of a VANET topology using DHC clustering. Several clusters are presented with one or two CHs in each cluster, the primary CH and the secondary CH (SCH). The primary CH, donated for simplicity by CH, leads various number of CMs inside each cluster. During the clustering process, the vehicle goes through a number of states and different clustering procedure steps that will be described in the following subsections. The notations used in this paper are listed in Table 2.

4.1. Vehicle States. When DHC algorithm is applied to vehicles in a VANET, each vehicle operates in one of the following four states:

TABLE 2: Used notations.

Description
Vehicle ID: $VID_i = i$.
Cluster ID of vehicle <i>i</i> .
2D position components of vehicle <i>i</i> .
2D velocity components of vehicle <i>i</i> .
Eligibility of vehicle <i>i</i> .
Eligibility of the current vehicle.
Number of consistency neighbors of vehicle <i>i</i> .
Number of consistency neighbors of the current vehicle.
Number of members of the cluster head <i>i</i> .
Number of members of the current cluster head.
Maximum allowed number of members in a cluster.
Node (or vehicle) with $= i$.
UV vehicle default cluster ID.
Consistency speed threshold.
Vehicle i consistent neighbor table.
Receiving time of the last received packet from n_i .
SNR of the last received packet from n_i .
LET of the last received packet from n_i .
Direction angle of vehicle i.
Direction matching indicator for vehicles i, j.
Service channel Tx range.
SCH ID of CH with ID= <i>i</i> .

(1) Unclustered Vehicle (UV) state: before starting clustering process, all vehicles begin with a UV state.

(2) CH state: a vehicle in CH state is supposed to be the most eligible vehicle among its one-hop neighbors to act as a leader. The CH takes some leading responsibilities depending on the purpose of clustering.

(3) SCH state: the vehicle in this state acts as a mirror of the CH in a cluster. It operates as a backup of the CH in the case of temporarily losing the connection between the CH and any CM.

(4) CM state: when a vehicle in UV state joins a cluster, it becomes a member and turns into CM state.

The transition from one state to another is triggered by some events as illustrated in Figure 2. We will explain these events in detail while explaining our proposed clustering algorithm procedure in the next subsection.

4.2. Clustering Procedure. Vehicles seeking to be clustered will follow some or all of the following steps.

4.2.1. Surrounding Exploration. Initially, all vehicles join the road in the UV state. When any vehicle decides to join the VANET network, it begins to send a periodic Hello message to advertise its existence and to share its mobility information with the surrounding neighbors. Figure 3 shows the information carried on the Hello message broadcasted each T_{Hello} period of time. This information includes the vehicle ID (VID), cluster ID (CID), two-dimensional position (x, y), two-dimensional velocity (v_x, v_y) , the number of consistent



FIGURE 2: Vehicle state diagram.



FIGURE 3: Hello message and CNT table content.

neighbors N, and vehicle eligibility to be the CH E. The initial values of the cluster ID and eligibility are set to a unique identifier CID_{UV} and zero, respectively.

Once a Hello message is received, the neighbor is classified as either a consistent or inconsistent neighbor. The consistent neighbors CNs must have the same road direction with a speed difference Δs less than a predefined limit Δs_{max} . The direction angle *a* of node n_i is calculated as in (1) depending on the velocity components. Equation (2) shows that any two vehicles, n_i and n_j , are supposed to have the same direction if the direction angle between them is less than θ .

$$\hat{a}(n_i) = \arctan\left(\frac{v_{y_i}}{v_{x_i}}\right),$$
 (1)

$$Dir_{ij} = \begin{cases} +1 & \text{if } \left| \hat{a} \left(n_j \right) - \hat{a} \left(n_i \right) \right| < \theta \\ -1 & \text{otherwise,} \end{cases}$$
(2)

where Dir_{ij} is the direction matching indicator. In addition, the distance difference Δd_{ij} between any two consistent neighbors n_i , n_j must be less than the service channel range *SCHR*. The CN information is stored in the Consistent Neighbors Table (CNT) alongside the receiving time (RT_i), the signal-to noise ratio (SNR_{ci}) of the received packet and the expected link expiration time (LET_{ci}) with this neighbor. The content of the CNT table is illustrated in Figure 3. After the exploration time T_E , the node should have received enough information to build an image about the neighborhood and becomes ready to select the best CH.

4.2.2. Eligible CH Selection. During the exploration time T_E , each vehicle calculates and advertises its eligibility periodically within Hello messages. When T_E expires, a vehicle automatically enters the CH selection step in which it compares its eligibility with consistent neighbors' eligibilities. The vehicle that has the highest E_c announces itself as a CH by sending a CHHello message (shown in Figure 5(a)) immediately and then periodically each T_{Hello} . A vehicle that has a lower E_c than other UV vehicles in its CNT will keep waiting until it receives a CHHello from a CN neighbor. The received CHHello message is stored in a Cluster Head Table (CHT) that is similar to CNT table.

It may happen that a vehicle n_c blocks waiting in order to receive a CH announcement from a more eligible vehicle n_i . At the same time, n_i is also still waiting for another vehicle n_j , and so on. This appears when a number of vehicles accidentally lined up with gradual eligibility. Even if this situation is rare to occur, a timeout $T_{timeout}$ is used in DHC to avoid such a blocking scenario. Advertising the eligibility within the Hello message, instead of using a separate packet, aims to reduce the control overhead of the proposed algorithm. Algorithm 1 illustrates the pseudocode of our CH selection and announcement algorithm.

Each node estimates its eligibility depending on a number of metrics as follows.

Popularity: since the vehicle can join a cluster that has a consistent CH only, the node with more CNs tends to form a larger cluster if it becomes a CH. To reduce the number of formed clusters in the system, we consider the number of consistent neighbor N_c as one of the CH selection metrics.

Relative position: the vehicle that is closer to the mean position of its CNs should have better opportunity to become the CH.

Relative speed: in order to increase the stability of a cluster, the CH should have the closest speed to the mean speed of its CMs.

