<table>
<thead>
<tr>
<th>Title</th>
<th>Parking backbone: toward efficient overlay routing in VANETs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Zhu, Jinqi; Liu, Ming; Wen, Yonggang; Ma, Chunmei; Liu, Bin</td>
</tr>
<tr>
<td>Date</td>
<td>2014</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/24062">http://hdl.handle.net/10220/24062</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2014 Jinqi Zhu 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.</td>
</tr>
</tbody>
</table>
**Research Article**

**Parking Backbone: Toward Efficient Overlay Routing in VANETs**

Jinqi Zhu, Ming Liu, Yonggang Wen, Chunmei Ma, and Bin Liu

1 School of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China
2 School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
3 School of Computer Engineering, Nanyang Technological University, Singapore

Correspondence should be addressed to Jinqi Zhu; jingpeiy719@163.com

Received 18 April 2014; Accepted 15 July 2014; Published 12 August 2014

Academic Editor: Nianbo Liu

Copyright © 2014 Jinqi Zhu 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.

In vehicular ad hoc networks (VANETs), message dissemination to one or more vehicles is very challenging, due to frequent network disconnections and uncertain locations of the destination vehicles. Deploying roadside units (RSUs) is a possible solution to overcome the challenges, but it often requires a large amount of investment. In this paper, we propose the idea of Parking Backbone, which does not need any RSUs but leverages a virtual overlay network formed by outside parked vehicles to track vehicles and to disseminate messages between moving vehicles. Our scheme consists of three parts. At first, to each road, parked vehicles both at roadside and off street are grouped into a cluster as far as possible. An urban overlay network is established based on this type of clusters for data transmission. Secondly, to a specific vehicle, a daily mobility model is established, to determine its location through a corresponding location prediction algorithm. Finally, a novel message delivery scheme is designed to efficiently transmit messages to destination vehicles through the proposed virtual overlay network. Thanks to the extensive and stable outside parking in cities, once grouped into the overlay structure, data transmission can be easily achieved over the Parking Backbone. Extensive simulation results prove that our scheme achieves high performance in message dissemination.

1. Introduction

Nowadays, as more and more vehicles are equipped with wireless devices, positioning systems, and digital maps, large-scale vehicular ad hoc networks (VANETs) are emerging as a new technology in the near future. In VANETs, vehicles are allowed to not only exchange messages with each other as vehicle-to-vehicle communication (V2V), but also exchange messages with roadside infrastructures as vehicle-to-infrastructure communication (V2I) [1]. The applications range from driving safety, intelligent transportation as well as information and entertainment applications.

In order to facilitate better road safety and comfort driving, message dissemination among vehicle drivers is fast becoming a requisite to vehicular users. Through vehicular message dissemination, we can check the latest traffic jam data, get traffic alert information, book desired restaurant, deliver announcement, inquire certain services, and have many other useful services, which greatly benefit our daily vehicle trips. However, due to the unique characteristics of VANETs, such as rapid-changing topology, high mobility of vehicle nodes, and intermittent node connectivity, how to efficiently disseminate message in VANETs becomes very challenging. Towards solving the problem, many existing works [2, 3] utilize one or two metrics for next hop selection. For example, in VADD [2], a message is delivered along the currently available shortest-delay trajectory to the destination, in which the message delivery delay depends on the vehicle density on each road. However, most of these algorithms assume that the sender knows the current location of the receiver in advance. As a matter of fact, to a specific vehicle, precisely determining its current location is usually difficult due to its high-speed movement. Thus, these foregoing works are probably not suitable to tackle message dissemination among moving vehicles. Moreover, flooding and opportunistic transmission are two possible methods to deliver messages among moving vehicles, but these two methods are not so efficient since flooding costs much traffic overhead and in opportunistic transmission, message delivery delay might be very long.
As promising augmentation to intervehicle communication, roadside units (RSUs) have drawn much research efforts [4, 5], for routing between vehicles using them. Since RSUs are infrastructure, it is much easier to send a message to a fixed RSU than to a moving object. In addition, the delay of sending the message through the fixed RSU-based network is much less than through the highly dynamic VANET. These approaches, however, have their own drawbacks. First, as the transmission range of RSUs is limited, they are hardly adaptive to rapid-changing traffic. Apparently, the message delivery delay depends on opportunistically encountered public RSUs on a user’s trip. The sparser the deployment of RSUs is, the worse the performance is. Second, the installation of RSUs is costly: Jupiter research [6] has shown that these costs can be as high as 5,000 US dollars per unit. Third, RSUs may not exist in certain urban area or may be damaged once in some incidental happened disasters (e.g., flood, power cut, and hurricane). For example, in the heavy hurricane Sandy that happened in New York, USA, since most of infrastructures stopped working due to being destroyed or loss of power, the infrastructure based network cannot offer service to users for a long time [7].

