EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

An enhanced low overhead and stable clustering scheme for crossroads in VANETs

Yan Huo[1,2*†], Yuejia Liu[1†], Liran Ma[3], Xiuzhen Cheng[2] and Tao Jing[1]

Abstract

In this paper, we study the clustering problem for crossroads in Vehicular Ad hoc Networks (VANETs). Considering the load balancing of both the whole network and each cluster based on the multiple metrics, an Enhanced Low Overhead and Stable Clustering (EnLOSC) scheme is presented to ensure the stability and security of clusters and to reduce the communication overhead in this case. The proposed capability metric, designed to find the vehicles with similar direction and better channel quality, is exploited in the processes of formation and maintenance to determine which node is suitable for a cluster head. Based on this, a Cluster Head Electing in Advance Mechanism (CHEAM) is developed in order to fairly select a new head for "isolated" vehicles that may not belong to a cluster. Meanwhile, other metrics are related to the node density and cluster size, which are exploited in the Cluster Merging and Splitting Mechanisms to keep the system load balancing and to improve the communication quality. Furthermore, the proposed Discovery and Elimination Scheme (DES) is designed to tackle the malicious nodes that may hurt the cluster communication. Accordingly, an enhanced cluster maintenance strategy with multi-metrics and a secure scheme is proposed so as to reduce the number of isolated vehicles, keep appropriate loading for each cluster head, and protect the whole link over cluster communication. Numerical results and discussion indicate that the cluster stability, communication overhead, load balance, and security can be significantly enhanced by our proposed scheme.

Keywords: VANETs, Cluster formation and maintenance, Load balancing, Cluster stability, Network overhead

1 Introduction

Communication in Vehicular Ad hoc Networks (VANETs), which are considered to be a special class of Mobile Ad hoc Networks (MANETs), becomes an important research topic with the spectrum allocation for Intelligent Transportation System (ITS) and the development of Dedicated Short Range Communication (DSRC) standards [1, 2]. In particular, the DSRC is an important technology designed for ITS, which requires a short-range, wireless link to transmit signals only for Vehicle-to-Vehicle (V2V) communication and Vehicle-to-Infrastructure (V2I) communication. Naturally, the multi-hop and relay technology typically exploited in MANETs is introduced in this network. However, the existing methods that enable communication in MANETs cannot be directly applied in VANETs due to the following characteristics [3, 4]. First, the fast movement of vehicles can lead to a highly dynamic and frequently disconnected network topology. Second, the trajectories of the vehicles in VANETs are strictly restricted by the layout of roads. Clustering-based methods that divide vehicles into clusters by taking advantage of the layout-determined trajectories are considered as effective ways to facilitate communication in VANETs. Stable communication can be achieved in highly dynamic VANETs through cluster-based communication where a leader is selected within each cluster to handle intra-cluster and inter-cluster traffic.

As we know, clustering, which has been already extensively researched in the past [5, 6], is the task of grouping a set of nodes (mobile devices, vehicles, etc.) with some similar properties based on the predefined rules. However, there exist various difficult challenges to design a reliable communication in the VANET scenario, many of

*Correspondence: yhuo@bjtu.edu.cn
†Equal contributors
[1] School of Electronics and Information Engineering, Beijing Jiaotong University, 100044 Beijing, China
[2] Department of Computer Science, The George Washington University, 20052 Washington, DC, USA
Full list of author information is available at the end of the article

which can be addressed by a clustered network [7]. The reason of the hard design is the rapidly changing network topology caused by the highly mobile environment, which may result in data congestion [8] and low Quality of Service (QoS) [9]. Additionally, inevitable situations, such as traffic jams and crossroads, also lead to contention and the hidden terminal problem, especially in a dense network.

Aimed at the clustering-based communication in VANETs, the goal of designed rules in clustering algorithms is to achieve stable, easy, quick, and efficient communication with necessary QoS requirements. Accordingly, many works have been done to develop effective clustering algorithms for VANETs with most of them focusing on the scenario of the highway or straight lanes [4, 10–19]. However, the performance of these schemes turns out to be unsatisfactory when it comes to a city scenario with crossroads. This is because a large number of vehicles can become isolated at crossroads. As a result, a considerable amount of communication overhead and congestion can result from the routing discovery processes for the isolated vehicles. To deal with these problems, we have proposed a novel clustering scheme for the crossroads in VANETs in [20], with the objective to stabilize the clusters, minimize the number of isolated nodes, and reduce the communication overhead. However, our former proposed scheme only focused on the efficiency of formation and maintenance methods using the vehicle mobility and transmission quality, regardless of the system load and network congestion. Furthermore, there may be some malicious vehicles that interfere or even destroy the cluster communication security. Thus, it is also important to design an effective mechanism in our scheme for the crossroads.

Accordingly, in order to enhance the security and system load balancing, we decide to present other novel metrics and introduce a malicious vehicle discovery and elimination scheme in the cluster strategy. The main contributions of this paper are threefold:

- We propose a novel clustering scheme named Enhanced Low Overhead and Stable Clustering (EnLOSC) for crossroads in VANETs, which includes a cluster formation algorithm and a cluster maintenance scheme.
- In order to implement the load balancing for both network and cluster during the maintenance phase, we propose the cluster size and node density metrics to analyze the network load for the purpose of ensuring the similar size and density of a cluster. In the meantime, we also introduce the capability metric to select and update a cluster head by considering both the mobility and the transmission power loss of the vehicle.
- A Cluster Head Electing in Advance Mechanism (CHEAM) is to help a cluster member select a new cluster by predicting its stay time, while the Cluster Merging and Splitting Mechanisms are proposed to keep the network load balancing by observing node density and cluster size. Meanwhile, a secure method, called a Discovery and Elimination Scheme (DES), is presented to find and remove malicious nodes in one cluster.

To the best of our knowledge, this is the first work to combine multi-metrics and security mechanism with the clustering algorithm for the crossroad situation in VANETs. The rest of the paper is organized as follows. Section 2 and Section 3 describe our problem formulation and the metrics for the cluster scheme, respectively. Accordingly, we show the details of cluster formation and maintenance algorithms for the purpose of achieving low overhead and secure and stable load balancing communication in Section 4 and Section 5. Sequentially, numerical analysis and discussion are presented in Section 6 to evaluate the performances of our scheme. In the end, the related works and conclusion are summarized in Section 7 and Section 8, respectively.

2 Problem formulation

Our system model is based on a bidirectional multi-lane city road scenario with a crossroad as illustrated in Fig. 1. We assume that all vehicles are equipped with GPS so that each vehicle is aware of its own location (represented by Cartesian coordinates), velocity, and direction (represented by direction vector) at any time. We further assume that the precise time is known and traceable to the Coordinated Universal Time (UTC). We also assume that all vehicles can send packets with a unified transmitting power P_t and decode received packets about the threshold P_r.