Average signal-to-noise ratio SNR: SNR of the received message determines the communication link quality between two vehicles. The vehicle which has better links with its consistent neighbors performs better when it becomes a CH.

Average link expiration time LET: it is the average communication link lifetime between current vehicle and its consistent neighbors.

Equation (3) calculates LET between n_c and a consistent neighbor n_i assuming that the street width can be ignored. The calculation is then done for one coordinator that is parallel to street direction.

$$LET_{ci} = \frac{SCHR - \Delta d_{ci} \cdot \text{sign} \left(\Delta s_{ci}\right)}{\left|\Delta s_{ci}\right|} \quad \forall n_i \in CNT_c, \quad (3)$$

where *SCHR* is the service channel range, Δd_{ci} the distance between n_c and n_i , Δs_{ci} is the speed difference between n_c and n_i , and sign(.) is the sign function, i.e., sign(Δs_{ci}) = 1 when n_c is faster than n_i and -1 otherwise.

ALGORITHM 1: CH selection and announcement.

The suitability of the current vehicle to be the CH is specified by its eligibility value, E_c , which is computed in (4) as a weighted sum of the considered metrics.

$$E_{c} = w_{d} \cdot f\left(\Delta d_{cm}\right) + w_{s} \cdot f\left(\Delta s_{cm}\right) + w_{N} \cdot g\left(N_{c}\right) + w_{LET} \cdot g\left(LET_{c}\right) + w_{SNR} \cdot g\left(SNR_{c}\right),$$
(4)

where Δs_{cm} is the speed difference between n_c and its consistent neighbors' mean speed, Δd_{cm} is the distance between n_c and the mean position of its consistent neighbors, $LET_c = (1/N_c) \sum_{i=1}^{N_c} LET_{ci}$, $SNR_c = (1/N_c) \sum_{i=1}^{N_c} SNR_{ci}$, and the constants w_d , w_s , w_N , w_{LET} , and w_{SNR} are the weights of corresponding metrics which indicate the impact of each metric on E_c . The values of these weights need to be optimized depending on the scenario using machine learning techniques. However, we defer the discussion of such optimization for a sequel paper. Since the values of these metrics, $f(\cdot)$ and $g(\cdot)$, are used to extract a score between 0 and 1 from each metric separately. Then, assuming that all weights are ones, E_c value ranges between 0 and 5. Such that the higher the E_c is, the more qualified the vehicle is to be a CH. The proposed normalization functions are defined by the formulas in

$$f(z) = e^{-(z/\sqrt{2}\sigma_z)^2},$$
 (5)

$$g(u) = \frac{1}{1 + e^{\alpha_u(\mu_u - u)}},$$
 (6)

where σ_z , μ_u , α_u are either predefined constants in case $u = SNR_c$ and $u = LET_c$ or dynamic variables that are calculated in real-time in case $u = N_c$, $z = \Delta d_{cm}$ and z = Δs_{cm} . Because each vehicle has information about neighbor positions, speeds, and number of CNs, this information is utilized to calculate the standard deviation of the neighbor positions from the mean position, the standard deviation of the neighbor speeds from the mean speed, the standard deviation, and the mean of N_i , which represent $\sigma_{\Delta d_{cm}}$, $\sigma_{\Delta s_{cm}}$, α_{N_c} , and μ_{N_c} , respectively. The reasons behind using these normalization functions are as follows: $q(\cdot)$ gives a higher score when N_c , SNR_c , and LET_c have higher values. At the same time, the score increasing rate should be slightly reduced when the metric value becomes very high; as these high values have insignificant impact on clustering performance (such as when the value of the LET tends to infinity), this function is illustrated in Figure 4(b), whereas we used $f(\cdot)$ function since the score achieved by relative position or speed metrics increases as the difference from the mean decreases, as Figure 4(a) shows. For example, the metric score equals 1 when this difference approaches to 0.

4.2.3. Joining a Cluster. Once a UV vehicle in the cluster head selection step receives a CHHello message, it checks whether (1) it is from a CN neighbor or (2) it is from a more eligible vehicle. By fulfilling these two conditions, the vehicle unicasts a join request JoinReq message (shown in Figure 5(b)) to the corresponding CH and changes its state to a CM. On receiving a JoinReq message, the CH in turn checks the consistency of the sender to be added to its CMs Table (CMT). The CH also makes sure that the number of CMs does not exceed the maximum limit N_{max} before adding any vehicle. Algorithm 2 shows the pseudocode of the joining procedure when the current vehicle is either a CH or a UV vehicle.

The proposed approach does not contain any affiliation handshaking to avoid adding an extra packet overhead. However, the lack of handshaking usually leads to a known problem called the faulty affiliation [15]. The faulty affiliation occurs in the absence of a joining acknowledgment after the j^{th} CH receives a JoinReq message from a vehicle n_i . If a JoinReq packet is lost or rejected by the j^{th} CH, the sender vehicle n_i will mistakenly believe that its affiliation process was successful.

To avoid join request rejection, the vehicle wishing to join a cluster must make sure that its request will be accepted by examining the j^{th} CH consistency and its current number of members Nm_j before requesting to join. To overcome the faulty affiliation problem when the sent JoinReq is lost, the j^{th} CH will consider the received Hello message from a nonmember vehicle n_i with $CID_i = j$ as a join request, and n_i will be added to its CMT if the affiliation acceptance conditions were satisfied. As a result, the faulty affiliation problem will be solved after at most one T_{Hello} period of time without any complication.

4.2.4. Cluster Maintenance. After selecting the cluster heads and forming the clusters, a set of procedures are proposed to maintain a relatively stable topology inside the cluster and preserve communication links quality between a CH and its CMs. Some of these procedures are of CH responsibilities



FIGURE 4: Proposed normalization functions for eligibility score calculation.



FIGURE 5: CHHello and JoinReq messages content.



