

## Research Article

# IPARK: Location-Aware-Based Intelligent Parking Guidance over Infrastructureless VANETs

Hui Zhao, Li Lu, Chao Song, and Yue Wu

*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Hui Zhao, jenniferzhao09@gmail.com

Received 16 October 2012; Accepted 13 November 2012

Academic Editor: Haigang Gong

Copyright © 2012 Hui Zhao 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.

It has been pointed out that about 30% of the traffic congestion is caused by vehicles cruising around their destination and looking for a place to park. Therefore, addressing the problems associated with parking in crowded urban areas is of great significance. One effective solution is providing guidance for the vehicles to be parked according to the occupancy status of each parking lot. However, the existing parking guidance schemes mainly rely on deploying sensors or RSUs in the parking lot, which would incur substantial capital overhead. To reduce the aforementioned cost, we propose IPARK, which taps into the unused resources (e.g., wireless device, rechargeable battery, and storage capability) offered by parked vehicles to perform parking guidance. In IPARK, the cluster formed by parked vehicles generates the parking lot map automatically, monitors the occupancy status of each parking space in real time, and provides assistance for vehicles searching for parking spaces. We propose an efficient architecture for IPARK and investigate the challenging issues in realizing parking guidance over this architecture. Finally, we investigate IPARK through realistic experiments and simulation. The numerical results obtained verify that our scheme achieves effective parking guidance in VANETs.

## 1. Introduction

Finding an available parking space in a crowded urban area or a large parking lot is always time-consuming, which would incur great economic losses, for example, searching for free parking spaces causes total of 20 million Euros economic damage per year in the district Schwabing of Munich, including 3.5 million Euros for gasoline and diesel and 150000 hours of waiting time [1]. Moreover, a recent study [2] discovers that about 30% of the traffic congestion is caused by vehicles cruising around their destination and looking for a place to park. Therefore, it is of great significance to address the problems associated with parking in crowded urban areas. Obviously, if drivers are provided with efficient parking guidance services, the parking spaces searching costs would be greatly reduced and the traffic congestion would be also relieved to some extent.

In the last decade, many research efforts have been devoted to designing efficient parking guidance systems [3–7]. These works could be generally classified into two categories: sensors based scheme and RSUs based scheme. In the

sensors based scheme [3–5], fixed sensors are deployed in the parking lot to monitor parking spaces, which provide real-time parking space availability information for the drivers looking for parking spaces. One of the biggest weaknesses of this approach is that it necessitates a large installation cost and operational cost in order to adequately monitor the parking spaces at a city-wide level. According to a Department of Transportation report [8], the installation cost of typical per spot parking management systems ranges from \$250–\$800 per spot. Moreover, the deployed sensors might become invalid over a period of time owing to the power exhaustion. In the RSUs based scheme [6, 7], it employs the parking lot RSUs to monitor the whole parking lot and perform parking guidance. This scheme is proved to be efficient and effective. However, deploying RSUs at a large scale also requires a large amount of investment and elaborate design.

In view of the drawbacks of the existing schemes, we put forward an intelligent parking guidance scheme IPARK, which harnesses the free resource offered by parked vehicles to perform parking guidance in urban areas. In IPARK,

the cluster formed by parked vehicles generates the parking lot map automatically, monitors the occupancy status of each parking space in real time and provides services for vehicles searching for parking spaces. Overall, IPARK could provide the drivers with two kinds of services: parking availability information dissemination outside the parking lot and real-time parking navigation inside the parking lot. Specifically, our IPARK scheme is substantiated with a distributed architecture. Over this architecture, the cluster formed by parked vehicles establishes the parking occupancy map according to the reported location of each parked vehicle, and provide real-time navigation for vehicles entering this parking lot. Moreover, different parking clusters in a specific area share their parking data periodically and jointly response to the parking data query request. We try to tackle the challenging issues in performing efficient parking guidance over the proposed architecture, for example, how to manage the parked vehicles to facilitate parking guidance and how to generate the internal map of a parking lot. Finally, we investigate our scheme through realistic experiments and simulation. The results prove that our scheme achieves high performance in parking guidance.

The original contributions that we have made in the paper are highlighted as follows.