In this paper, we propose the idea of Parking Backbone, which does not need any RSUs but leverages a virtual overlay network formed by outside parked vehicles to delivery messages between moving vehicles. With wireless devices enabled in the vehicle, parked vehicles can easily communicate with any vehicles driving through them. Owing to the huge storage capacity of the vehicle, a large amount of available information can be stored inside the parked vehicle. Thanks to the extensive and stable outside parking in cities, once grouped into the overlay structure, data transmission can be easily achieved over the Parking Backbone. Our scheme consists of three parts. At first, to each road, parked vehicles both at roadside and off street are grouped into a road cluster as far as possible. An urban overlay network is established based on road clusters for data transmission. Then, to a specific vehicle, a daily mobility model is established, to determine its location through a corresponding location prediction algorithm. Finally, a novel message delivery scheme is designed to efficiently transmit messages to destination vehicles through the proposed virtual overlay network. We investigate our scheme through realistic survey and simulation. The results prove that our scheme achieves high performance in message dissemination.

The major contributions of this work are as follows.

(i) To the best of our knowledge, we are the first to consider the using of Parking Backbone network on top of the VANETs in message dissemination. Our scheme aims at fully exploiting the benefits of outside parking, without requiring any extra infrastructure investment.

(ii) To a specific vehicle, we propose a location prediction algorithm to predict the vehicle’s location in a precise manner. Since messages can then be routed along an optimal routing to the destination vehicle, compared with flooding and opportunistic transmission, message transmission overhead in our protocol is largely reduced.

(iii) We evaluate the proposed scheme through realistic surveys and extensive simulations. The numerical results obtained verify that our scheme achieves effective routing performance.

The rest of paper is structured as follows. Section 2 presents a brief overview of related works. Section 3 describes the assumption and the framework of Parking Backbone. In Section 4, we explain the design of Parking Backbone. Section 5 focuses on vehicle location prediction, while Section 6 discusses message dissemination through Parking Backbone. Section 7 evaluates Parking Backbone through realistic survey and simulation, and Section 8 summarizes the paper.

2. Related Works

In recent years, many routing schemes have been proposed for VANETs. For example, in RBVT [8], a source vehicle that needs to transmit a message to a destination first broadcasts a route request message (RR) to discover a connected path to the destination. After RR reaches the destination vehicle, it sends a route reply that contains the connected path to the source vehicle. Since RBVT sends messages only when a connected path is available, the message delivery ratio might be very low. Skordylis and Trigoni [9] developed a D-Greedy algorithm for traffic routing. In D-Greedy algorithm, a vehicle makes decision on whether to forward a message or to keep carrying it based on the message’s TTL and its distance to the RSU.

Alternatively, approaches based on RSUs to disseminate data between mobile vehicles have also been considered. For example, in [4], Mershad et al. operate by using vehicles to carry and forward messages from a source vehicle to a nearby RSU and, if needed, route these messages through the RSU network and finally send them from an RSU to the destination vehicle. In addition, overlay networks on top of VANETs have been discussed in previous works. In [10], Hsieh and Wang propose an overlay multicast for VANETs (OMV), in which interested vehicles join a multicast tree initiated by a source vehicle. The drawback of the method is that the connectivity between two group member nodes depends on the intermediate nonmember nodes, which is a major cause of long delay in data transmission.

Finally, the authors in [11] believe that the huge amount of parked vehicles is an abundant and underutilized computational resource that can be tapped into for the purpose of providing third-party or community services. Since vehicles are parked above 95% of a day average [11], permitting their communication in the parking state will obviously improve the connectivity of VANETs. This means that more and more attention will be paid to the parked vehicles, which inspires our motivation to utilize Parking Backbone to perform intelligent message routing in VANETs.
3. Proposed Framework

3.1. Assumptions. First, we assume that the onboard battery has no strict size and weight limits, so that the power consumption is not a problem in daily parking. Second, we assume that each vehicle is equipped with a positioning system (e.g., GPS) and an electric map, which are already available to most of cars nowadays and will be common in the future. Third, we assume that some vehicle users will share their devices and data during parking. This could be achieved by incentive mechanism. According to the previous work of VANETs in [12], the businessmen may be willing to provide all sorts of incentives to make it attractive for the owner of parked vehicles to share the resources in their parked vehicles. Finally, we assume that the capacity of the vehicle is large enough for data storage.