As shown in Fig. 1, there exist a number of clusters. Nodes that are in a dashed box belong to the same cluster. The *cluster heads* are red nodes, and the *cluster members* are black ones. In addition, the nodes in gray color are called *hopping cluster members* because they are about to leave the current cluster and hop to a new one. The *undecided nodes* in white color are the isolated nodes. In a cluster, there are one cluster head, several cluster members, and hopping cluster members. The cluster head is responsible for handling the intra-cluster communication and relays the inter-cluster communication among clusters. Note that we use node and vehicle interchangeably in the rest of this paper,

Similar to [15], only those nodes that are moving in the same direction can be clustered together. Once a node joins in a cluster and becomes a cluster member, a timer named TimerS that is related to its predicted stay time in

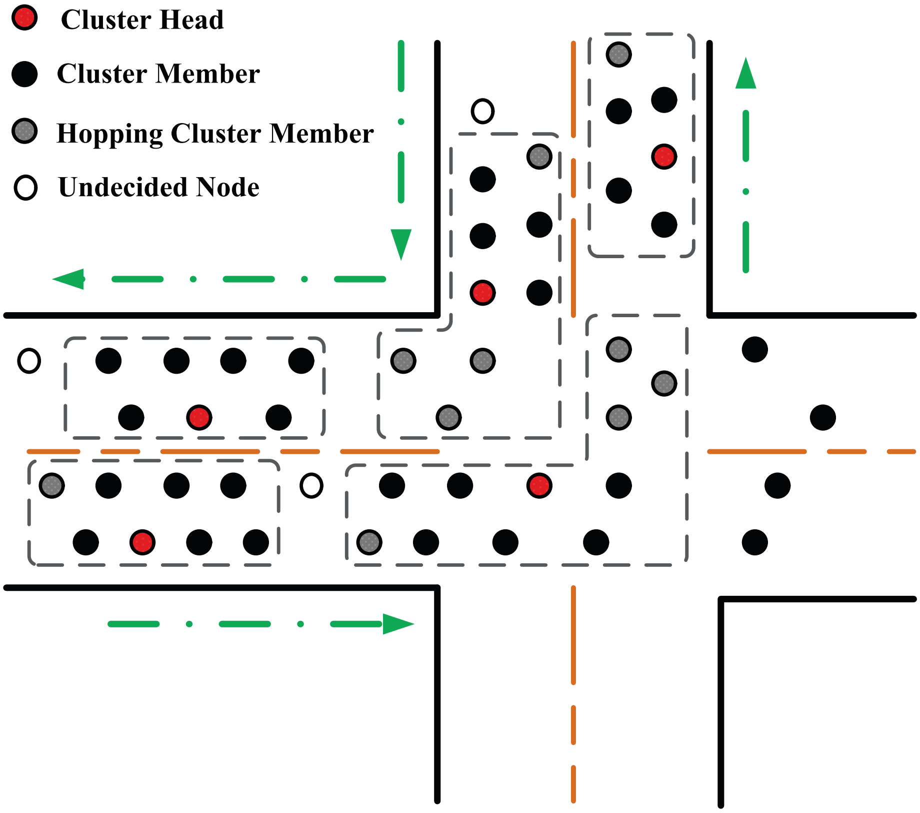

Fig. 1 System model

the cluster starts. The definition of TimerS for member j in cluster i is:

$$\mathrm{TimerS}(i,j) = T_{j,i}^{\mathrm{stay}} - T_f, \qquad (1)$$

where $T_{j,i}^{\mathrm{stay}}$ is the predicted stay time of member j in cluster i which is detailed in Section 5.1.1. T_f is the ideal time of a cluster formation procedure, which includes the packet transmission cost and capability metric comparison cost.

Because of the dynamic topology of VANETs, a cluster member may change to be a hopping cluster member. When the change happens, the hopping member starts searching for a new cluster head to join even though it may still belong to the current cluster. In addition, the isolated nodes continuously search for clusters to join. Note that if there are too many isolated nodes in a dense network, the total communication overhead increases significantly and can lead to poor network performance. Therefore, it is important to design a clustering algorithm that reduces the number of isolated nodes as much as possible.

3 Metrics for clustering strategy

This section describes three metrics, the capability, scale, and node density, which are exploited to design formation and maintenance algorithms by the node and network properties, respectively, for the crossroad scenario in VANETs.

3.1 The capability metric

Taking into account both the node's mobility and the transmission power loss, we firstly design a metric to measure a node's capability of acting as a cluster head.

Actually, nodes can obtain their position, velocity, and direction information based on the data derived from GPS. Let $\vec{D_i} = D_{ix}\vec{x} + D_{iy}\vec{y}$ be the direction vector of node i, where $\vec{x}$ and $\vec{y}$ are the unit vectors of the X and Y axes. The angle between the direction of two nodes i and j can be calculated as:

$$\theta_{i,j} = \arccos \frac{\vec{D_i} \cdot \vec{D_j}}{|\vec{D_i}|\,|\vec{D_j}|} = \arccos \frac{D_{ix}D_{jx} + D_{iy}D_{jy}}{\sqrt{D_{ix}^2 + D_{iy}^2}\sqrt{D_{jx}^2 + D_{jy}^2}}. \qquad (2)$$

We consider node i and node j are moving in the same direction when $\theta \leq \pi/4$ and consider they are moving in different directions when $\theta > \pi/4$. In this way, we could avoid mistakenly labeling the node as changing lanes when it is actually turning at the crossroads.

Similarly, let v_i and v_j denote the velocity of nodes i and j obtained from the GPS. The relative velocity of node i and node j can be calculated as follows:

$$v_{i,j}^{\mathrm{rel}} = \left| v_i - v_j \right| . \tag{3}$$

We use the Relative Velocity Metric (RVM) to indicate the relative mobility between two moving nodes:

$$\mathrm{RVM}(i,j) = \log \frac{v_{\max}}{v_{\max} - v_{i,j}^{\mathrm{rel}}}. \tag{4}$$

Here, $v_{\max}$ is the upper boundary of the velocity. When node i has n one-hop (direct) neighbors, the RVM value of node i can be calculated as:

$$\mathrm{RVM}(i) = \frac{1}{n} \sum_{j=1}^{n} \mathrm{RVM}(i,j) = \frac{1}{n} \sum_{j=1}^{n} \log \frac{v_{\max}}{v_{\max} - v_{i,j}^{\mathrm{rel}}}. \tag{5}$$

Clearly, $\mathrm{RVM}(i)$ is not smaller than 1. A smaller $\mathrm{RVM}(i)$ indicates that node i's velocity is more similar with that of its direct neighbors. That is, a node with smaller RVM is more likely to stay with its direct neighbors for a longer time due to their similar velocity. Therefore, a node with a lower RVM value is preferred to act as the cluster head to make the cluster more stable.

As described in Section 2, P_t is the unified transmission power of all nodes and $P_r(i,j)$ denotes the received power of node i from node j. We define the transmission Power Loss Metric (PLM) between node i and node j as:

$$\mathrm{PLM}(i,j) = \log \frac{P_t}{P_r(i,j)}. \tag{6}$$

When node i has n direct neighbors, the PLM of node i can be presented as:

$$\mathrm{PLM}(i) = \frac{1}{n} \sum_{j=1}^{n} \log \frac{P_t}{P_r(i,j)}. \tag{7}$$

A $\mathrm{PLM}(i)$ that is not smaller than 1 is related to the average channel quality and the sum of distance between a node and its direct neighbors. A node with a smaller PLM value is more likely to have a shorter communication distance and better channel quality with its direct neighbors.