- (i) To the best of our knowledge, we are the first to consider the using of parked vehicles in parking guidance. Our scheme aims at fully exploiting the benefits of parked vehicles, without requiring any infrastructure investment.
- (ii) We propose a distributed architecture for parking guidance over infrastructureless VANETs and tackle the challenging issues in perform parking guidance over this architecture.
- (iii) We evaluate the proposed scheme through analyzing realistic parking data and performing simulation in NS-2.33. The numerical results obtained verify that our scheme achieves effective parking guidance in VANETs.

The remainder of this paper is structured as follows. Section 2 makes a brief overview of related work. Section 3 describes the framework of IPARK. In Section 4, we explain the design of IPARK step by step. Section 5 evaluates IPARK through realistic experiments and simulation. Finally, Section 6 summarizes the paper.

## 2. Related Work

In recent years, many research efforts have been devoted to developing effective parking guidance schemes for urban areas.

Some researchers propose to exploit the wireless sensor networks to achieve parking guidance. In [3], an optical wireless sensor network is deployed in the parking lot, which monitors the parking spaces and informs drivers of the number of available parking spaces and in which area should they be directed to. Similar scenario has also been proposed in [4], which detects the presence of parked

vehicles over parking spots using fixed sensors and provides the parking statistics for vehicles searching for parking spaces. Furthermore, the city of San Francisco is presently installing a stationary sensor network to cover 6000 parking spaces under the SFPark project [5]. This network detects the presence of a vehicle using a magnetometer and delivers the data to a centralized parking monitoring system, which then carries out parking guidance based on the collected data. The deficiencies of the sensors based schemes is as follows. First, deploying sensors in a large parking lot can be very expensive. Second, the sensors can become inaccurate and would stop functioning easily when time passes.

Alternatively, approaches based on RSUs to accomplish efficient parking management have also been considered. SPARK [6] employs RSUs to surveil and manage the parking lot. It provides the drivers with real-time parking navigation service, intelligent antitheft protection, and friendly parking information dissemination. In [7], Panayappan et al. also provide a RSUs-based approach for parking space availability monitoring. The parking lots are managed by RSUs, and these RSUs can provide open parking space information to the drivers, which is very similar as the proposed SPARK scheme. However, deploying RSUs at a large scale also requires a large amount of investment and elaborate design.

In addition, Caliskan et al. [9] propose a topology independent scalable information dissemination algorithm for free parking places discovery. With the friendly parking lot information disseminated by the parking automats and inter-vehicle broadcast, the drivers can conveniently find their preferred free parking lot. However, this scheme relies on the roadside parking fees payment terminal, which has not gained popularization until now. ParkNet [10] collects parking space occupancy information by moving vehicles. Each ParkNet vehicle is equipped with a GPS receiver and a passenger-side-facing ultrasonic rangefinder to determine parking spot occupancy. The data is aggregated at a central server, which builds a real-time map of parking availability and could provide this information to clients that query the system in search of parking. The drawback of ParkNet is that periodically reporting the parking status to a central server over cellular uplink would incur high communication cost. In addition, it only focuses on the roadside parking, which has limited practicality.

Finally, in [11, 12], the authors believe that the huge amounts of vehicles on parking lots are an abundant and underutilized computational resource that can be tapped into for the purpose of providing third-party or community services. This means that more and more attention will be paid to the parked vehicles, which inspires our motivation to utilize parked vehicles to perform intelligent parking guidance.

## 3. System Model

*3.1. Assumptions and Design Goals.* First, we assume that vehicles are equipped with GPS and preloaded electric maps, which are already popular in new cars and will be common in the future. Second, we assume that some vehicle users will share their devices during parking. This could be motivated

by effective incentives. In view of the powerful resources possessed by parked vehicles, 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 [12]. Finally, we assume that vehicles parked within the same parking lot form one cluster, which is feasible owing to the high occupancy of urban parking lot.

According to a real world urban parking report [13], street parking, off-street parking and interior parking (garages or underground parking lots) account for 69.2%, 27.1%, and 3.7% of total, respectively. Here, we mainly focus on the outside parking lots, including roadside parking lots and off-street parking lots. Our IPARK aims at exploiting the cluster formed by vehicles parked within the same parking lot to surveil the whole parking lot and provide effective guidance for vehicles searching for parking spaces. Specifically, this intelligent parking scheme is designed to achieve the following two goals: real-time parking navigation inside the parking lot and efficient parking information dissemination outside the parking lot.