3.2. Framework View. According to a real world urban parking report, street parking, off-street parking, and interior parking (garages or underground parking) account for 69.2%, 27.1%, and 3.7% of total, respectively. In this study, we mainly focus on the outside parked vehicles, including street parked vehicles and off-street parked vehicles. We aim at constructing a stable Parking Backbone on top of the highly dynamic VANETs. Then we provide effective message routing between moving vehicles and to track vehicles for this purpose over the proposed virtual infrastructure. Since parked vehicles are natural roadside nodes characterized by large number, long time staying, and wide distribution, they can serve as stable service infrastructure for the underlying dynamic VANETs. Our layered framework architecture is shown in Figure 1. It consists of three layers named physical layer, logical layer, and application layer, respectively.

(1) Physical Layer. The physical layer is the vehicular ad hoc network, in which moving vehicles or parked vehicles communicate with each other through one-hop or multihop transmission.

(2) Logical Layer. The logical layer is our Parking Backbone formed by outside parked vehicles that work as both data storage units and data transmission units in the virtual infrastructure. This virtual overlay network self-organized in a distributed manner for the support of target vehicle tracking and message dissemination on top of the VANET.

(3) Application Layer. Application layer focuses on safety and comfort applications for vehicle users. In this layer, the driver can send data to a target vehicle or issue certain request from it on the driver’s demand.

As shown in Figure 1, since the source vehicles depend on the proposed overlay network for destination vehicles tracking and messages transmission, how to effectively establish and maintain the overlay network is an important issue that needs to be carefully addressed. Accordingly, we suggest a novel overlay network design approach in this paper, as Parking Backbone over VANETs.

4. Design of Parking Backbone

For the idea of Parking Backbone, we first define the road cluster and then focus on the overlay network design based on road clusters.

4.1. Road Cluster. In Parking Backbone, we try to group all parked vehicles both on one road and off the road into a road cluster. It is viable, even though some parked vehicles are isolated from each other or belong to different partially distributed groups. The reason is that, for the on-street vehicles, the width of a parking space is about 5 meters. Since the transmission range of the vehicle is about 250 meters.
and the off-peak occupancy ratio averages almost 80%, for a parked vehicle in a parking lot, the probability of finding another parked vehicle is very high. For the off-street parked vehicles, some vehicles will drive in and out of the parking lots where they parked in. These moving vehicles bridge the disconnection in the store-carry-forward way and help to maintain the whole road cluster.

A typical road cluster is as shown in Figure 2, which has two cluster heads $H_1$ and $H_2$ and some cluster members as $M_1$–$M_{10}$. The creation process of a road cluster is described as follows. At first, the vehicles located at the two ends of a road are elected to be cluster heads, so that a moving vehicle entering or leaving the road will encounter one of them. In a two-way road, the two cluster heads, respectively, provide services for the vehicles coming from the nearest intersection. After the cluster head is determined, other parked vehicles on that road or off the road join the cluster and become the cluster members. The cluster members periodically report their ID number, position, routing request, and other useful information to the two cluster heads. Thus, the cluster heads are able to manage all cluster members and their resources. We notice that the road cluster will malfunction once the cluster head became invalid. For example, in Figure 2, the cluster head $H_1$ which is isolated at the road end may have no chance to inform others of its leaving. Thus, we introduce two quasi heads, as $QH_1$ and $QH_2$ in Figure 2, to enhance the robustness of the road cluster. A quasi head is the cluster member which is on the same road with a cluster head and next to it. The quasi head always keeps a copy of recent cluster status from the cluster head.

4.2. Overlay Network Construction. Instead of deploying RSUs, we try to construct a Parking Backbone overlay network on top of the physical VANETs, to achieve target vehicle tracking and efficient data delivery. Concretely, the virtual overlay network construction process has the following steps.