ALGORITHM 2: Joining a cluster.

```
(1) if (n_c.state == CH \&\& n_c receives
          CHHello from n_i)
       if (|\Delta s_{ci}| \le \Delta s_{max} \&\& Dir_{ci} == 1
(2)
             && \Delta d_{ci} \leq SCHR) \forall n_i \in (CMT_c \bigcup n_c)
(3)
           if (Nm_i + Nm_c + 1 \le N_{max} \&\& E_i \le E_c)
             n_c.state \leftarrow CM
(4)
(5)
              CH_c \leftarrow n_i
(6)
              CID_{c} \leftarrow VID_{i}
(7)
              send JoinReq to n_i
(10)
             end if
(11)
        end if
(12) end if
(13) if (n_c.state == CM)
        if (n_c \text{ receives Hello from } n_i \&\&
(14)
               VID_i == CID_c)
           CID_{c} \leftarrow CID_{i}
(15)
          send JoinReq to n<sub>CIDi</sub>
(16)
(17)
          erase n_i from CHT_c
(18)
         end if
(19) end if
```

ALGORITHM 3: AMIM merging.

while the others are joint tasks between the CH and CMs.

(a) SCH Selection. After the selection of a CH, this CH selects its representative in the cluster that we called the SCH. The selection is based on the eligibility E_c of each CM. A CM with the highest eligibility will be declared in the next CHHello as a SCH.

In our approach, the secondary cluster head is always functioning even if the CH is still alive in the cluster. The differences between the CH and SCH are that the SCH has no packet overhead; it does not send any packet except Hello message that is sent by any ordinary CM. In addition, the CH decides the SCH selection and demobilization by periodically checking each CM eligibility in the CMT table before sending the CHHello message. The role of SCH is very important in stabilizing the cluster when the CH is likely to lose the link with its CMs because of CHHello packet frequent loss, or when the CH and a member becomes temporary inconsistent due to a sudden speed change.

Once a CM receives a CHHello from its CH, it checks the SID field. If the SID matches its own VID, it becomes the SCH and has the responsibility of relaying any received packet from its neighbors to the CH. This role will last until this SCH receives another CHHello from its CH with different SID.

(b) All-Member-Interests-Based Merging. Cluster merging is to combine two clusters to form a larger one. Cluster merging threatens the stability for the sake of decreasing the number of clusters. When a merging decision is taken by the CH depending on its interest, there is no guarantee that the CMs are welcomed in the new merged cluster, leading to unnecessary reclustering. However, merging has another ulterior advantage which is reducing the number of alienated clusters. An alienated cluster is a cluster that has a lonesome vehicle acting as a CH. By increasing the number of alienated CHs, clustering becomes senseless.

In order to benefit from merging without suffering from instability, cluster merging in our AMIM scheme is based on all CMs interests instead of CH's as in the existing works discussed in Section 2. When two consistent CHs are moving within each other's range, the two CHs begin the cluster merging procedure. Merging conditions that must be checked by each CH are: First, all members of the current CH are consistent with the target CH. Second, the total number of members after merging is less than the maximum limit N_{max} . Third, the current CH is less eligible than the target CH. Whenever these conditions are met for a CH, it resigns from the CH role and sends JoinReq to be a CM in the new cluster. These conditions are more likely to be satisfied in case of alienated clusters rather than of larger clusters.

Algorithm 3 shows that when CMs receive a Hello message from their head n_i instead of a CHHello, they implicitly understand that n_i has merged with n_j that has the ID presented in CID_i field. Then, the CM sends JoinReq and becomes a CM without turning to the UV state. Although n_j will check the possibility of adding these new CMs, their request to join the new cluster cannot be rejected. This is because all joining conditions have been verified in advance by n_i before it decided to merge.

(c) Cluster Replacement. Normal vehicular traffic behavior causes CMs to move away of their CH. This unavoidable relative movement inside the cluster leads to a gradual decrease in link quality and expected remaining link lifetime. The member will end up alone outside the cluster, moving to the UV state and starting the reclustering process. To reduce unnecessary transition into the UV state, a CM has the right to change its cluster under certain conditions.

ALGORITHM 4: Cluster replacement.

Algorithm 4 illustrates that when the LET_{ci} between a CM and its CH n_i reaches a lower threshold LET_{Lth} , the CM begins looking for another CH n_j in its CHT table. The LET difference between n_i and n_j , if any, must exceeds the LET difference threshold ΔLET_{th} .

Thus, our method does not allow a CM to optionally leave its cluster unless there is no doubt that it will lose contact with the CH after a very short time, and the new CH will allow this member to stay longer by providing higher LET. The CM then sends a JoinReq message to the n_j to join its cluster. According to n_i , when it receives a Hello message from one of its members with different CID, it knows that this CM has changed the cluster and deletes it from its CMT.

4.2.5. Leaving a Cluster. From CM perspective, the CM checks the consistency and updates its CHT on receiving a CHHello from a CH. If the corresponding CH becomes inconsistent with a CM or the cluster member missed a CHHello from the CH, this indicates losing the connection with the CH. Thanks to the presence of the SCH, the CM does not have to leave the cluster unless the SCH is out of rage or inconsistent as well. In addition, the CM will not forsake its cluster immediately, but after a specific timeout T_{leav} to overcome the effect of control packet loss on clustering stability. This is what we call "Wait before Secession (WbS)" scheme. After losing both CH and SCH by a T_{leav} , the CM will move to the UV state and check its CHT to look for any alternative CH. In case of empty CHT, this vehicle will become a CH.

From CH perspective, the CH believes that a CM leaves the cluster if the CH stops hearing the CM heartbeat which is the periodic Hello message. Another reason is that the CM becomes inconsistent with the CH. In these situations, the CH erases it from its CMT, also after a timeout T_{out} , since this may be because of a transit change in vehicles mobility or due to a packet loss.

From SCH perspective, the SCH announce itself as a CH if it missed the CHHello message for a time equals to T_{S2P} unless another CH is available in its CHT.