*3.1.1. Real-Time Parking Navigation.* While a vehicle enters a large parking lot, the parking lot cluster should send the current parking occupancy map (a map showing each parking spot as occupied or vacant) to it and help it find an available parking space.

*3.1.2. Efficient Parking Information Dissemination.* While a vehicle user is driving on a road and issues a parking data query request, a parking lot nearby should respond to this request and return the query results (the number of parking spaces available on each parking lot nearby) at a small delay.

*3.2. Architecture.* To achieve the aforementioned two design goals, the architecture of IPARK is elaborately designed, which is as shown in Figure 1. It consists of three components: parking lot clusters, intermediate nodes and end users.

Parking lot clusters are composed of vehicles parked within the same parking lot. They work both as parking monitoring units and data dissemination units, which gather the information necessary for parking guidance and provide services for the vehicles in search of parking spaces.

Intermediate nodes are the vehicles helping to connect different parking lot clusters. They could be either vehicles moving on the road or individual parked vehicles in a private parking lot.

End users are the vehicle users who are seeking for an unoccupied parking space while driving on a road or entering a large parking lot.

Over this architecture, each parking lot cluster monitors the corresponding parking lot and establishes a parking occupancy map in real time, based on which it provides navigation services for the incoming vehicles. Moreover, the parking lot clusters within a specified area share their real time parking space availability information periodically, which jointly respond to the parking data query request.

This proposed architecture has the following two distinct advantages. First, it fully exploits the benefits of parked vehicles, and does not require any infrastructure investment.

Thus, it is cost saving. Second, with the parking availability information distributed to many parking lot clusters, a driver seeking for an available parking lot could get a response within short time. In addition, with the parking space occupancy map in hand, a driver entering a large parking lot could find a vacant parking space swiftly. Thus, it is time saving. The above two merits make IPARK attractive to both parking managers and vehicle users. Thus, it could easily obtain supports from all participants.

## 4. The Design of IPARK

At a high level, the design goal of IPARK is to exploit the parking lot cluster to realize efficient parking guidance over VANETs. To achieve this goal, the following four issues need to be addressed.

- (1) How to manage the parked vehicles within a parking lot to facilitate parking guidance?
- (2) How to obtain the internal map of a parking lot?
- (3) How to achieve effective parking navigation inside the parking lot?
- (4) How to realize efficient parking availability information dissemination outside the parking lot?

In this section, we attempt to investigate feasible solutions for the above issues.

*4.1. Parking Lot Cluster.* In IPARK, we perform parking guidance on the basis of parking lot cluster. Vehicles parked within the same parking lot are considered to be one parking lot cluster. A typical parking lot cluster is as shown in Figure 2, which has one cluster head  $H$  and some cluster members as  $M_1-M_{23}$ . It is established as follows. At the beginning, there is not any parked vehicle. Then, among the earliest incoming vehicles, the one which provides PVA (Parked Vehicle Assistance) service becomes the cluster head. After the cluster head is determined, other parked vehicles join the cluster and become the cluster members. We notice that the parking lot cluster will malfunction at once if the cluster head became invalid (e.g., the wireless device is turned off suddenly or the cluster head is left). Thus, we introduce a quasi-head, as  $QH$  in Figure 2, to ensure fault-tolerance with respect to exception handling. A quasi-head is the cluster member next to a cluster head, which always keeps a copy of recent cluster status from the cluster head. It becomes a special cluster member, working as a “warm backup” for the management of the parking lot cluster. While the cluster head is invalid, the quasi-head becomes the new cluster head and a vehicle close to the cluster head is appointed as the new quasi-head. Once the cluster head is determined, it manages all cluster members, act as local service access points, and perform parking guidance. Whenever a parking lot becomes empty, the parking lot cluster needs to be re-established and the parking data needs to be gathered again. However, it has been proved that the possibility for a parking lot in urban areas being completely empty is very slim [14, 15]. Once a parking lot