1. Each road cluster starts a neighbor discovery process using a simple mechanism with the help of intermediate vehicles. For instance, a cluster head of road cluster $J$ periodically broadcasts a $Nei.REQ$ message that contains its location and cluster number to the road clusters within two or three hops (the TTL time is set as 2 or 3 accordingly). When a road cluster $I$ receives a $Nei.REQ$ message from road cluster $J$, it records $J$ as its neighbor cluster. After the neighbor discovery process, each road cluster has the location of other road clusters within a certain range. We also notice that there exist singular road clusters which have no neighbor found during neighbor discovery process. The main reason can be explained from two aspects. The first reason is that the singular road cluster may be located in a relatively remote area; no outside parking vehicles are near it. The other reason stems from the fact that roadside parking is prohibited at some main city roads. If a road cluster is surrounded by roads where roadside parking is prohibited, the distance between it and other road clusters could be very long, and no neighbor road cluster will be found in the $Nei.REQ$ message broadcasting process.

2. To organize road clusters into a Parking Backbone, we let each road cluster connect with all its neighbor clusters. Thus, each road cluster becomes a member node and each connection between two neighbor cluster members becomes a virtual link of the overlay network. Moreover, each member node maintains the topology map of the virtual network. This could be realized with the help of intermediate vehicles; for example, each member node broadcasts its location to other member nodes over the network. This process is similar to link-state broadcast [13]. Figure 3 illustrates the concept of our overlay network.

As a survey [14] that explores on-street parking in cities points out, the utilization of the parking spaces is quite stable. The occupancy ratio, which is defined as occupied space-hour/available space-hour, averages 93% throughout one day. Even off-peak occupancy ratio averages almost 80%. In addition, the authors in [15] spatially describe the detailed
parking distribution in Montreal city in Canada at 12:00 and 22:00, respectively, which demonstrates the substantive vehicle sources in all hours of one day. Consequently, we conclude that, guaranteed by the high utilization and wide coverage of outside parking, establishing a stable Parking Backbone is feasible.

The proposed overlay network has the following distinct advantages. First, the virtual topology can remain static even though the underlying physical topology is changing rapidly. Thus, we can shield the complicated implementation details of the underlying physical network and tackle the message transmission issue on top of the VANETs. Second, as member nodes of the overlay network, each road cluster can be assigned a static network address like a static IP address on the Internet. A routing protocol similar to the link-state routing can then be designed for efficient data delivery over the overlay network. The other advantage is message bundling. That is, messages with the same destination can be bundled into one message, with the aim of reducing processing overhead and bandwidth consumption.

A real world urban parking utilization report [16] gives a parking study conducted in old town Alexandria, from which we obtain the density of roadside parking reaches one roadside parked vehicle per 7.58 meter road lengths. Similarly, [17,18] provide the roadside parking situation and the road construction in Singapore, respectively, from which we acquire that the density of roadside parked vehicles in Singapore reaches one roadside parked vehicle every 52.63 meters in average, as shown in Table 1. Due to the high density of parked vehicle in the two cities, we deduce that most of roadside parked vehicles can mutually connect in the city. Thus, our Parking Backbone achieves good network connectivity.

### 5. Vehicle Location Prediction

In this section, we first present the idea of vehicle tracking and locating. Second, how to establish vehicular mobility models to achieve vehicle tracking is described. Third, we focus on the method to track a target vehicle. Finally, the algorithm to locate the target vehicle based on its mobility model is discussed.

#### 5.1. Overview of Vehicle Tracking and Locating

To transmit the message to a mobile target object, some previous works try to use flooding or large-scale multicopy delivery. However, due to their large traffic overhead and high energy consumption in message transmission, these algorithms are impractical to be applied in large networks. So, if we can...
needs to send a message are concisely described.

steps of a specific vehicle tracking and location acquisition by giving a simple example. In Figure 4, a moving vehicle is extremely needed. Based network. Thus, a new method to track and locate a vehicle area is a tough work, especially in no fixed infrastructure. Track the location of the target vehicle, the message can then be transmitted more efficiently. However, due to the highly moving speed, tracking and precisely locating a moving vehicle are a tough work, especially in no fixed infrastructure based network. Thus, a new method to track and locate a vehicle is extremely needed.

Here we explain how to track and locate a specific vehicle by giving a simple example. In Figure 4, a moving vehicle $S$ needs to send a message $P$ to a target vehicle $D$, in which steps of a specific vehicle tracking and location acquisition are concisely described.