Taking both $\mathrm{RVM}(i)$ and $\mathrm{PLM}(i)$ into consideration, we define a capability metric M to describe the capability of a node to be a cluster head:

$$M(i) = \mathrm{RVM}(i) + \mathrm{PLM}(i). \tag{8}$$

A node with a smaller M value implies that this node has more similar mobility with its direct neighbors and better channel quality. In other words, a more stable cluster can be formed by selecting a node with a smaller M value as the cluster head.

3.2 The cluster size metric

Generally, there exists a challenge brought by the clustering algorithm, which is related to the number of cluster members. As we know, the scale of a cluster will be too large to maintain and update when there are many nodes in this cluster, which may result in a heavy load on the cluster head and worse communication quality. To deal with it, we decide to take into account the communication load on each cluster head, which is represented by the number of members in this cluster, for the purpose of the load balancing and well-performed clustering scheme that is suitable for the high node density scenario.

Actually, each cluster has its own head and members after the initialization and formation stage. For clarity, we assume that N_{member} is the number of head's members, which can be recorded by the periodic broadcast packets, the Cluster Member Announcement (*CMA*) packets, in our scheme shown in Section 5, during the maintenance process. Accordingly, we can give the definition and the illustration of the cluster size metric as below.

$$k_s(i) = \frac{N_{\mathrm{member}}(i)}{N_{\max}(i)} \tag{9}$$

where the cluster size metric $k_s(i)$ is related to the number of i's members. Intuitively, if the number of members is greater than $N_{\max}(i)$, it means that this cluster needs to be reformed and updated in order to keep the communication load of the head i bearable.

Here, we assume that $N_{\max}$ is an ideal upper limit of the number of cluster head's members, which represents the maximum number of members a cluster head can manage and handle without disconnection and congestion. The value of $N_{\max}(i)$ for cluster head i is determined by two factors: the max bandwidth $B(i)$ and the bandwidth allocation method. For simplicity, we consider two kinds of bandwidth allocation. The first one, called the equal bandwidth allocation for each cluster member, is shown in Fig. 2, which is assumed that every cluster member occupies a unit bandwidth B_0. Therefore, the value of $N_{\max}$ can be described as

$$N_{\max} = \frac{B(i)}{B_0} \tag{10}$$

The other method, depicted as Fig. 3, is allocated hierarchically based on the demand and priority of members in a cluster. In this case, some special nodes that are in charge of forwarding massive or important messages require more bandwidth than others. Prioritizing the bandwidth of cluster members with high, medium, or low, we assume that there are only one node with high priority, two with medium priority, and the rest with low priority in one cluster, of which bandwidth are defined as B_{high}, B_{medium}, and B_{low}, respectively. Note that nodes

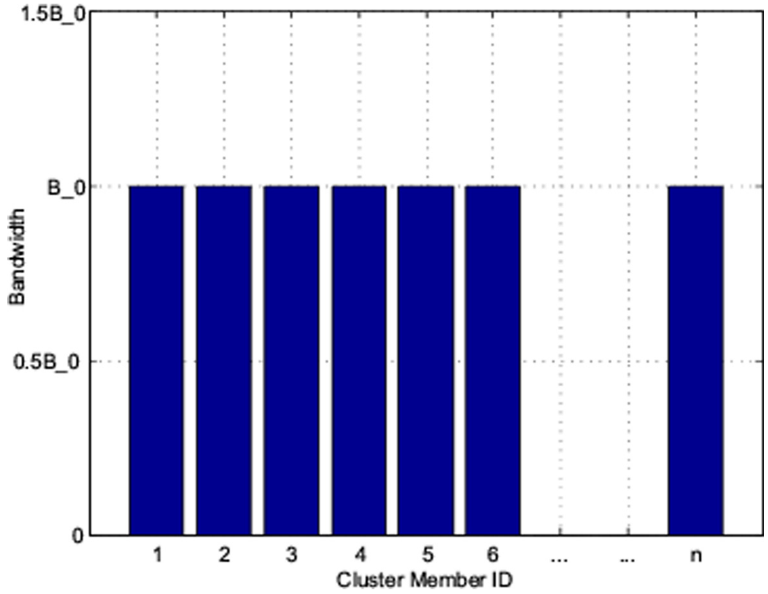

Fig. 2 Bandwidth allocated equally

with high or medium priority will occupy more bandwidth than ones with low priority. Without loss of generality, the relationship of bandwidth among high, medium, and low priority is depict as $B_{\text{high}} = 2B_{\text{low}}$ and $B_{\text{medium}} = 1.5B_{\text{low}}$. Thus, the maximum number of cluster head *i*'s members can be calculated as:

$$\begin{aligned} N_{\max} &= N_{\text{high}} + N_{\text{medium}} + N_{\text{low}} \\ &= 1 + 2 + \frac{B(i) - B_{\text{high}} - 2B_{\text{medium}}}{B_{\text{low}}} \\ &= \frac{B(i)}{B_{\text{low}}} - 2 \end{aligned} \tag{11}$$

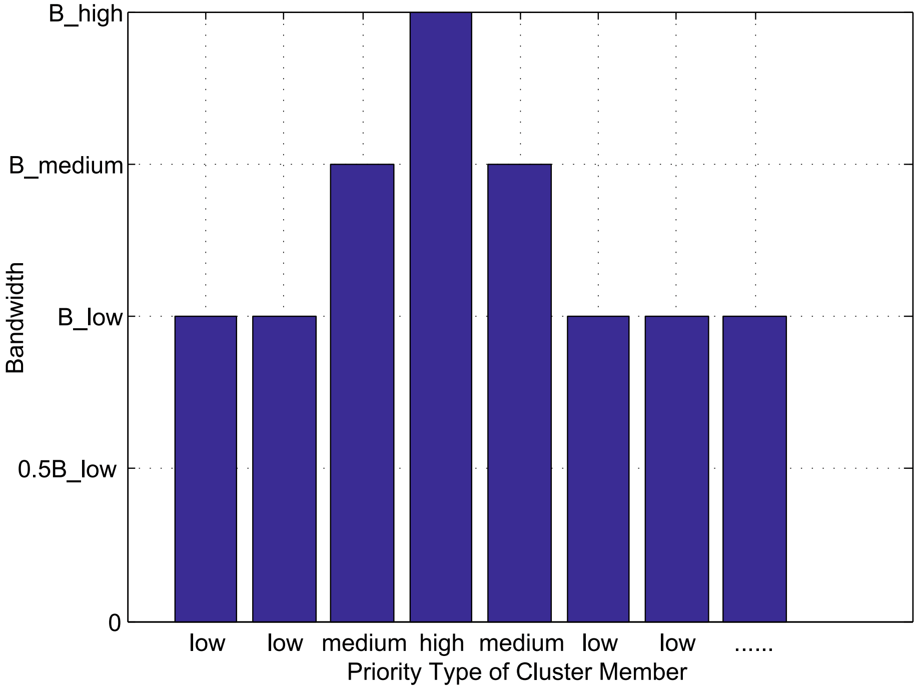

Fig. 3 Bandwidth allocated hierarchically

3.3 The node density metric

Similar to the cluster size, each node should be sensible of its direct neighbors to help to determine or select whether or which node around it is suitable to the head and what cluster size should be restricted. Assuming $N_{\text{neighbors}}(i)$ is the number of node *i*'s direct neighbors, we exploit the node density metric, $k_d(i)$, to describe the number ratio between *i*'s neighbors and $N_{\max}$.