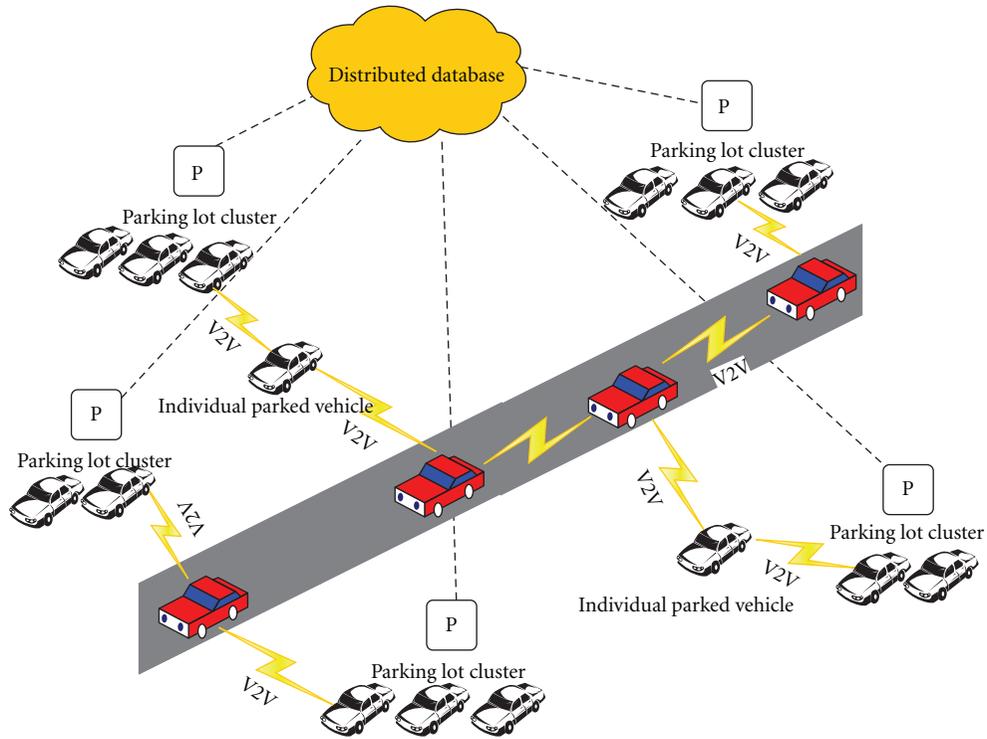


FIGURE 1: System architecture.

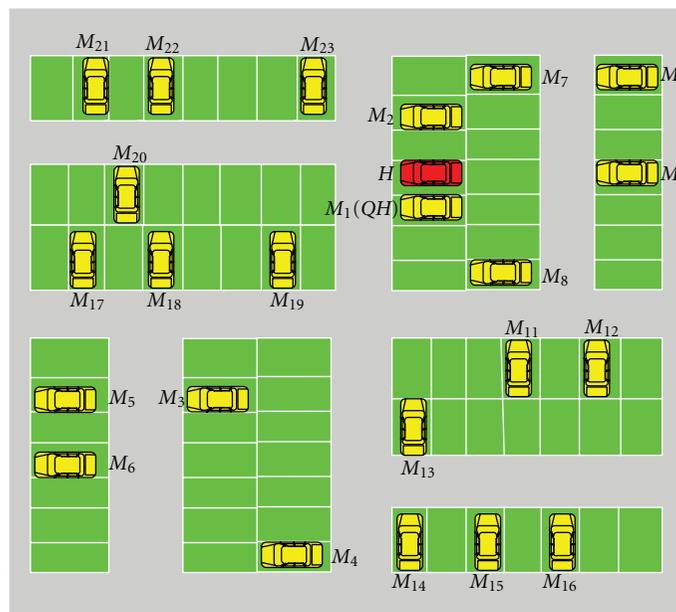


FIGURE 2: A typical parking lot cluster.

cluster is established, it could work steadily for a long period of time.

**4.2. Location-Aware-Based Map Generation.** To realize effective parking guidance, the parking lot cluster needs to obtain an internal map of the parking lot beforehand. This map should include the total number of parking spaces and their specific locations. The total number of parking spaces

indicates the capacity of the parking lot, while the specific location could be either a physical location or a logical location. As there is no way for a vehicle getting such a map until now, we offer an automatic parking lot map generation scheme.