(1) Message transfer: when a driver issues a message $P$, the source vehicle transmits $P$ to the road cluster it is about to encounter (named $R1$) and records the encounter time as $t$ simultaneously.

(2) Query: if no information about vehicle $D$ is found in the head of $R1$, $R1$ then distributes a query message containing the encounter time and the ID of $R1$ over the Parking Backbone, for obtaining the mobility model of the destination vehicle.

(3) Echo: Parking Backbone members are checked on the destination vehicle’s mobility model during the dissemination of the query message, with which some road clusters that contain the mobility model of $D$ are found. Each of these road clusters then sends an echo message back to $R1$.

(4) Location prediction: the road cluster $D$ which is about to pass by (named road cluster $R2$) is predicted according to both its mobility model and time $t$.

(5) Transmit again: message $P$ is sent to $R2$ over Parking Backbone.

(6) Receive: $P$ is transmitted to the destination vehicle $D$ as soon as $D$ encounters $R2$.

There are several challenges in the implementation of this idea. First, since our target vehicle tracking and locating lie in explicit vehicular mobility model, how to describe and model vehicle’s daily moving is the key content. Second, efficient vehicle location tracking is necessary. Finally, precise location prediction should be considered.

5.2. Establishing Mobility Model. Table 2 lists some vehicles tightly related to our life. Buses support public transportation on some predefined routines. Taxies provide business services for passengers. Ambulances and other special vehicles respond to emergency calling, while personal vehicles are controlled by individual drivers and follow their respective social activities. Since different vehicles move in different ways, in the following, we introduce how to build mobility models for main types of vehicles.

5.2.1. For Personal Vehicle. Many researches validate that personal vehicles have regular mobility. Two to four main locations cover more than 70% of the overall trips of a personal vehicle. Some previous mobility prediction schemes try to establish the mobility pattern of a personal vehicle by recording its node-specific trajectories from GPS data. Since a vehicle will encounter a road cluster when entering a road where parking is permitted, we inspire to describe the movement of the personal vehicle by recording the road clusters it passes by. For example, a series of trips taken by the driver can be kept as records in Table 3, where “No.” represents the ID of the road cluster that the driver encountered in history, while the meeting time with each road cluster is separated into day (Mon., Tues., Wed., Thurs., Fri., Sat., Sun.) and the concrete time on that day. During movement, whenever a driver meets a road cluster, the onboard device will log the encounter time and record the ID of the road cluster.

We also find from Table 3 that, unlike usual, the driver took a different trip on Tuesday. This is possible for that he might go to other places such as hospital before going to workplace on that day. Moreover, in order to shorten data size, the onboard device merges the same trip trajectories by letting both the mean time value and the variance indicate the encounter time fluctuation of these trajectories. A simplified mobility model of Table 3 is shown in Table 4. With the mobility model, home road cluster and destination road cluster for each vehicle are defined in Table 3.

<table>
<thead>
<tr>
<th>Table 2: Vehicles and regular features.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle type</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>Bus</td>
</tr>
<tr>
<td>Personal vehicle</td>
</tr>
<tr>
<td>Ambulance, and so forth</td>
</tr>
<tr>
<td>Taxi</td>
</tr>
</tbody>
</table>

Figure 4: A typical destination vehicle tracking and location acquisition.
5.2.2. Home Road Cluster. The road cluster the driver starts with and goes back to after a whole day’s driving (like road cluster 2 in Table 4).

5.2.3. Destination Road Cluster. If the encounter time interval between a road cluster and its next road cluster is larger than a threshold $\mu$, this road cluster is a destination road cluster for this vehicle. Generally, the vehicle stays at its destination road cluster for a long time.

When a moving vehicle shows an up-to-date trip, the vehicle updates its previous mobility model and informs the new model to all its encountered road clusters timely. Consequently, road clusters always keep the latest mobility model of vehicles.