$$k_d(i) = \frac{N_{\text{neighbors}}(i)}{N_{\max}(i)} \tag{12}$$

Obviously, when $0 < k_d(i) < 1$, the density is not so high that the node *i* is capable to relay and communicate within the cluster or among clusters. In one word, the larger $k_d(i)$ is, the denser the nodes are distributed in the radius range of the detected node *i*.

4 Cluster formation scheme

According to the various metrics defined in the previous subsection, we present the EnLOSC scheme that contains a cluster formation algorithm and a cluster maintenance scheme. For the convenience of description, we assume every node uses its own ID as an identification and the cluster ID is represented by the cluster head's ID.

Before forming or changing cluster-based network topology, there are some *undecided nodes* or *hopping cluster members* that want to join in a new cluster. For this purpose, these nodes broadcast *HELLO* packets, which contain the position, velocity, and direction information, to their direct neighbors. At the same time, when there are cluster heads in the network, the heads also broadcast their information, called Cluster Head Announcement (*CHA*) packets that contain the cluster head's ID, position, velocity, and direction.

Accordingly, when node *j* receives *HELLO* packets or *CHA* packets, it adds the senders' IDs into its Direct Neighbors List (DNL). Then, it uses (2) to calculate the angle θ between its direction and each direct neighbor's direction. If $\theta > \pi/4$, the corresponding neighbor is considered moving in a different direction and deleted from its DNL. After checking θ and updating DNL, node *j* computes the *M* value based on (8) and sends it to other nodes in its DNL.

If there is only one cluster head in node *j*'s DNL, it sends a *ClusterJoin* packet including its ID to the head and becomes a member of the cluster. If there are more than one cluster heads in node *j*'s DNL, it selects the head with the smallest *M* value and sends a *ClusterJoin* packet to the head. If there is no cluster head in node *j*'s DNL, it compares its *M* value with that of its direct neighbors. When node *j* finds its *M* value is smaller than that of any node in its DNL, it will be elected as a cluster head and change its state into *cluster head*. After that, it will broadcast a *ClusterInvite* packet to its direct neighbors which contains the cluster ID and its *M* value. Another node in the network who receives this *ClusterInvite* packet will reply a *ClusterJoin* packet to node *j* if *j*'s *M* value is smaller than that of any other received *ClusterInvite* packet.

Once a cluster is formed, the cluster head periodically (with the time period being T_c) broadcasts *CHA* packets. Similarly, the cluster members regularly broadcast Cluster Member Announcement (*CMA*) packets containing the cluster member's ID, position, velocity, and direction. Through this way, the cluster head and the cluster members know each other and maintain the cluster. Additionally, because the undecided nodes broadcast *HELLO* packets periodically with T_c until they join in a cluster, the DNL of all nodes should be updated periodically as well.

The details of cluster formation algorithm are shown in Algorithm 1.

Algorithm 1 Cluster formation algorithm

```
Require: node: $j$; cluster set: $\Omega$; DNL set: $\Theta$
Calculate $M(j)$ for node $j$
Send $M(j)$ to all DNL nodes $\gamma \in \Theta(j)$
if $\exists$ cluster head $i \in \Theta(j)$ whose $M$ is the smallest of all cluster heads then
    Add $j$ to $\Omega(i)$
else
    if $M(j) \le \forall M(\gamma)$ then
        $j$ becomes the Cluster Head and sends ClusterInvite to $\Theta(j)$
    else
        $j$ receives ClusterInvite packets and $M(\gamma)$ is the smallest among senders
        Add $j$ to $\Omega(\gamma)$
    end if
end if
Go To Cluster Maintenance
```

5 Cluster maintenance scheme

In most clustering schemes, nodes are only allowed to get together when they are moving in the same direction. Nevertheless, a cluster member, which has to leave the current cluster and join in another one, always falls into the undecided state. Due to every undecided node in the network needing to frequently start the formation algorithm described in 4, more number of undecided nodes will lead to the high network overhead. It is more serious specially at the crossroad scenario when using traditional schemes, even resulting in the degrading or interruption of communications in intra- and inter-cluster. Therefore, a Cluster Head Electing in Advance Mechanism (CHEAM) is proposed to reduce the number of undecided nodes in this section.

Besides, node density is still not taken into consideration in the existing clustering strategies, though the cluster size will be increasing along with the growing node density. To a certain degree, disconnection and congestion in the cluster-based VANETs will occur when the cluster size exceeds the upper limit that the head can bear, which deteriorates the Quality of Service of the communication. On the contrary, it is certain that the cluster size is so small as to waste communication resources. Considered both the cluster size metric and the node density metric, a Cluster Merging and Splitting Mechanism is presented for the purpose of alleviating and avoiding the bad performance of the network caused by the uneven node density.

Furthermore, because of the existing malicious nodes that may interfere with communication and damage network performance, we also design a Discovery and Elimination Scheme (DES) mechanism to avoid malicious nodes and to protect users' privacy.

Finally, a secure cluster maintenance algorithm with load balancing and low overhead is introduced based on the above mechanisms.

5.1 Cluster Head Electing in Advance Mechanism

5.1.1 Stay time prediction

Before proposing CHEAM, we first introduce how to predict the ideal stay time of a cluster member in its cluster.

The stay time of a cluster member plays a key role in our cluster maintenance algorithm. Therefore, we study the stay time prediction problem before presenting the cluster maintenance algorithm.

