A Novel Mobility-Based Clustering Algorithm for VANETs

1 Mingming CHEN, 2 Fan YANG
Department of Communication Engineering, Xiamen University, Xiamen 361005, China
E-mail: xmumingming@gmail.com, yfh1101@126.com

Received: 16 May 2014 /Accepted: 31 July 2014 /Published: 31 August 2014

Abstract: Due to the development of communication and other related technologies, vehicular ad hoc networks (VANET) have been experiencing a rapid development in the recent years. Clustering has been proved an effective way to manage the wireless resource, control the network topology, and simplify the routing in VANETs. However, it is difficult to construct stable clusters in VANETs, because of the high mobility of vehicles and poor quality of wireless channels. In this paper, we propose a novel mobility-based clustering algorithm (MBC) for VANETs, in order to enhance the stability of clusters. Different from the existing work, this algorithm is tailored for VANETs. It takes the direction, speed difference and position of vehicles into consideration, when selecting the cluster head. Simulation results show that the proposed clustering algorithm significantly increases the average cluster head lifetime, and reduces the average number of clusters changed per vehicle during the simulation.

Keywords: Clustering algorithm, Vehicular ad hoc network, VANET, MANET, Clusters.

1. Introduction

As part of Intelligent Transportation Systems (ITS), vehicular ad hoc networks have received increased interests from both academia and industry recently. Applications in VANETs can be classified into the following categories: safety-related applications (e.g., emergency or incident warning, lane change assistance, and intersection management), traffic management and transportation efficiency (e.g., toll collection and intersection management), user infotainment services and the Internet connectivity [1]. Different applications have different quality-of-service (QoS) requirements. Generally, safety-related applications have a very strict demand of delay, while non-safety applications require a fair access but can tolerate more transmission delay. In VANETs, each vehicle is equipped with an on-board unit (OBU), which is a wireless interface to communicate with other vehicles as well as roadside units (RSU). Roadside units are stationary network nodes along the road, and they provide the access to the Internet. They are usually deployed on the transportation infrastructure, such as street lights and road signs [1]. Thus, communications in VANETs consist of two modes: vehicle-to-vehicle (V2V) and vehicle-RSU (V2R).

Compared to other ad hoc networks, vehicular ad hoc networks are characterized by high node mobility, frequent link breakages and topology changes. Due to these drawbacks, it is of great challenge to design an efficient MAC scheme to meet different QoS demands of all applications. In addition, it is very difficult to utilize the wireless resource in VANETs, due to the rapid changes of topology. In addition, the exchange of topology will introduce high communication overhead [2]. Different medium access control (MAC) protocols and topology...
management schemes have been proposed in the literature to deal with these problems. Among all these schemes, clustering has received much attention recently. The hierarchical structure of clusters has been proved to be effective to provide QoS, manage network bandwidth, and simplify routing. Some clustering algorithms have been proposed in the literature [3-11] to construct stable clusters in VANETs.

For clustering schemes, how to keep the stability of clusters is a challenging task, especially in VANETs with high node mobility and frequent topology changes. Frequent changes of cluster head and members introduce a large overhead in both MAC and upper-layer protocols. Further, it also influences the performance of the whole network heavily. Several clustering algorithms for VANETs have been proposed in the literature. However, most of them are derived from mobile ad hoc networks (MANET). Mobility features of vehicles are not considered in these algorithms. Thus, they can not produce the most optimal results for VANETs. In this paper, we propose a novel algorithm to form stable clusters in VANETs. Different from the existing work, the proposed algorithm is tailored for VANETs. This algorithm takes the speed difference, position and direction of vehicles into consideration, when selecting the cluster head. In addition, speed difference is also considered, when joining a cluster, so that the cluster member can stay in the cluster for a longer time.

The remainder of this paper is organized as follows. In Section 2, related works are presented. Section 3 describes the proposed clustering scheme. Section 4 shows and discusses the simulation results. This paper concludes with Section 5.

2. Related Works

Many clustering algorithms have been designed for mobile ad hoc networks (MANET) in the literature. Based on these algorithms, some clustering schemes have been proposed for VANETs. However, they do not consider the unique features of VANETs, such as the mobility pattern of vehicles, sufficient energy and etc. Therefore, they can not produce stable clusters in VANETs.