To establish a parking lot map, one intuitive way is letting the parked vehicles report the GPS readings at the corresponding parking spots to the cluster head. However,

the existing GPS technology is not accurate enough to locate each parking space. Even the GPS reader corrected by WAAS has an error of 3 m (standard deviation), not to mention the ones in the area without WAAS. Thus, instead of utilizing the physical location, we record the logical location of each parked vehicle. A vehicle estimates its logical location according to the road topology of the parking lot and the recorded GPS trajectory. Before turning off the engine, it reports the estimated logical location to the cluster head, based on which the cluster head detects the existence of a parking space. We will explain how to get the accurate logical location of a vehicle.

As demonstrated in Figure 2, a typical parking lot is composed of many parking spaces and several roadways. After entering a parking lot, the vehicle users have to drive along the roadways and park in one of the vacant spaces. While analyzing the GPS trajectories of different vehicles, we observe the following two characteristics. First, when a vehicle is turning from one roadway to another, the trajectory direction displays a significant difference, as shown in Figure 3. Second, the GPS error is highly correlated for a long driving distance, which is reflected by the fact that the vehicle trajectory is nearly paralleled with the real roads in Figure 3. It is worth to note that, we are not the first ones to make such observations, similar characteristics have already been discovered and utilized by many works [10, 16]. We assume that the turnings in the GPS trajectories could be matched to the road bends in the parking lot using the existing map matching techniques [16, 17]. Then, according to the first observation, the turning of a roadway and the related GPS reading could be easily recognized from the vehicle trajectory. Thus, we could take the roadway turnings as the reference points of the parking lot, based on which the relative locations of the parked vehicles are estimated. According to the second observation, while a vehicle turns from one roadway to another and drives on the new roadway, the GPS error could be deemed as identical within a certain driving distance. That is to say, if we calculate a distance on the basis of the GPS readings at the turning and a road point within a certain distance, it could be used to represent the relative distance from that road point to the turning (referenced point). By this means, the logical location of the parked vehicle could be obtained easily. The logical location estimation algorithm is depicted in Algorithm 1.

The parking lot cluster generates the parking lot map through the aggregation of reported logical location of different parked vehicles. Theoretically, once each parking space is occupied one time, the parking lot map could be generated. However, the logical location reported by the parked vehicle might not accurate occasionally owing to some unpredicted errors. Therefore, we let the parking lot cluster remove the ones with obvious derivation and correct the parking lot map through long periods of time. An example of a parking lot map is as shown in Figure 4. Each parking space is represented by  $\langle \text{Referencepoint}, \text{Logicalnumber} \rangle$ , which indicates the logical location relative to the chosen reference point. Although this location representation scheme is very simple, it could uniquely identify a parking space. After each parking space is occupied over a period of time, the capacity of the



FIGURE 3: GPS trajectory sample.

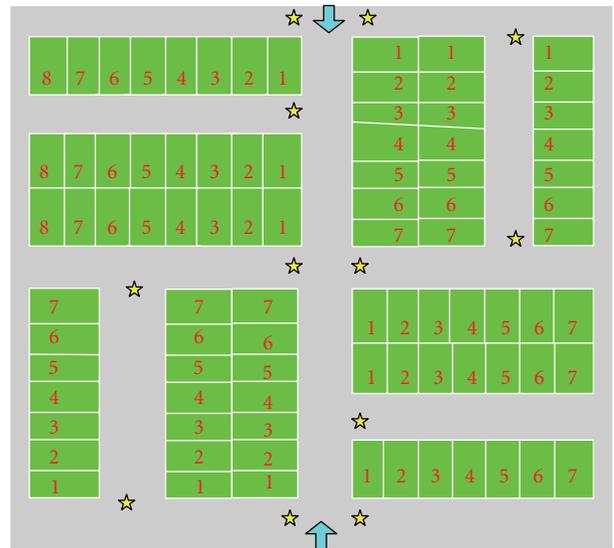


FIGURE 4: An example of logical parking lot map.