Assuming v_i^{Ins}, v_j^{Ins}, $\left(x_i^{\text{Ins}}, y_i^{\text{Ins}}\right)$, and $\left(x_j^{\text{Ins}}, y_i^{\text{Ins}}\right)$ are the instantaneous velocities and positions of the cluster head i and its member j which are contained in the *CHA* and the *CMA* packets, respectively, the instantaneous distance between head i and member j can be represented by:

$$D^{\text{Ins}} = \sqrt{\left(x_i^{\text{Ins}} - x_j^{\text{Ins}}\right)^2 + \left(y_i^{\text{Ins}} - y_j^{\text{Ins}}\right)^2}. \tag{13}$$

Comparing the position and the velocity of the head i with those of the member j, four different stay time prediction results of member j in cluster i, $T_{j,i}^{\text{stay}}$ can be obtained,

$$T_{j,i}^{\text{stay}} = \begin{cases} \frac{R+D^{\text{Ins}}}{v_j^{\text{Ins}}-v_i^{\text{Ins}}} & \text{if head } i \text{ is in front of member } j \text{ and } v_i^{\text{Ins}} < v_j^{\text{Ins}} \\ \frac{R+D^{\text{Ins}}}{v_i^{\text{Ins}}-v_j^{\text{Ins}}} & \text{if head } i \text{ is in front of member } j \text{ and } v_i^{\text{Ins}} > v_j^{\text{Ins}} \\ \frac{R-D^{\text{Ins}}}{v_j^{\text{Ins}}-v_i^{\text{Ins}}} & \text{if head } i \text{ is behind member } j \text{ and } v_i^{\text{Ins}} < v_j^{\text{Ins}} \\ \frac{R-D^{\text{Ins}}}{v_i^{\text{Ins}}-v_j^{\text{Ins}}} & \text{if head } i \text{ is behind member } j \text{ and } v_i^{\text{Ins}} > v_j^{\text{Ins}}, \end{cases} \tag{14}$$

where R is the communication radius of a mobile node.

5.1.2 CHEAM

Considering the definition of the predicted stay time and the hopping cluster member mentioned before, the detail of CHEAM can be described as below.

The main idea of CHEAM is to select the most stable (optimal) head for the hopping cluster member as a substitute in advance. In this procedure, the scheme needs to detect the direction and predict the stay time of all members in the cluster. Additionally, the substitute head could be the current head if there are no other candidates with smaller M. Once a substitute is selected, the hopping cluster member hops into the new cluster and becomes a cluster member. Through this way, the number of the undecided nodes can be significantly reduced so that the cluster-based network overhead is minimized. Accordingly, the CHEAM is introduced in Algorithm 2.

Algorithm 2 Cluster Head Electing in Advance Mechanism

```
Require: cluster member: $j$; cluster head: $i$; undecided node: $u$; cluster set: $\Omega$; DNL set: $\Theta$
Each $j \in \Omega(i)$ calculates $T_{j,i}^{stay}$ and starts TimerS as soon as $j$ joins in the cluster of $i$.
if TimerS expires or $\theta_{j,i} > \pi/4$ then
    $j$ becomes a hopping cluster member and executes Cluster Formation.
end if
```

As mentioned in Section 4, the node j will start TimerS and compute the predicted stay time as soon as it joins in the cluster i. In Algorithm 2, j becomes a hopping cluster member and executes the formation algorithm if the TimerS expires or the direction between j and i is larger than $\pi/4$.

In one word, the hopping cluster members are always ready to shift from one cluster to another. The member will start the cluster head selection procedure when it changes into a hopping state.

5.2 Cluster Merging and Splitting Mechanism

5.2.1 Cluster merging

The Cluster Merging Mechanism is used to combine two nearby clusters whose cluster size metrics are smaller than 0.5. Therefore, the waste of network resources can be obviously avoided by the integrated cluster.

Actually, the Cluster Merging Mechanism is started when cluster head i detects that $k_d(i) > 1$ and $k_s(i) < 0.5$. Then i broadcasts a *CMerge* packet to its direct neighbors in DNL so as to request merging. Once another cluster head i' replies to the request, these two clusters will be combined and a new cluster head with the lowest M should be selected. The Cluster Merging Mechanism is illustrated in Algorithm 3.

Algorithm 3 Cluster Merging Mechanism

```
Require: cluster head: $i$, $i'$; DNL set: $\Theta$
Calculate size metric $k_s(i)$ and density metric $k_d(i)$ of head $i$
if $k_d(i) > 1$ and $k_s(i) < 0.5$ then
    Head $i$ broadcast a CMerge packet to its neighbors in $\Theta$
    if $i$ receives the response of CMerge from head $i'$ then
        for The members in both cluster $i$ and $i'$ do
            Execute Cluster Formation.
        end for
    end if
end if
```

Algorithm 4 Cluster Splitting Mechanism

```
Require: cluster head: $i$; cluster member: $j$
if $k_d(i) \geq k_s(i) > 1$ for $\forall$ cluster head $i$ then
    Head selects $n = \lceil k_s(i) \rceil + 1$ candidates with lower $M$
    Head broadcasts CHT packets to $n$ candidates
    Candidates broadcast TCHInvite packets to other members
end if
for $\forall$ member $j$ who receives TCHInvite or ClusterInvite do
    Sequence received heads from 1 to $N_{received}$
    Generate $N_{random} \in [1, N_{received}]$ randomly
    Select and join in a new cluster with head $N_{random}$
end for
```

5.2.2 Cluster splitting

The main idea of our Cluster Splitting Mechanism is that the cluster head will start to inform some members to become new cluster heads when it meets all the following conditions: (1) the cluster size is beyond the maximum value and (2) the node density in the radius range of the cluster head is high. Thus, these new heads will invite neighbor nodes into their own clusters based on *Cluster Formation*. In this way, the average size of clusters can be reduced and the problem that a cluster head with high communication overhead resulted from high node density scenario can be easily solved.

Specifically, when the node density and size metric of cluster head i satisfy $k_d(i) \geq k_s(i) > 1$, head i may search for $n = \lceil k_s(i) \rceil + 1$ candidate nodes with a lower M value than others in its cluster and send Cluster Head Transformation (*CHT*) packets to them. Here, $\lceil \cdot \rceil$ represents the ceiling function. Those nodes who receive *CHT* packets will change their current state into a cluster head and broadcast a transformed cluster head invitation, *TCHInvite* packets, to cluster members as soon as possible. Obviously, a cluster member can receive N_{received} packets whether they are *TCHInvite* or *ClusterInvite* and sequence these received heads from 1 to N_{received}. In order to select any one of heads to join in, members may generate a random integer N_{random} among $[1, N_{\text{received}}]$ and join into this new cluster with head N_{random}.

Algorithm 4 shows the Cluster Splitting Mechanism.

5.3 Discovery and Elimination Scheme

In this section, a malicious node is assumed as a node which only greedily occupies the network resources by sending packets too frequently, e.g., bandwidth or the forwarding time slot. There are also some external attackers attempting to paralyze both network and cluster, by means of spreading viruses all over the network or installing trojans on nodes without permission. We assume that the adversary models of both malicious nodes and external attackers cannot overhear, eavesdrop, or even tamper with the plaintext of other nodes. Nevertheless, we should still prevent these malicious nodes and attackers from damaging the network and design a security protocol, the Discovery and Elimination Scheme (DES), to protect the privacy of cluster nodes.

Assume that all cluster heads are considered as managers which are aware of their cluster members' IDs, while the members in the cluster cannot know the ID of each other. Therefore, the procedure of DES is as below.

- *Generate and broadcast hash ID*: Every packet broadcasted by cluster members must include a control word, which is a hash value returned by a certain hash function, e.g., the nonreversible SHA hash function, from the ID of this cluster member.
- *Verify hash ID and forward data*: When the cluster head receives a packet, it compares the hash(ID) control word to its own hash table immediately. In other words, the cluster head will consider the transmitter as its member and forward the packet to the destination, if the hash(ID) in the packet matches one of the items in the head's hash table. Particularly, once a cluster member with hash(ID), occupying a high proportion of resources, is detected, it is deemed as a malicious node which should be expelled out of the current cluster for the purpose of avoiding the congestion.
- *Communication among members*: Due to the lack of other members' ID information, a cluster member receives a packet that must be a plaintext so that it can know the content of the packet rather than the source. Thus the members' privacy can be protected.