5. Performance Evaluation

In this section, the performance of DHC is evaluated extensively. In our evaluation, we consider various simulation



FIGURE 6: Katy freeway from Google Map to NETEDIT.

environments and scenarios. In addition, different performance metrics are used to compare the proposed algorithm with another highly cited Threshold Based (TB) clustering algorithm approach [23]. Both algorithms are implemented on NS3 [39]. The evaluation details are described below.

5.1. Simulation Environments and Scenarios. In order to make an extensive performance evaluation, we consider two simulation environments. The first environment is a synthetic straight 5 km length highway. This highway has two moving directions with 6 lanes per direction; each lane is divided into a large number of edges. In addition, we selected a real highway from the Open Street Map (OSM) [40] which took into consideration road directions, number of lanes, and maximum allowed speed of the real street. Figure 6 shows the selected highway which is about 5 km of Katy Freeway in the Southern United States with three to five lanes per direction.

Vehicles traffic is generated using Simulation of Urban Mobility (SUMO) [41]. SUMO graphical network editor, NETEDIT, was used to design the synthetic highway and edit the realistic one. Then, different types of vehicle are injected to get different scenarios. Vehicles' speed during simulation is computed in SUMO depending on a modified Kruass

Vehicles Types	Acceleration $[m/s^2]$	Deceleration $[m/s^2]$	Speed	l Factor	Speed Deviation	
Type 1	2.9	7.5		1.3		
Type 2	1.12	3.27	1	1.11	0.13	
Type 3	1.85	0.9	1	.13	0.11	
		TABLE 4: Simulation scenari	os.			
Scenario		Vahiela Tymas				
	Туре	Lane Max. (Speed)	No. Lanes	Length	venicie Types	
S1	Synthetic	13.8-30 m/s	6 lanes	5 km	Type 1	
S2	Synthetic	30.5 m/s	6 lanes	5 km	Type 2	
S3	Realistic (Katy Freeway)	Realistic (29.06 m/s)	3-5 lanes	≈ 5 km	Type 3	

TABLE 3: Vehicle mobility parameters.

Car-following model using the attributes listed in Table 3. In addition, SUMO LC2013 lane-changing model was used to determine how vehicles change the lane on our multilane road.

Every vehicle has its own trip from a source edge to a destination edge which are chosen uniformly at random using the randomTrips.py tool provided by SUMO. The vehicles are randomly generated on the road edges within the first second of simulation by choosing a very high issue rate. Furthermore, the vehicles must pass at least 50 edges, before reaching their destination and leaving the simulation, which means that they would never leave before the end of the simulation. If a vehicle reaches the end of the highway, it turns and enters the opposite direction or the road. Such simulation setup stabilizes the number of vehicles during simulation.

The two simulation environments and three vehicles types are used to generate the three testing scenarios illustrated in Table 4. S1 and S2 Scenarios use the synthetic highway while S3 is built on the realistic one. Each testing scenario aims to provide a different simulation test to observe the behavior of the proposed clustering algorithm under various conditions. The first scenario S1 is a very dynamic traffic scenario in which the vehicles change their speed too much during simulation. This is because the successive highway edges have different maximum speeds, ranging from 13.8-30 m/s, in addition to the high acceleration and deceleration of the vehicles, vehicles type 1 in Table 3. In contrast to S1, S2 is considered a relatively static, with high speed vehicles, scenario as the maximum speed for all edges was fixed to 30.5 m/s. Scenario S3 is a fully realistic scenario in which we used real vehicle characteristics from [42] such as acceleration, deceleration and speed in addition to using the real highway. Our proposed clustering algorithm is implemented on Network Simulator NS3.26 with the simulation parameters shown in Table 5. The simulation runs 200 s for each scenario and each simulation point is averaged over 10 runs with the confidence interval illustrated. The clustering procedure starts after 1s and stops with the end of the simulation.

Vehicular communication channel modeling is very challenging compared to other wireless channels [35]. This is due to the special characteristics of vehicular communications, such as vehicles mobility, dynamic environment

Parameter	Value
NS3 version	3.26
Simulation time	200 s
SCHR	300 m
Frequency	5.9 GHz
Channel bandwidth	10 MHz
MAC protocol	IEEE 802.11p
T _{Hello}	0.1 s
ΔS_{th}	6 m/s
LET _{Lth}	3 s
T_E	$1.1 \ T_{Hello}$
ΔLET_{th}	20 s
T _{timeout}	3 T _{Hello}
T _{leav}	2 T _{Hello}
T _{out}	$1.1 \ T_{Hello}$
T_{s2p}	3 T _{Hello}
θ	$\pi/2$
μ_{LET}, μ_{SNR}	30, 20
$\alpha_{LET}, \alpha_{SNR}$	4, 2
$\overline{w_d, w_s, w_{N_c}, w_{LET}, w_{SNR}}$	1

TABLE 5: NS3 Simulation Parameters.

and antenna dimensions [43]. Furthermore, modeling the vehicular communication channel is highly affected by the environment (urban, suburban, highway, rural, ..., etc.) [44, 45]. Since VANET clustering is sensitive to signal propagation characteristics, we test each simulation scenario under the following three channel models:

- (i) Tow-Ray Ground Model: this model is widely used in the literature to evaluate VANET protocols. It is considered as a good path-loss model for rural environments characterized by the presence of a few scatterers [46].
- (ii) Tow-Ray Ground plus Nakagami-m fading: we use this model to study the behavior of the clustering process in the presence of small-scale fading. Nakagamim fading was proposed by [47, 48] to model the



FIGURE 7: CH lifetime comparison between DHC and TB. (a) Scenario S1. (b) Scenario S2. (c) Scenario S3.

multipath fading in rural, urban and highway environments.

(iii) Log-distance path-loss and log-normal shadowing: the impact of large-scale fading (shadowing) is studied using this model. The used model parameters are: a path-loss exponent equals to 1.61, a reference distance of 1 meter and the shadowing log-normal variable has zero mean and 3.4 dB standard deviation, as recommended by [49], to suit urban environments where there are a lot of static and dynamic obstacles that absorb, reflect, scatter, and, thus, attenuate the propagated signal.