One of the most frequently mentioned clustering algorithm is MOBIC [3]. This algorithm was originally designed for MANETs. But, it can also works in VANETs. It is based on the well-known lowest-ID algorithm. In the lowest ID algorithm, each node is assigned a unique ID, and the node with the lowest ID in its neighborhood is selected as the cluster head. This algorithm does not consider the mobility of nodes. In order to represent the mobility of nodes, mobility metric is proposed in [3]. The mobility metric for selecting a cluster head is based on the ratio between two successive measurements of the received power at any node from its neighbors. Its performance is moderate, and is often used for comparison with other VANET clustering algorithms.

In [4], a direction based algorithm is proposed for urban area. According to the predefined path and the position of a vehicle, clusters are formed before the road intersection. However, this algorithm can only be used in some special situations of VANETs, and does not consider the mobility of vehicles. In [5], a cluster-based location routing (CBLR) is presented. Nodes use HELLO messages to distribute their states. When a node enters the system, it enters the undecided state and then announces itself as a cluster head if it does not receive a HELLO message within a period of time from other nodes; otherwise, it registers as a member node. In [6], the authors modified the algorithm used in CBLR. When two cluster heads come into each other’s communication range, a cluster should be dismissed and merge with other clusters or generate a new cluster. A metric for making the decision is proposed, which takes into consideration the mobility, the connectivity and the distance. These two algorithms use the basic method to form clusters, and the cluster head is elected voluntarily. Thus, they can not produce stable clusters in VANETs. In [7] and [8], the authors propose two similar cluster-based multi-channel MAC protocols for QoS provisioning. Traditional algorithms are adopted for the cluster formation. If there are no cluster heads around a node, then it announces itself as a cluster head; otherwise, it joins one of these clusters. When two cluster heads come into each other’s communication range, the cluster with less members is dismissed. These two papers focus on the design of MAC protocol, and the proposed clustering algorithms also do not consider the mobility of vehicles.

Authors of [9] present a new beacon-based clustering algorithm. It uses the position information contained in two successive hello messages to elect the cluster head on cluster reformation. In [10], the authors present a mobility-based clustering scheme, which forms clusters using the Affinity Propagation algorithm in a distributed manner. However, the speed difference is not considered in this paper. In [2], a speed based clustering method for highways is presented. Neighbors of a node are divided into two groups: stable and unstable neighbors. Only nodes in stable neighbors can form a cluster. Authors of [11] propose a clustering and OFDMA based MAC protocol. A metric for measuring the stability of a node is proposed and the cluster head is elected based on this metric. Further, it uses the fuzzy logic to predict the future speed and positions of all cluster members, in order to increase the stability of a cluster. However, the overhead of this protocol is high, and the accuracy of prediction is uncertain.

3. Clustering Algorithm

In this section, we discuss the detail operation of the clustering algorithm, including the scenario and
3.1. Scenario and Assumptions

We consider a multi-lane highway scenario as shown in Fig. 1. Each vehicle is assigned a unique ID, based on its MAC address. Each vehicle on the road is equipped with a GPS receiver that determines its position and speed. Each node (the terms “node” and “vehicle” are used interchangeably throughout this paper) maintains a list of its one-hop neighbors, and periodically broadcasts a HELLO message that contains its current direction, position and speed to its neighbors. The interval between two consecutive broadcasts is \( T \). All the neighbors’ information is stored in the list for a certain time. If a node does not hear any information from a neighbor for a while, it will delete the neighbor from the list. To increase the stability of clusters, only nodes with the same movement direction can be in a cluster. The cluster head is selected based on its neighbors’ movement information. In each cluster, a node is elected to be the cluster head (CH) and the other nodes are cluster members (CM).

3.2. Cluster Head Election Algorithm

Each node measures and updates the movement information of itself and the neighboring nodes every \( T \) period. In order to increase the stability of a cluster, two key factors, speed difference and position are considered, when selecting the cluster head. For node \( i \), let \( v_i \) denote its speed, and \( p_i \) its position. Then, a function \( pr_i = g(v_i, p_i) \) is defined to compute the priority that node \( i \) is selected as the cluster head. Every node computes its priority, and broadcasts the priority to its neighbors along with the movement information every \( T \) period. The node with the smallest value of \( pr_i \) in its neighborhood is given the highest priority to be selected as the cluster head.