5.4 Maintenance scheme

So far, the details of every mechanism are described above; we now propose the cluster maintenance scheme introduced as below.

Algorithm 5 Cluster Maintenance Scheme

```
Require: cluster member: $j$; cluster head: $i$; cluster set: $\Omega$;
    DNL set: $\Theta$
 1: Start **Discovery** & **Elimination Scheme** during data
    transmission
 2: Each node broadcasts **CHA**, **CMA** or **HELLO** packets
    periodically with $T_c$
 3: Recalculate $M$ for every nodes in cluster $i$ periodically
 4: if $M(j) < M(i)$ then
 5:     Replace the cluster head with a member as a new
        head
 6: end if
 7: Execute **CHEAM** described in Algorithm 2 as soon as
    a node joins in a cluster
 8: Head $i$ periodically calculates $k_d(i)$ and the $k_s(i)$
 9: if $0 < k_s(i) \leq k_d(i)$ or $k_d(i) > 1 \& 0.5 \leq k_s(i) \leq 1$
    then
10:     Head $i$ maintains itself based on LOSC proposed in
        [20]
11: else if $k_d(i) > 1 \& k_s(i) < 0.5$ then
12:     Execute **Cluster Merging Mechanism** described
        in Algorithm 3
13: else if $k_d(i) \geq k_s(i) > 1$ then
14:     Execute **Cluster Splitting Mechanism** described
        in Algorithm 4
15: end if
```

In Algorithm 5, the *Discovery and Elimination Scheme* is run as long as there exists data transmission during the maintenance, for the purpose of protecting the cluster communication. After receiving the broadcast packets, we should recalculate the capability metric to identify which node should be changed into the cluster head so as to maintain cluster stability. In order to keep load balancing, Algorithm 5 also introduces *Merging* and *Splitting* from lines 8 to 15, which is based on the determination of node density and cluster size.

6 Simulation and discussion

In this section, we first carry out an extensive simulation study on MATLAB platform to evaluate the performance of the proposed scheme in a crossroad scenario. The LOSC scheme in [20] and a variant of the Lowest-ID algorithm called MOBIC clustering algorithm in [10] are also implemented as a comparison with our scheme. The following subsection is to analyze the security of EnLOSC.

6.1 Numerical evaluation

The simulation scenario is a two-lane crossroad as shown in Fig. 1. The communication between two vehicles follows the free-space path loss: $\mathrm{FSPL} = \left(\frac{4\pi df}{c}\right)^2$, where f is the signal frequency, c is the speed of light in a vacuum, and d is the distance between the transmitter and the receiver. Without loss of generality, the transmitting and receiving antenna gain are assumed to be 1, and the communication radius is 100 m. Besides, the number of vehicles N_n is set up from 50 to 200, and the vehicle speed is selected randomly between 30 and 50 km/h.

In Fig. 4, we illustrate the performance of Cluster Merging and Splitting Mechanism on cluster size controlling. The maximum number of cluster members in EnLOSC is set to be 30 in our simulation. As depicted, the average cluster size under LOSC and MOBIC schemes grows significantly by the increase of the number of vehicles in the whole network, while this argument under the EnLOSC algorithm has a little change by a different number of vehicles. In comparison with LOSC and MOBIC schemes, the result in the figure reports the better performance of controlling the cluster size by merging and splitting mechanism in EnLOSC under either high or low density scenarios. In other words, the overhead of cluster heads by high node density and the waste of resources resulting from low node density can be significantly cut down when using EnLOSC.

Figure 5 shows the performance of cluster stability that is represented by the average number of cluster heads changing per second. Intuitively, we believe that the cluster-based network is not stable if this average value is large because of the frequent head handoff. Depicted in Fig. 5, the number of head changing in the LOSC scheme is lower than that of MOBIC with the various number of vehicles, which means the stability of clusters formed by our previous algorithm in [20] is better than that formed by MOBIC. Obviously, the cluster stability of the proposed EnLOSC is slightly higher than LOSC. The reason for the increasing number of cluster head shifting in EnLOSC is the cost of keeping load balancing for every cluster-by-cluster merging and splitting schemes. Besides, it can be also inferred that this average value in MOBIC is vulnerable to the large number of vehicles. In other words, compared with MOBIC, both LOSC and EnLOSC schemes are more suitable for the crossroad scenario in VANETs.

For the purpose of illustrating the communication overhead, we explore the average number of undecided nodes which is calculated for the duration of periodic broadcasting T_c. During each T_c, every node in a cluster broadcasts either a *CHA* or a *CMA* packet. In contrast to MOBIC, the result in Fig. 6 reports that the previous work in [20] and our enhanced scheme achieve

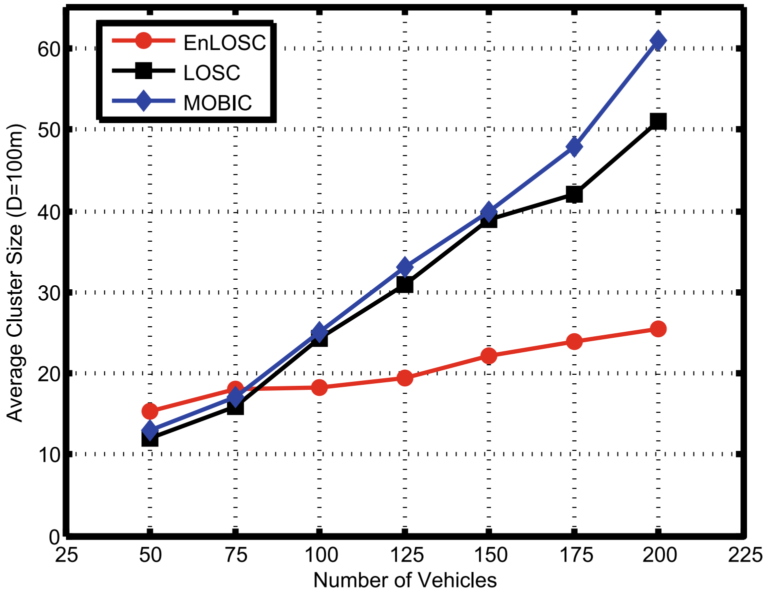

Fig. 4 Analysis of cluster size

great improvements on reducing the average number of undecided nodes due to the proposed CHEAM. Moreover, because of the scattered cluster coverage in VANETs and the restricted communication radius, there may exist some undecided nodes between two clusters. The smaller clustering structure resulting from the cluster splitting schemes in EnLOSC will narrow those gaps, which can reduce the probability of those nodes not being in any one cluster. Accordingly, the congestion and overhead caused by the undecided nodes can be cut down significantly in the crossroad when using both LOSC and EnLOSC.