5.1.1. Evaluation Metrics. Various evaluation metrics are used to evaluate the performance of the proposed algorithm. Some of the applied metrics are commonly used in the literature, while others are proposed, such as CH Alienation and Vehicle Alienation Time, to get a strict assessment. We classify the evaluation metrics into two categories: stability metrics and efficiency metrics.

Stability Metrics. They are used to estimate the ability of an algorithm to form and maintain stable clusters. The most common stability metrics are as follows:

(i) CH lifetime (CHL) is the average time that a vehicle spends in the CH state before changing its status.

(ii) CM lifetime (CML) is the average time that a vehicle spends in the CM state before changing to another state.

(iii) Number of changed states (NCS) is the average number of states that a vehicle changes during simulation time. Stable clustering requires this metric to be as small as possible.

Efficiency Metrics. Clustering algorithm stability is not an end in itself, but means to achieve better performance when it is used for a specific purpose such as routing, medium access control or security, etc. The metrics that indicate the algorithm efficiency are as follows:

(i) *Control Packet Overhead (CPO)*. It is the average number of control packets that a vehicle sends during simulation time excluding Hello messages.

(ii) *CH Alienation (CHA)*. It is the average percentage of time that the CH remains lonely with zero CM during its lifetime. A 100% of a CH Alienation means that no clustering took place and each cluster has only one participant which is the CH.

(iii) *Vehicle Alienation Time (VAT)*. It is the average percentage of time that the Vehicle remains in UV state during simulation time.

(iv) *Cluster Formation Rate (CFR)*. It is the average number of formed clusters per second by all vehicles in the system.

5.1.2. Comparison with TB. The performance of the proposed DHC algorithm is compared extensively with the performance of the TB algorithm. In order to achieve a fair comparison between the two algorithms, the common parameters were set to the same values. For example, T_E and T_{Hello} were set to 1.1 T_{Hello} and 0.1s, respectively. The comparison covers all above-mentioned evaluation metrics, simulation scenarios, and different vehicles densities.

(*a*) *Stability Metrics Comparison*. The average CH lifetime is compared for DHC and TB algorithms in different densities, simulation scenarios, and channel models as illustrated in Figure 7.

For both algorithms, the lifetime of a CH starts from the moment at which the vehicle announces itself as a CH to the time it merges with another cluster or to the end of the simulation, whichever comes first. As a result, the factors that affect the CH lifetime are the CH selection and merging schemes. In Figure 7, we observe that increasing vehicles density leads to smaller CH duration. The reason is that, the higher density means larger number of clusters and thus increases the probability of clusters merging that puts an end to a CH life. As long as AMIM merging scheme sets tough conditions to allow merging, DHC shows better CH lifetime in most scenarios.

Due to the radio signal attenuation under log-distance path-loss and log-normal shadowing, the coverage of the CH



FIGURE 8: CM lifetime comparison between DHC with one and two CHs and TB. (a) Scenario S1 with Two-Ray Ground. (b) Scenario S2 with Nakagami-m fading. (c) Scenario S3 with Nakagami-m fading.

becomes limited and irregular, causing smaller radius and noncircular clusters that contain smaller number of members (or even only the CH) to be formed. The condition of all members consistency, that is included in AMIM scheme, becomes less important when most of the clusters has no members which causes the CH lifetime for both TB and DHC to be close under log-normal shadowing. This seems clearer in Figure 7(a) for scenario S1 in which the vehicles change their speed frequently that makes coming across a consistent CH in the range a difficult matter. This interprets the longer CH lifetime for both algorithms in Figure 7(a) in presence of shadowing and dynamic mobility.

Under two-ray ground path-loss, DHC average CHL outperforms that in TB particularly when the vehicle density is high. With high number of vehicles, the cluster becomes more dense and harder to merge in DHC compared to TB. By adding Nakagami-m fading, TB suffers from successive leaving and remerging whenever a CHHello packet is lost. In TB, once a CM stops receiving the periodic CHHello from its CH, it announces itself as a CH in case there is no other CHs in its CHT. After receiving the CHHello again, this newly formed CH remerges and records a very short life. Since packet loss is common to occur in the presence of fading, TB experiences a very poor stability in presence of Nakagami-m fading. On the other hand, introducing the SCH and WbS approaches makes DHC more resilient in presence of packet loss as we can observe from Figure 7.

The second stability indicator is the Cluster Member Lifetime. To show the virtue of adding the second head on the stability of DHC, we compare the CML of TB and DHC with one and two CHs in Figure 8. Since the instability occurs in dynamic mobility and in presence of packet loss, we show CML values in Figure 8(a) for scenario S1 with Two-Ray Ground channel model, Figure 8(b) for scenario S2 with Nakagami fading, and Figure 8(c) for scenario S3 with Nakagami fading, too. The superiority of DHC with two heads is achieved in all scenarios and different channel models. This happens because of the SCH existence in each cluster to be connected whenever the link with the CH is lost. Adding a second cluster head adds more resilience to our algorithm and it becomes more effective in increasing the CM time in case of low packet delivery ratio and dynamic mobility. The DHC with one head is slightly better than TB in most cases due to the WbS feature. It is noticeable from Figure 8 that the CML is smaller in dynamic scenarios, Figure 8(a), than in static ones, Figures 8(b) and 8(c). The sudden vehicle speed changes lead to a consistency deprivation between the CH and some of its members. Then, many cluster members leave their clusters whenever the vehicles cross from street edge to another that has a different speed limit.