3.2.1. Speed Difference

Vehicles moving on the road have different speed. Further, the speed of a car changes with time. The node with a closer speed to its neighbors should be given a higher priority to be selected as the cluster head, so that it can be the cluster head for a longer time. Here, we adopt the exponential-weighted average of speed over a period \( T \), instead of the instant value, to calculate its priority. For node \( i \), let \( v_n^i \) denote the exponential-weighted average speed. Let \( v^i \) denote the new value of speed, \( v_{n-1}^i \) the previous of speed. Then,

\[
    v^i_n = (1 - \alpha) v_{n-1}^i + \alpha v^i_n
\]

where \( 0 \leq \alpha \leq 1 \) is the smoothing factor and chosen here to be 0.5.

To calculate its priority, each vehicle needs to know how close its speed is to the average speed of its neighbors. For node \( i \), Let \( u_{v_i} \) denote the average speed of its neighbors. Then, \( u_{v_i} \) is calculated as

\[
    u_{v_i} = \frac{\sum_{j=1}^{N} v_{w_j}}{N},
\]

where \( v_{w_j} \) is the speed of any neighbor, \( N \) is the number of the neighbors of node \( i \).

Then, the speed difference between node \( i \) and the average speed of its neighbors can be calculated. Here, in order to avoid that one parameter dominates the result of the priority, normalization technique is adopted. For node \( i \), let \( v_{max} \) denote the maximum speed of its neighbors. Then, the normalized speed difference between node \( i \) and the average of its neighbors is calculated as

\[
    v_{norm_i} = \frac{v^i_n - u_{v_i}}{v_{max} - u_{v_i}}.
\]

3.2.2. Position

Node position is another factor that determines the stability of a cluster. The node with the closer distance to its neighbors should be given the higher priority to be selected as the cluster head, because the probability that it leaves the cluster is smaller. Let \( (x_i, y_i) \) denote the position coordinate of node \( i \), \( (x_j, y_j) \) the position coordinate of any neighbor \( j \) of node \( i \). Then, the mean position of the neighbors of node \( i \) is
distance between two neighboring CHs is larger than 0.6 here.

Let \( (x_{\text{max}}^i, y_{\text{max}}^i) \) denote the node position that has the longest distance to the mean position of the neighbors of node \( i \). Then, the normalized distance \( d_{\text{norm}}^i \) of node \( i \) with the mean position of its neighbors is

\[
d_{\text{norm}}^i = \frac{\sqrt{(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2}}{\sqrt{(x_{\text{max}}^i - \bar{x}_i)^2 + (y_{\text{max}}^i - \bar{y}_i)^2}}.\tag{5}
\]

Finally, the priority \( p_r \) can be calculated as

\[
p_r = W_1 p_{\text{norm}}^i + W_2 v_{\text{norm}}^i, \tag{6}
\]

where \( W_1 \) and \( W_2 \) are the weights for \( p_{\text{norm}}^i \) and \( v_{\text{norm}}^i \) respectively. \( W_1 \) and \( W_2 \) are the set to be 0.4 and 0.6 here.

### 3.3. Clustering Process

The clustering scheme includes 6 processes: initialization, cluster head electing process, joining process, losing contact with the CH temporarily, leaving process and merging process.

1) Initialization: When a vehicle enters the highway, it is initialized as the isolation node.

2) Joining process: Every CH broadcasts the invitation message (IM) together with its movement information every \( T \) period. Once an isolation node receives an IM, it checks whether the CH’s direction is the same with itself. Then, it checks whether the speed difference between the CH and itself is within \( \pm \Delta V_{\text{th}} \). If the direction is the same and the difference is within \( \pm \Delta V_{\text{th}} \), it sends a request message (RM) to the CH. After the CH receives the RM message, it sends back an acknowledge (ACK) and accepts the requesting node as a cluster member. If a node receives more than one IMs, it sends the RM to the CH with the closest speed difference.

3) Head electing process: If an isolation node cannot receive any IM during \( T \) period, it starts the head electing process. It gathers the information of its neighboring isolation nodes and the proposed algorithm is used to elect the cluster head as described in section 3.2. The isolation node within a CH’s transmission range \( R_{CH} \) is not allowed to start the head electing process, in order to ensure that the distance between two neighboring CHs is larger than \( R_{CH} \).

4) Losing contact with the CH temporarily: When a CM can not receive the movement information broadcasted by the CH every \( T \) period, the state of this node changes from cluster member to temporary member. It does not leave the cluster immediately, because this disconnection may be due to the poor quality of the wireless link. If the temporary member receives the information broadcasted by the CH again in the next \( mT \) period, the state changes to CM again.

5) Leaving process: When a temporary member cannot receive the CH’ movement information consecutively for \( m \) times, the state of this node changes to the isolation node. Meanwhile, the CH will delete this member from the member list.