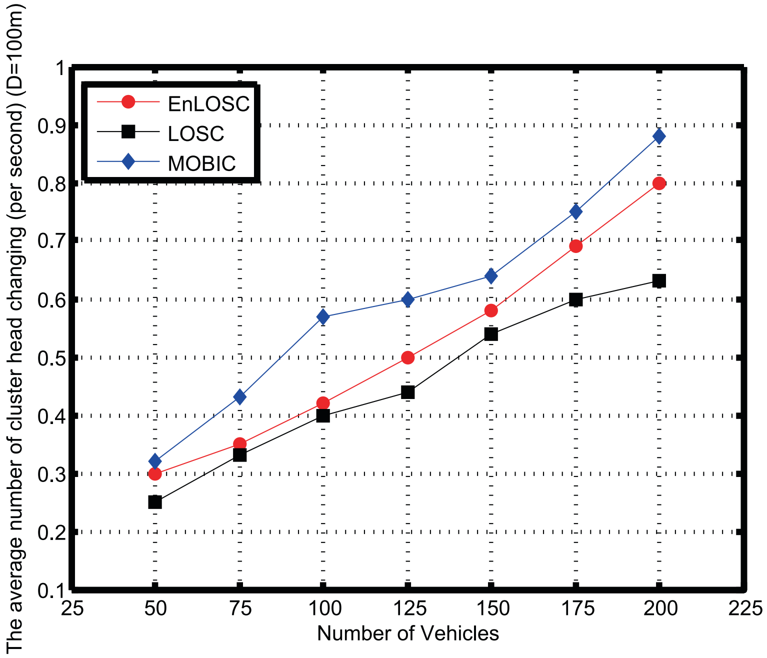

Fig. 5 Analysis of cluster stability

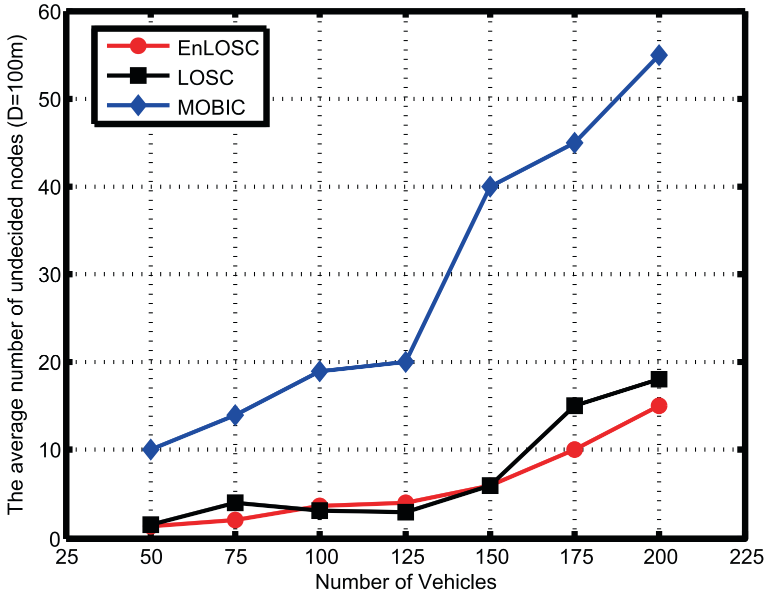

Fig. 6 Analysis of undecided nodes

6.2 Security analysis

According to the rule of DES, every node in one cluster, which wants to transmit its own data, should add a hash(ID) to the plaintext. Because the cluster head has the existing hash table including all the members' hash(ID), it knows the source of each packet. In this subsection, we investigate the security issues of our proposed DES from the following aspects.

Firstly, if there exists some external attackers, who want to make cluster paralysis, they have to follow this protocol. Since the attackers' IDs are not stored at the cluster's hash table, if the attackers broadcast the message, they cannot pass the cluster's verification and the packets will be dropped. Meanwhile, the cluster head can also refuse to forward a member's data, if this member does not follow the protocol, for example, the member does not provide its hash(ID) to the cluster head, who can decline the request.

Secondly, when a cluster member sends packets to the head constantly, the network resource will be occupied exclusively, which represents the head. Thus, the cluster head, as a relay, cannot transmit other members' data. Within this situation, to overcome the problem, the head can set up different thresholds of the resource utilization for each member and then will calculate the percentage of each member in terms of the resource utilization based on the number of hash(ID). If the member has a high frequency of hash(ID) than the threshold, who is called as a malicious node, the malicious node will be denied providing relay service.

Thirdly, owing to the broadcast packets with hash(ID), which hides the real ID of each member, the DES also provides simple anonymous communication. That means that other members cannot know which member broadcasts those packets.

What's more, even if we introduce the DES, the computation of proposed clustering maintenance scheme has not increased significantly because of the low complexity hash algorithm.

7 Related work

Cluster head selection plays an important role in clustering algorithms. Various metrics have been proposed to describe a vehicle's capability of functioning as a cluster head in VANETs. In this section, we briefly review the work related to clustering methods that are based on these different metrics.

In [16], the metric is defined by considering the traffic flow on the lane. A vehicle on the lane with the most traffic flow is selected as the cluster head. The clustering algorithm proposed in [17] defines the metric as a function of the path loss. A vehicle with a smaller path loss from other vehicles has a higher metric value. It is concluded in [14] that the performance of cluster-based communication can be further improved by exploring the geography information for cluster head selection. Based on this conclusion, [13] and [21] combine the geography information together with the traffic information and the task information to define their metrics. To further extend the reliable clustering method in highway scenarios in

VANETs, Ibrahim exploited dense traffic to design a CASCADE scheme in [22] in order to enable both safety (collision warning) and information (congestion notification) applications.

The aforementioned clustering schemes usually cause frequent re-affiliation and cluster head changes since they do not consider the effects of fast movements of vehicles in VANETs. To solve these problems, mobility-based clustering algorithms are put forward. In [15], Song et al. use the moving direction of vehicles together with the location information to design a clustering algorithm. Only the vehicles moving in the same direction can form clusters and the cluster head is selected according to the location information. In [11], Basu et al. designs a mobility metric by measuring the fluctuation of a vehicle's received power during successive transmissions. A vehicle with a smaller fluctuation is considered as a vehicle with a smaller relative speed with others, and it is more likely to be selected as a cluster head. The performance of this scheme degrades significantly when the vehicle's speed varies sharply and frequently due to the fact that the vehicle's acceleration is not considered in the mobility metric. Basu et al. [10] solves this problem by designing a mobility metric consisting of both the relative velocity and the relative acceleration to represent a vehicle's ability to be the cluster head. Besides, a metric called the Aggregate Local Mobility (ALM) measure is considered to design a criterion triggering cluster re-organization strategy with a contention-based scheme in [17]. However, the metric is only defined by the relative mobility calculated through the current and previous distances between a node and its neighbor.