The last stability metric to be compared is the average number of state changes per vehicle that is shown in Figure 9. Figure 9 shows that DHC with two heads has smaller number of state transitions in all cases. Even in absence of the SCH, DHC still has a smaller number of changed states. This is because of AMIM scheme role to reduce state transitions. DHC reduction of state changes is up to 94% and 86% compared to TB and DHC with one head, respectively, under Nakagami-m fading channel with high vehicles density. As each reclustering event causes at least two state transitions in TB, it is worth mentioning that Figures 8(b)-8(c) and 9(b)-9(c) show that DHC is more resilient in presence of Nakagami-m fading. The Nakagami-m fading model maps to a very lossy environment characterized by a high loss rate in the upper layers.

As a summary for stability metrics comparison between DHC and TB, DHC is more stable under all mobility scenarios and channel models especially in the presence of fading. With low number of vehicles and simple channel model, TB gives a little bit better higher CH lifetime, whereas DHC has the best CH lifetime in all other cases. For CML and



FIGURE 9: Number of changed states comparison between DHC with one and two CHs and TB. (a) Scenario S1 with Two-Ray Ground. (b) Scenario S2 with Nakagami-m fading. (c) Scenario S3 with Nakagami-m fading.



FIGURE 10: Control packet overhead comparison between DHC and TB. (a) Scenario S1. (b) Scenario S2. (c) Scenario S3.

NCS stability metrics, DHC proved its superiority in all tested scenarios, channel models and vehicles density.

(*b*) *Efficiency Metrics Comparison.* Since a high packet overhead is considered one of the serious problems that faces clustering in VANET, CPO should not be ignored when any two clustering approaches are compared. CPO of DHC and TB algorithms are compared in Figure 10.

Clustering packet overhead is mainly composed of Hello packets overhead, cluster formation overhead, and cluster maintenance overhead. Both TB and DHC have the same Hello packet overhead; therefore, we do not consider this overhead in the CPO calculations.

In the cluster formation step, TB needs an additional message to initiate the cluster formation that is not required by DHC, as the formation in DHC starts automatically after the TE time. Then, the cluster formation process in TB needs one extra packet for each formed cluster.

Cluster maintenance overhead is particularly caused by the periodic Hello messages that include Hello and CHHello packets (not considered also in CPO). SCH selection in DHC does not need any special packets to be sent. Cluster merging and cluster replacement need only a JoinRequest to be unicasted to the related CH. Merging JoinRequest message is needed by both DHC and TB. Cluster Replacement scheme is not included in TB. However after the CM, that is threatened with expulsion, loses the connection with its CH it has to send the same JoinRequest to another CH, or announce itself as a CH by sending another message. As a result, both TB and DHC have the same maintenance overhead theoretically. However, cluster merging is more frequent to happen with TB than with DCH. Moreover, frequent reclustering problem that TB suffers from causes the number



FIGURE 11: Cluster formation rate comparison between DHC and TB. (a) Scenario S1. (b) Scenario S2. (c) Scenario S3.



FIGURE 12: Vehicle Alienation Time comparison between DHC and TB. (a) Scenario S1. (b) Scenario S2. (c) Scenario S3.

of clusters to dramatically increase, principally in case of packet loss or dynamic environment.

As reclustering and cluster merging are less to happen under the log-distance channel model in a dynamic scenario, CPO is in low levels for both TB and DHC in Figure 10(a) that represents scenario S1. In less dynamic scenarios, Figures 10(b) and 10(c), the average control packets transmitted by each vehicle are almost independent of the vehicle density when the DHC is used. With TB, similar behavior is shown unless the fading is included, then, the CPO increases considerably under all mobility scenarios.

To compare how much the reclustering occurs in both TB and DHC, the average Cluster Formation Rate was measured and depicted in Figure 11. This rate is higher for TB in all cases in particular when the effect of fading or shadowing was taken into consideration, as expected.

The existence of a cluster with only the CH, or a vehicle that does not belong to any cluster, contradicts the concept of clustering and increases the number of clusters in the network. In this paper, we propose paying attention to Vehicle Alienation Time and CH Alienation before validating any clustering algorithm.

Figure 12 shows the comparison between VAT time when each of the compared algorithms is used. As we mentioned in Section 4.2.4, there are some procedures in DHC that were suggested to reduce VAT such as the cluster replacement scheme to avoid turning to UV state. Another way to reduce VAT was to set harsh conditions to diminish merging unless all CMs are guaranteed to join the merged cluster instead of becoming unclustered. For these reasons, DHC exhibits smaller alienation time in all scenarios as can be noticed from Figure 12. On the other hand, in case of recurrent clustering, that is much more in TB, the vehicle transits to UV state causing VAT time to increase. As an example, the vehicle spends up to 1.25% of the simulation time unclustered when TB algorithm is used under Nakagami-m fading, Figure 12(c),



FIGURE 13: CH Alienation comparison between DHC and TB. (a) Scenario S1. (b) Scenario S2. (c) Scenario S3.

while this time is reduced to 0.1% with DHC clustering algorithm.

Figure 13 shows the CHA for the compared algorithms. First observation of this figure tells that the more vehicles in the simulation, the less CHA values due to cluster size growing. In addition, the CHA value is considerably more under log-normal channels than other models for all scenarios presented in Figure 13. This is interpreted by the smaller cluster size as mentioned previously. In Figure 13(c), this alienation reaches 96% in TB and 92% in DHC which indicates that clustering almost lost its meaning with low number of vehicles in shadowed environments.

In general, our proposed DHC algorithm provides an evident efficiency compared to the TB algorithm under all tested cases, especially when channel fading was taken into consideration as shown by the above simulation results.

6. Conclusion and Future Research

In this paper, we have presented a mobility-based clustering approach that creates a minimum number of stable clusters in both urban and highway scenarios. Different vehicles mobility and link quality parameters, such as the relative position, speed, direction, popularity, SNR, and link expiration time, are used for cluster head selection. The cluster maintenance stage of the proposed approach introduced new features, including cluster replacement and all-memberinterests-based merging that contributed to the increased stability of the clusters. The proposed DHC scheme outperforms existing approaches in terms of cluster stability and efficiency, under different vehicles density, channel models, and traffic scenarios, especially in dynamic mobility environments. The improvement in the cluster stability due to adding the second CH has been investigated, by comparing the algorithm stability metrics with and without adding the second cluster head. It is also observed from the results that the fading significantly influences clustering algorithm performance, unless the packet loss was considered in the

algorithm design as in DHC. In addition, our results showed that validating any clustering algorithm without taking into consideration different conditions and scenarios results in misleading estimation of the relative performance between tested algorithms.