5.2.4. For Taxi. Huang et al. [19] prove that taxi mobility not only has regular microscopic behavior, but also shows macroscopic features. A taxi located in a certain area prefers to go to another area as the destination of a travel. Since it is very difficult to capture the microscopic regularity of the taxi, here we try to find the highly frequented locations of the taxi, in order to capture the taxi’s macroscopic movement. For a taxi $m$, suppose the probability of encounter road cluster $i$ while driving is expressed as

$$P_{mi} = \frac{f_{mi}}{N_m},$$

where $N_m$ is the total frequency of encounter road clusters while $f_{mi}$ denotes the times of meeting road cluster $i$ in taxi $m$’s moving history. Here we define a threshold $\eta$, if and only if $P_{mi}$ is not smaller than $\eta$; road cluster $i$ is one of the most frequently visited locations of this taxi.

5.2.5. For Bus. Generally, buses have predefined trips. The schedule of the bus, including the first bus, the last bus, and the frequency of the bus service, is available from predefined routines. Since electric map is widely deployed in buses nowadays, the road clusters that a bus will meet while moving can be obtained easily. Accordingly, a bus can estimate the meeting time with each road cluster on its trip according to its start time, its average moving speed, and the distance from the start point to each road cluster calculated using the electric map. However, since the movement of the bus is seriously affected by traffic conditions (e.g., traffic jam), using the bus schedule to predict bus movement is not so accurate. Thus, instead of utilizing bus schedule, we use the same method as we used with personal vehicle, to predict bus movement.

5.3. Mobility Model Diffusion and Location Tracking. When a moving vehicle meets a cluster head, it will report its mobility model. By this means, vehicle mobility models in the urban area are recorded after a time period. When a source vehicle $S$ issues a message $P$ to a vehicle $D$, since it has no knowledge of the location of $D$, it should firstly track the target vehicle. To track a specific vehicle, the cluster head of $R1$ does the following three operations step by step.

(1) It checks whether $D$ is now within the same road cluster with itself. If so, it initiates internal communication to distribute $P$ to $D$ directly.

(2) Otherwise, it checks whether it has the mobility model of $D$, if so, indicating that $D$ can be tracked immediately.

(3) If it does not have the mobility model of $D$, a query message that contains the encounter time between $S$ and $R1$ is broadcasted over the overlay network, for target vehicle tracking, as we described in Figure 5. Due to the small size of the query message (contains only the ID of destination vehicle and the ID of the road cluster which issues the query message), query message broadcasting brings a litter extra overhead to the whole network. In our simulation, it is shown that the total size of query message for all vehicles to broadcast once is almost 1 KB.

5.4. Location Prediction. After obtaining the mobility model of the destination vehicle, the location of the destination vehicle should be determined, for efficient message transmission.
Input:
\( (\alpha, \beta) \), the mobility model of destination vehicle \( D \) and a threshold \( \epsilon \) for location selection

Output: ID Numbers of the destination road clusters

1. match(\( \alpha \), day value in mobility model of \( D \))
2. for \( m = 1; m \leq k; m++ \) do
3. if \( (T_k + d_k) \geq \beta \geq (T_k - d_k) \) then
4. \( S \cdot add(l) \);
5. end if
6. if \( ((T_{m+1} - T_m) \geq \beta \geq (T_{m+1} - d_{m+1})) \) then
7. \( S \cdot add(m+1) \);
8. end if
9. else
10. \( S \cdot add(m) \);
11. end if
12. end if
13. if \( ((T_m - d_m) < \beta < (T_m + d_m)) \) then
14. if \( ((T_{m+1} - T_m) \geq \mu) \) then
15. \( S \cdot add(m) \);
16. else
17. \( S \cdot add(m+1) \);
18. end if
19. end if
20. end if
21. end for
22. return \( S \);

Algorithm 1: Destination road cluster prediction.

In the following, we propose a simple algorithm, as shown in Algorithm 1, to effectively predict the next location of the destination vehicle.

The main idea of the algorithm is tried to find the road cluster the destination vehicle is currently parked in or about to encounter by comparing the encounter time value provided in the query message with the history time records in mobility model of the destination vehicle. Note that \( d_1, d_2, \ldots, d_k \) in Algorithm 1 denote variance of each time record and \( T_1, T_2, \ldots, T_k \) represent mean time value of each time record on day \( \alpha \).

6. Message Dissemination

To disseminate messages to destination vehicles efficiently, we abstract the whole overlay network as a weighted connected graph \( G(V, E) \), where \( V \) is the set of road clusters, \( E \) is the set of virtual links between adjacent members, and weight \( D_{ij} \) on \( E \) represents the connectivity degree between adjacent member nodes \( i \) and \( j \). Since some adjacent road clusters are intermittent connected by moving vehicles, it is necessary to predict the connectivity degree between them. Figure 5 shows one such weighted connected graph.