6) Merging process: When two cluster heads come into each other’s transmission range, the cluster merging process holds. The cluster head with fewer members gives up the role of cluster head and joins the cluster with more members. The other members of the cluster head either join the neighboring clusters or form a new cluster.

### 4. Simulation Results

To evaluate the performance of MCB algorithm, we compare it with the MOBIC algorithm proposed in [3]. Simulations are performed using Network Simulator (NS2), version 2.35. The simulations are carried out for a 4 lanes highway with a length of 10 km and a width of 10 m per lane. Vehicle velocity varies from 60 to 120 km/h. Vehicles move on the highway according to the freeway mobility model as describe in [12]. The data rate is set to 1 Mbps and the periodic messages are sent every 100 ms. The size of the message including the mobility information is 100 bytes. To study the performance of the proposed algorithm for different cluster size, we used different transmission ranges. The transmission range varies from 200 to 400 m. For media access, we use the IEEE 802.11p [13].

In the simulation, three events can change the state of a vehicle [2].

1) A vehicle leaves its cluster and forms a new one.

2) A vehicle leaves its cluster and joins a nearby cluster.

3) A cluster head merges with a nearby cluster.

Three metrics are defined as follows to evaluate the performance of the clustering algorithms.

1) Average cluster head lifetime (ACHL): The cluster head time is an important metric to show the stability of a cluster. Long cluster head time implies the high stability of clusters. The average cluster head lifetime is defined as the average time period from the moment when a vehicle becomes a cluster head to the lifetime of the time when it is merged by another cluster head.

\[
ACHL = \frac{\sum_{i=1}^{n}(t_{CH}^i) - t_{CH}^i}{n}, \tag{7}
\]
where \( n \) is the number of clusters formed during the simulation time, \( t_i^b \) the time when cluster \( i \) is built, and \( t_i^m \) the time when cluster \( i \) is merged.

2) **Average number of cluster changes (ANCC)**: Due to the three aforementioned events, a vehicle changes its state during the simulation. The average number of clusters changed per vehicle during the simulation can be used to evaluate the stability of clusters.

\[
ANCC = \frac{1}{m} \sum_{i=1}^{m} c_i,
\]

where \( m \) is the number of vehicles, and \( c_i \) is the number of cluster changes of vehicle \( i \) during the simulation.

3) **Number of clusters**: Due to high mobility of vehicles on the road, clusters are created and vanished over time. Thus, the total number of clusters created over the simulation period defines the cluster formation rate. Small number of clusters means the low cluster formation rate, and it shows the clustering algorithm produce the stable clusters.

Fig. 2 and Fig. 3 compare the average cluster head lifetime of MC and MOBIC algorithms. The average cluster head lifetime is significantly increased, when MCB algorithm is adopted. This is because the probability that a CH leaves its cluster decreases, when MCB algorithm is adopted. MCB selects the more stable CH than MOBIC, because speed difference is considered in MCB. In addition, with the increase of transmission range, the average cluster head lifetime increases. This is because more members can join a cluster, when the transmission range increases. Thus, the probability that the cluster is dismissed decreases.

Fig. 4 and Fig. 5 compare the average number of cluster changes of MCB and CB algorithms. The average number of cluster changes per vehicle is decreased significantly, when MCB algorithm is adopted. This is because the probability that a cluster is dismissed decreases, when MCB is adopted. Thus, the time when a cluster member stays in the same cluster is longer than that of MOBIC. In addition, with the increase of transmission range, the average number of cluster changes becomes smaller. This is because the probability that a cluster member leaves its cluster decreases, when the transmission range becomes bigger.

Fig. 6 and Fig. 7 illustrate the number of clusters created by MCB and MOBIC algorithms during the simulation. We can see that the number of clusters created by MCB algorithm is smaller than that of MOBIC algorithm. This is because MCB algorithm creates more stable clusters. Thus, the probability that a cluster is dismissed or a cluster member leaves its cluster decreases. Then, the average lifetime of clusters created by MCB algorithm is longer than that of MOBIC algorithm.
5. Conclusion

In this paper, we propose a mobility-based clustering algorithm for VANETs, aiming to form stable clusters. Speed difference, position and direction of vehicles are taken into consideration in this algorithm. The exponential-weighted moving average of the speed is adopted in this algorithm, in order to accurately describe the movement of vehicles. The simulation results show that the proposed algorithm significantly increases the stability of a cluster, compared to the existing algorithms.

References