Our future research targets an intersection turning prediction scheme and algorithm parameters adaptation will be added to enable using DHC in complicated urban and highway scenarios. Furthermore, machine learning techniques should be used to optimize the different algorithm parameters in real-time, and thus optimize the performance of DHC in various scenarios.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- S. I. Boucetta, C. Z. Johanyák, and L. K. Pokorádi, "Survey on software defined vanets," *Gradus*, vol. 4, no. 1, pp. 272–283, 2017.
- [2] A. Rasheed, S. Gillani, S. Ajmal, and A. Qayyum, "Vehicular ad hoc network (vanet): A survey, challenges, and applications," in *Vehicular Ad-Hoc Networks for Smart Cities*, A. Laouiti, A. Qayyum, and M. N. Mohamad Saad, Eds., pp. 39–51, Springer, Singapore, 2017.
- [3] S. Vodopivec, J. Bešter, and A. Kos, "A multihoming clustering algorithm for vehicular ad hoc networks," *International Journal* of Distributed Sensor Networks, vol. 10, no. 3, Article ID 107085, 2014.
- [4] L. Liu, C. Chen, T. Qiu, M. Zhang, S. Li, and B. Zhou, "A data dissemination scheme based on clustering and probabilistic

broadcasting in VANETs," Vehicular Communications, vol. 13, pp. 78–88, 2018.

- [5] A. A. Khan, M. Abolhasan, and W. Ni, "An evolutionary game theoretic approach for stable and optimized clustering in vanets," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4501–4513, 2018.
- [6] P. Basu, N. Khan, and T. D. C. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proceedings of the* 21st International Conference on Distributed Computing Systems Workshops, ICDCSW 2001, pp. 413–418, USA, April 2001.
- [7] B. Subba, S. Biswas, and S. Karmakar, "A game theory based multi layered intrusion detection framework for wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 25, no. 4, pp. 399–421, 2018.
- [8] C. BrijilalRuban and B. Paramasivan, "Cluster-based secure communication and certificate revocation scheme for VANET," *The Computer Journal*, vol. 62, no. 2, pp. 263–275, 2019.
- [9] R. R. Sahoo, R. Panda, D. K. Behera, and M. K. Naskar, "A trust based clustering with ant colony routing in VANET," in *Proceedings of the Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*, pp. 1–8, IEEE, Coimbatore, India, 2012.
- [10] A. Sahoo, S. K. Swain, B. K. Pattanayak, and M. N. Mohanty, "An optimized cluster based routing technique in vanet for next generation network," in *Information Systems Design and Intelligent Applications*, S. C. Satapathy, J. K. Mandal, S. K. Udgata, and V. Bhateja, Eds., pp. 667–675, Springer, New Delhi, India, 2016.
- [11] V. Sethi and N. Chand, "A destination based routing protocol for context based clusters in VANET," *Communications and Network*, vol. 09, no. 03, pp. 179–191, 2017.
- [12] A. Shahen Shah, H. Ilhan, and U. Tureli, "CB-MAC: a novel cluster-based MAC protocol for VANETs," *IET Intelligent Transport Systems*, 2018.
- [13] A. F. Shah, H. Ilhan, and U. Tureli, "qCB-MAC: QoS aware cluster-based MAC protocol for VANETs," in *Intelligent Computing*, vol. 857 of *Advances in Intelligent Systems and Computing*, pp. 685–695, Springer, Cham, Switzerland, 2018.
- [14] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-clusterbased IEEE 802.11p and LTE hybrid architecture for VANET safety message dissemination," *IEEE Transactions on Vehicular Technology*, pp. 2621–2636, 2016.
- [15] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A comparative survey of VANET clustering techniques," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 657–681, 2017.
- [16] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, "A unified framework of clustering approach in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1401–1414, 2018.
- [17] L. Zhao, Y. Li, C. Meng, C. Gong, and X. Tang, "A SVM based routing scheme in VANETs," in *Proceedings of the 16th International Symposium on Communications and Information Technologies, ISCIT 2016*, pp. 380–383, China, September 2016.
- [18] M. M. Caballeros Morales, C. S. Hong, and Y.-C. Bang, "An Adaptable mobility-aware clustering algorithm in vehicular networks," in *Proceedings of the 13th Asia-Pacific Network Operations and Management Symposium: Managing Clouds, Smart Networks and Services, APNOMS 2011*, Taiwan, September 2011.
- [19] A. Ahizoune and A. Hafid, "A new stability based clustering algorithm (SBCA) for VANETs," in *Proceedings of the 2012 IEEE*

37th Conference on Local Computer Networks Workshops, LCN Workshops 2012, pp. 843–847, USA, October 2012.