In order to calculate \( D_{ij} \), we let adjacent road cluster heads periodically send a delay probe packet to each other. Since, for two neighbor nodes, the less the average message transmission delay between them is, the better they connect with each other, we let the reciprocal of the estimated transmission delay between two neighbors be their connectivity degree value. The process of estimating the connectivity degree can be described as follows:

\[
D_{ij,n} = \frac{t_n + T_{ij,(n-1)}}{D_{ij,n-1}T_{ij,(n-1)} + d_{ij,n}t_n},
\]

where parameters \( D_{ij,n}, T_{ij,(n-1)}, d_{ij,n}, \) and \( t_n \) represent the value of \( D_{ij} \) after receiving \( n \) delay probe replies, the sum of timestamps of the \( n \) delay probe replies, the \( n \)th delay probe reply, and the timestamp of the \( n \)th delay probe reply, respectively.

We explain our message dissemination from two modes. In the straightway mode, messages are routed along the maximum spanning tree to their destination road clusters. The maximum spanning tree could be easily acquired at each member node through the classic Kruskal’s algorithm. In the intersection mode, if two adjacent road clusters on the optimal path cannot connect with each other directly,
Table 5: Roadside parking in survey.

<table>
<thead>
<tr>
<th>Street</th>
<th>Policy</th>
<th>Density</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{04}, R_{15}, R_{26}$</td>
<td>No limits</td>
<td>280–320 veh/km</td>
<td>308 veh/km</td>
</tr>
<tr>
<td>$R_{37}, R_{79}$</td>
<td>Strict limits</td>
<td>15–25 veh/km</td>
<td>21 veh/km</td>
</tr>
<tr>
<td>$R_{45}, R_{46}, R_{47}$, $R_{48}, R_{49}, R_{98}$</td>
<td>Moderate limits</td>
<td>72–180 veh/km</td>
<td>95 veh/km</td>
</tr>
</tbody>
</table>

Figure 6: Select the next vehicle in the intersection.

Figure 7: Road topology in simulation.

7. Performance Evaluation

In this section, we investigate realistic parking and traffic profile in real urban environments. We also evaluate the performance of Parking Backbone in NS-2.33.

7.1. Survey. We performed a six-week survey on an ordinary urban area of Chengdu, a city in China, for collecting realistic parking and traffic profile. As shown in Figure 7, we extract a real street map with the range of 1600 m $\times$ 400 m, which contains both 10 intersections and 14 bidirectional roads totaled up to 7,860 meters and 8 off-street parking lots. Each intersection is marked by a number from 0 to 9.

During the survey, we investigated the traffic and roadside parking statistics at 16:00, 18:00, and 22:00 of every Tuesday, Thursday, and Saturday. We counted the vehicles parked along each street within 5 meters and skipped those parked in the middle of obstacles or too far from the roads. To 33 on-street parking lots, only fringed vehicles along road direction were calculated. As show in Table 5, there are three classes of streets with different parking limits. The first class permits free parking at roadside, as $R_{04}, R_{15},$ and $R_{26}$, which results in a very high node density. The second one, as $R_{37}$ and $R_{79}$, lacks public parking spaces. These streets have a very low vehicle density that comes from some reserved parking spaces and illegal parking. The remaining streets belong to the third one, where it has a moderate vehicle density. Generally, the parked vehicle numbers are stable in different hours of a day. During the survey, we also calculated daily traffic by counting the passing vehicles within fifteen minutes at random positions and found traffic fluctuating from 300 veh/h (vehicle per hour) to 2200 veh/h at different times of one day.
7.2. Simulation Environment. In this section, simulations are conducted in NS-2.33. We use the open source software VanetMobiSim-1.1 [20] to generate realistic urban mobility traces. The default number of vehicles is set to 200, and their radio range is set to 250 m. The average speed ranges from 40 to 80 kilometers per hour. The MAC protocol is 2 Mbps 802.11. In the simulation, parked vehicle nodes are located on random positions of each street or each off-street parking lot. For parked vehicles located on street, they follow the density collected in Table 5. The average parking time is 41.40 minutes with a standard deviation of 27.17, which is provided in [21]. Each vehicle generates a new request $R_i$ every 30 s. By default, the sender chooses another random vehicle as the destination. The threshold values $\mu$ and $\eta$ are set to 2 hours and 60%, respectively.