To the best of our knowledge, these existing schemes we referred to consider either the highway scenarios or the straight-lane scenarios. None of them considers the complicated and challenging crossroad scenarios where large numbers of vehicles can become isolated, and thus, considerable communication overhead and network congestion can be generated using these existing schemes. In this paper, we tackle the challenge of designing a low overhead and stable clustering scheme for crossroads in VANETs.

8 Conclusions

Based on our previous studies in [20], we present an Enhanced LOSC that focuses on not only the stability and network overhead but also the load balancing and security in the crossroad scenario in this paper. A new capability metric M, which is related to the relative velocity and the power loss, is introduced to describe a node's capability of being a cluster head and exploited in the maintenance algorithm to achieve the cluster head electing. Meanwhile, in order to maintain load balancing of the head in a clustering-based network, we also use other metrics expressed by node density and cluster size to adjust the number of nodes in a cluster. Furthermore, the proposed security method, DES, can protect the security of the cluster from attackers and malicious nodes and also provide a simple anonymous communication to preserve nodes' privacy. Compared with the existing MOBIC clustering algorithm and the previous LOSC scheme, the simulation results show that there are less isolated nodes in VANETs by using the EnLOSC scheme, which can ensure the more stable and load balancing clusters. For future research, we will consider how to design a secure strategy for more complex VANETs to deal with the wiretapping problem caused by some eavesdroppers.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

We are very grateful to all reviewers who have helped improve the quality of this paper. Thanks also to Beijing Review copyeditor Yu Nan for improving the language. This work was supported by the National Natural Science Foundation of China (Grant No. 61172074 and 61471028), the Fundamental Research Funds for the Central Universities (2015JBM016), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No.20130009110015), the NSF awards CNS-1352726, and the financial support from the China Scholarship Council.

Author details

[1]School of Electronics and Information Engineering, Beijing Jiaotong University, 100044 Beijing, China. [2]Department of Computer Science, The George Washington University, 20052 Washington, DC, USA. [3]Department of Computer Science, Texas Christian University, 76129 Fort Worth, TX, USA.

Received: 20 November 2015 Accepted: 24 February 2016
Published online: 03 March 2016

References

1. JB Kenney, Dedicated short-range communications (DSRC) standards in the United States. Proc. IEEE. **99**(7), 1162–1182 (2011)
2. G Xin, H Yan, C Zhipeng, O Tomoaki, Intersection-based forwarding protocol for vehicular ad hoc networks. Telecommun. Syst, 1–10 (2015). doi:10.1007/s11235-015-9983-y
3. Z Rawashdeh, S Mahmud, A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. EURASIP J. Wirel. Commun. Netw. **2012**(1), 15 (2012)
4. W Fan, Y Shi, S Chen, L Zou, in *IET International Conference on Communication Technology and Application*. A mobility metrics based dynamic clustering algorithm for VANETs, (2011), pp. 752–756. doi:10.1049/cp.2011.0769
5. S Vodopivec, J Bester, A Kos, in *Telecommunications and Signal Processing (TSP) 2012 35th International Conference On*. A survey on clustering algorithms for vehicular ad-hoc networks, (2012), pp. 52–56. doi:10.1109/TSP.2012.6256251
6. L Guo, C Ai, X Wang, Z Cai, Y Li, in *Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th International*. Real time clustering of sensory data in wireless sensor networks, (2009), pp. 33–40. doi:10.1109/PCCC.2009.5403841
7. C Shea, B Hassanabadi, S Valaee, *Mobility-based clustering in VANETs using affinity propagation* (IEEE, 2009), pp. 1–6
8. W Chen, S Cai, Ad hoc peer-to-peer network architecture for vehicle safety communications. IEEE Commun. Mag. **43**(4), 100–107 (2005). doi:10.1109/MCOM.2005.1421912
9. R Ramanathan, M Steenstrup, Hierarchically-organized, multihop mobile wireless networks for quality-of-service support. Mob. Netw. Appl. **3**(1), 101–119 (1998). doi:10.1023/A:1019148009641

10. P Basu, N Khan, TDC Little, in *International Conference on Distributed Computing Systems Workshop*. A mobility based metric for clustering in mobile ad hoc networks, (2001), pp. 413–418. doi:10.1109/CDCS.2001.918738
11. Y Gunter, B Wiegel, HP Grossmann, in *IEEE Intelligent Transportation Systems Conference*. Cluster-based medium access scheme for VANETs, (2007), pp. 343–348. doi:10.1109/ITSC.2007.4357651
12. M Sood, S Kanwar, in *Communication and Information Technology Applications (CSCITA)*. Clustering in MANET and VANET: a survey.in International Conference on Circuits, Systems, (2014), pp. 375–380, doi:10.1109/CSCITA.2014.6839290
13. T Song, W Xia, T Song, L Shen, in *IEEE International Conference on Communication Technology (ICCT)*. A cluster-based directional routing protocol in VANET, (2010), pp. 1172–1175. doi:10.1109/ICCT.2010.5689132
14. Y Harikrishnan, J He, in *International Conference on Computing Networking and Communications (ICNC)*. Clustering algorithm based on minimal path loss ratio for vehicular communication, (2013), pp. 745–749, doi:10.1109/ICCNC.2013.6504181
15. RA Santos, RM Edwards, NL Seed, in *International Workshop on Mobile and Wireless Communications Network*. Using the cluster-based location routing (CBLR) algorithm for exchanging information on a motorway, (2002), pp. 212–216. doi:10.1109/MWCN.2002.1045724
16. Z Wang, L Liu, M Zhou, N Ansari, A position-based clustering technique for ad hoc intervehicle communication. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **38**(2), 201–208 (2008). doi:10.1109/TSMCC.2007.913917
17. E Souza, I Nikolaidis, P Gburzynski, in *IEEE International Conference on Communications (ICC)*. A new aggregate local mobility (alm) clustering algorithm for VANETs, (2010), pp. 1–5. doi:10.1109/ICC.2010.5501789
18. B Zhou, Z Cao, M Gerla, in *International Conference on Wireless On-Demand Network Systems and Services, 2009*. Cluster-based inter-domain routing (CIDR) protocol for MANETs, vol. 2009 (WONS, 2009), pp. 19–26, doi:10.1109/WONS.2009.4801843
19. Y Huang, M Chen, Z Cai, X Guan, T Ohtsuki, Y Zhang, in *Global Communications Conference (GLOBECOM), 2015*. Graph theory based capacity analysis for vehicular ad hoc networks (IEEE, 2015)
20. Y Huo, Y Liu, X Xing, X Cheng, L Ma, T Jing, in *Wireless Algorithms Systems, and Applications (WASA)*. A low overhead and stable clustering scheme for crossroads in VANETs, vol. 2015, (2015), pp. 232–242
21. Y Luo, W Zhang, Y Hu, in *International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*. A new cluster based routing protocol for VANET, vol. 1, (2010), pp. 176–180, doi:10.1109/NSWCTC.2010.48
22. K Ibrahim, MC Weigle, in *GLOBECOM Workshops, 2008*. Cascade: cluster-based accurate syntactic compression of aggregated data in VANETs (IEEE, 2008), pp. 1–10, doi:10.1109/GLOCOMW.2008.ECP.59