- [20] K. A. Hafeez, L. Zhao, Z. Liao, and B. N.-W. Ma, "A fuzzylogic-based cluster head selection algorithm in VANETs," in *Proceedings of the International Conference on Communications* (ICC), pp. 203–207, IEEE, Canada, 2012.
- [21] G. V. Rossi, Z. Fan, W. H. Chin, and K. K. Leung, "Stable clustering for Ad-Hoc vehicle networking," in *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2017.
- [22] S.-C. Lo, Y.-J. Lin, and J.-S. Gao, "A multi-head clustering algorithm in vehicular ad hoc networks," *International Journal* of Computer Theory and Engineering, vol. 5, no. 2, pp. 242–247, 2013.
- [23] Z. Y. Rawashdeh and S. M. Mahmud, "A novel algorithm to form stable clusters in vehicular ad hoc networks on highways," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 15, 2012.
- [24] C. Shea, B. Hassanabadi, and S. Valaee, "Mobility-based clustering in VANETs using affinity propagation," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, November 2009.
- [25] R. T. Goonewardene, F. H. Ali, and E. Stipidis, "Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks," *IET Intelligent Transport Systems*, vol. 3, no. 2, pp. 148–158, 2009.
- [26] R. Sugumar, A. Rengarajan, and C. Jayakumar, "Trust based authentication technique for cluster based vehicular ad hoc networks (VANET)," *Wireless Networks*, vol. 24, no. 2, pp. 373– 382, 2018.
- [27] X. Cheng and B. Huang, "A center-based secure and stable clustering algorithm for VANETs on highways," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 8415234, 10 pages, 2019.
- [28] I. Ahmad, R. M. Noor, I. Ahmedy et al., "VANET-LTE based heterogeneous vehicular clustering for driving assistance and route planning applications," *Computer Networks*, vol. 145, pp. 128–140, 2018.
- [29] X. Ge, Q. Gao, and X. Quan, "A novel clustering algorithm based on mobility for VANET," in *Proceedings of the 2018 IEEE* 18th International Conference on Communication Technology (ICCT), pp. 473–477, Chongqing, China, 2018.
- [30] E. Souza, I. Nikolaidis, and P. Gburzynski, "A new Aggregate Local Mobility (ALM) clustering algorithm for VANETs," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2010.
- [31] R. Ghosh and S. Basagni, "Mitigating the impact of node mobility on ad hoc clustering," Wireless Communications and Mobile Computing, vol. 8, no. 3, pp. 295–308, 2008.
- [32] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau, and D. Gruyer, "Chain-Branch-Leaf: A clustering scheme for vehicular networks using only V2V communications," *Ad Hoc Networks*, vol. 68, pp. 70–84, 2018.
- [33] A. Mehmood, A. Khanan, A. H. H. M. Mohamed, S. Mahfooz, H. Song, and S. Abdullah, "ANTSC: An intelligent naïve bayesian probabilistic estimation practice for traffic flow to form stable clustering in VANET," *IEEE Access*, vol. 6, pp. 4452–4461, 2018.
- [34] J. Garbiso, A. DIaconescu, M. Coupechoux, and B. Leroy, "Autoadaptive multi-hop clustering for hybrid cellular-vehicular networks," in *Proceedings of the 20th IEEE International Conference*

on Intelligent Transportation Systems (ITSC), pp. 1-6, Japan, October 2017.

- [35] L. Liang, H. Peng, G. Y. Li, and X. Shen, "Vehicular communications: A physical layer perspective," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10647–10659, 2017.
- [36] Y. Huo, Y. Liu, X. Xing, X. Cheng, L. Ma, and T. Jing, "A low overhead and stable clustering scheme for crossroads in VANETs," in *Wireless Algorithms, Systems, and Applications,* Lecture Notes in Computer Science, pp. 232–242, Springer, Cham, Switzerland, 2015.
- [37] Y. Chen, M. Fang, S. Shi, W. Guo, and X. Zheng, "Distributed multi-hop clustering algorithm for VANETs based on neighborhood follow," *EURASIP Journal on Wireless Communications* and Networking, vol. 2015, no. 1, p. 98, 2015.
- [38] R. S. Bali, N. Kumar, and J. J. P. C. Rodrigues, "An intelligent clustering algorithm for VANETs," in *Proceedings of the 3rd International Conference on Connected Vehicles and Expo* (ICCVE), pp. 974–979, Austria, 2014.
- [39] "Network simulator, ns-3," http://www.nsnam.org/, Accessed August, 2018.
- [40] "Open Street Map, OSM," https://www.openstreetmap.org, Accessed August, 2018.
- [41] "Simulation of Urban MObility, SUMO," https://sourceforge .net/projects/sumo/, Accessed August, 2018.
- [42] A. Marella, A. Bonfanti, G. Bortolasor, and D. Herman, "Implementing innovative traffic simulation models with aerial traffic survey," in *Transport Infrastructure and Systems*, G. DellAcqua and F. Wegman, Eds., pp. 571–577, CRC Press, Italy, 2017.
- [43] W. Viriyasitavat, M. Boban, H.-M. Tsai, and A. Vasilakos, "Vehicular communications: survey and challenges of channel and propagation models," *IEEE Vehicular Technology Magazine*, vol. 10, no. 2, pp. 55–66, 2015.
- [44] H. Nourkhiz Mahjoub, A. Tahmasbi-Sarvestani, S. M. O. Gani, and Y. P. Fallah, "Composite α-μ based DSRC channel model using large data set of RSSI measurements," *IEEE Transactions* on *Intelligent Transportation Systems*, vol. 99, pp. 1–13, 2018.
- [45] M. Boban and W. Viriyasitavat, "Channel models for vehicular communications," in *Vehicular Ad Hoc Networks*, pp. 335–363, Springer, 2015.
- [46] J. Karedal, N. Czink, A. Paier, F. Tufvesson, and A. F. Molisch, "Path loss modeling for vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 323–328, 2011.
- [47] M. K. Samimi, T. S. Rappaport, and G. R. Maccartney, "Probabilistic omnidirectional path loss models for millimeter-wave outdoor communications," *IEEE Wireless Communications Letters*, vol. 4, no. 4, pp. 357–360, 2015.
- [48] G. P. Grau, D. Pusceddu, S. Rea, O. Brickley, M. Koubek, and D. Pesch, "Vehicle-2-vehicle communication channel evaluation using the CVIS platform," in *Proceedings of the 7th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP 2010)*, pp. 449–453, UK, July 2010.
- [49] J. Kunisch and J. Pamp, "Wideband car-to-car radio channel measurements and model at 5.9 GHz," in *Proceedings of the 2008 IEEE 68th Vehicular Technology Conference*, pp. 1–5, Canada, September 2008.