To compare our scheme with other VANET routing protocols, we simulated two protocols: CAN DELIVER [4] and RBVT [8]. We modified the design of RBVT to make each intermediate vehicle carry the message for a specific time (default of 10 s) if it finds that the next link is broken. To simulate the performance of CAN DELIVER, four RSUs were deployed uniformly across the map. The transmission range of RSU is set to 250 m. The metrics used for comparison are message delivery ratio, message delivery delay, and average vehicle traffic, which is the average traffic generated, received, and forwarded by a vehicle during the simulation time.

7.3. Results under Default Parameters. In Figure 8, we compare the message delivery ratio in Parking Backbone and the other two protocols under default parameters. We can see that Parking Backbone achieves higher message delivery ratio than CAN DELIVER and RBVT. This is mainly because, based on stable and large amount of roadside parking and off-street parking for message delivery, Parking Backbone has stable contact opportunities and high-quality contacts with parked vehicle. On the other hand, since the number of RSUs is limited in CAN DELIVER, the distance of the nearest RSU to a vehicle is long. Hence, the opportunity to send a message to the nearest RSU is small. RBVT has the lowest delivery ratio since it sends messages only when a connected path is available.

7.4. Varying the Number of Vehicles. We study the effect of varying the number of vehicles on the performance of the three schemes. Figure 9(a) shows that the message delivery ratio of our scheme is higher than those of the other two protocols. With the increase of the number of vehicles, the delivery ratio of CAN DELIVER and RBVT increases steadily. The reason is that more vehicles provide better possibilities and more contact opportunities to select a relay vehicle to delivery messages in dense traffic. However, since Parking Backbone mainly is based on stable roadside parking and on-street parking for message transmission, it has stable contact opportunities and high-quality contacts, and slightly affected by vehicular number change. In Figure 9(b), the message delivery delay of CAN DELIVER is less than RBVT in dense conditions, which proves that routing message via RSUs is faster in such cases. Figure 9(c) shows that the average vehicle traffic of CAN DELIVER and RBVT becomes high when the number of vehicles increases due to redundancy, which produces high traffic.

7.5. Varying the Request Rate. In this section, we discuss the performance of the three schemes when varying the request rate $R_r$ between 1 to 60 requests per minute. As Figure 9(d) describes, the message delivery ratios of the three schemes decrease as $R_r$ increases due to high congestion and message collisions. Figure 9(f) shows that the average vehicle traffic of CAN DELIVER is less affected than that of our scheme as $R_r$ increases since the latter uses flooding to find vehicle mobility model before a message is sent. On the other hand, as can be seen from Figure 9(e), since Parking Backbone tries its best to use stable parked vehicles to deliver messages, it speeds up the dissemination of message.

7.6. Varying the Number of RSUs. In this section, we uniformly deploy 1, 4, 10, and 20 RSUs in the simulation area and then compare the performance of CAN DELIVER with both our scheme and RBVT. The results are shown in Figures 10(a)–10(d). In Figure 10(a), the delivery ratio of our protocol is higher than CAN DELIVER with 4 RSUs but lower than the latter with 10 and 20 RSUs. This is reasonable since when more RSUs are deployed, a vehicle to its nearest RSU will be closer. Hence, the opportunity to send a message to the nearest RSU will increase. Figure 10(b) depicts that the message delivery ratio of CAN DELIVER increases as the number of RSUs increases. The reason is the same as described in Figure 10(a). We also notice that when the number of RSUs is large enough such that the whole network is well covered, the delivery ratio will significantly increase and the average message delivery delay will largely decrease (see Figure 10(c)).

8. Conclusion
This paper proposes Parking Backbone, which lets extensive parked vehicles support vehicle tracking and message dissemination as roadside units. In this paper, we first investigate the design of Parking Backbone. Then, we establish mobility model for different vehicles and propose algorithm to predict the rough location of the destination vehicle. Finally, effective
Figure 9: Impact of number of vehicles and request rate.
message delivery scheme is discussed. The evaluation of Parking Backbone confirms its effectiveness as compared to recent routing protocols for VANETs.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgment**

This work was supported by the National Natural Science Foundation of China (Grant nos. 61103227, 61103226, 61272526, 61170256, 61173172, 61173171, and 61370204).

**References**