parking lot could also be estimated. Assume that the parking lot is divided into  $n$  blocks by the roadway, then, the capacity of the parking lot could be calculated as follows:

$$C = \sum_{i=1}^n C_i. \quad (1)$$

Additionally, as the GPS error is only correlated within a certain driving distance, the threshold within which the GPS error correlation could be utilized to compute the logical location of a parked vehicle is of great significance. In our simulation, we investigate this threshold through large number of experiments under urban scenario. The results show that within a driving distance of 100 m, 90% of the error between the estimated distance and the real distance is less than 1.5 m, which is small enough to uniquely determine a parking space. Thus, we take 100 m as the threshold. Out of this threshold, the reference point is redefined. While a vehicle drives as long as 100 m along a road segment, the average GPS reader at that point is selected as a new reference point, based on which the relative distance is estimated.

```

Require: GPS trajectory data  $GPS\_DATA$  of vehicle  $v_j$ ,
road topology  $PARK\_ROADS$  of the parking lot, a
threshold  $\Phi$ 
Ensure: The logical location of the parked vehicle  $v_j$ , represented
by  $(R_j, d_j)$ .
1: Preprocess the GPS trajectory data and remove the GPS
readers with large derivation as
 $GPS\_DATA = preprocess(GPS\_DATA)$ ;
2: for  $i = 1; i \leq size(GPS\_DATA) - 2; i +$  do
3:  $A = GPS\_DATA[i], B = GPS\_DATA[i + 1], C =$ 
 $GPS\_DATA[i + 2]$ ;
4: Find the angle between  $\overrightarrow{AB}$  and  $\overrightarrow{BC}$  as
 $\theta = angle(\overrightarrow{AB}, \overrightarrow{BC})$ ;
5: if  $\theta \geq \Phi$  then
6: Put  $B$  into the turning point linked list as
 $S.add(B)$ ;
7: end if
8: end for
9: Derive the trajectory path  $PATH$  of vehicle  $v_j$  as an
ordered set
 $PATH = (S[n - 1]S[n] \mid 1 \leq n \leq size(S))$ 
10: Match the derived path  $PATH$  with the road segments
within the parking lot and find the last road segment the
vehicle drive on as
 $R_j = match(PATH, PARK\_ROADS)$ ;
11: Compute the distance between the turning point of  $R_j$ 
and the road point corresponding to the parking space of
vehicle  $v_j$  as
 $d_j = distance(turningPoint(R_j), roadPoint(v_j))$ ;
12: return  $(R_j, d_j)$ 

```

ALGORITHM 1: Logical location estimation within a parking lot.

**4.3. Parking Navigation.** In SPARK [6], the parking navigation is realized through recommending an available parking space to the incoming vehicle and then guiding it to that space step by step. However, this scheme is impractical for the following two reasons. First, it neglected the demand of vehicle users. For the vehicle user, compared with being recommended with a vacant parking spot, they prefer to get the occupancy status of the parking lot, and then choose one available parking space according to personal preference and driving proficiency. Second, considering the fact that the vehicle driver might not adopt the recommended parking space, it is unreasonable to guide the vehicle users to a recommended parking space step by step.

In our scheme, we let the parking lot cluster establish a parking occupancy map and send this map to the incoming vehicles. To achieve this goal, a vehicle is required to report the estimated logical location to the cluster head before turning off the engine. For the cluster head, it detects the occupancy status of each parking space and maintains a record containing three fields as follows for it:

PID: the ID of this parking space.

OCC: the occupancy status of this parking space. 0 represents unoccupied, and 1 represents occupied.

VID: if this parking space is occupied, VID represents the ID of the parked vehicle.

With the internal map of the parking lot and the occupancy status of each parking space, it is not difficult to derive the corresponding parking occupancy map. One part of a parking occupancy map is as shown in Figure 5.

While a vehicle entering a large parking lot, it would periodically broadcast a beacon message. After receiving this beacon message, the cluster head sends the parking occupancy map to this vehicle, and the vehicle user chooses to stop in one of the vacant parking spot.

In order to reduce the transmission overhead, the cluster head could determine whether to provide parking navigation service according to the current occupancy ratio of the parking lot. While the parking occupancy is less than a certain threshold, it is easy for a vehicle user to find a vacant parking spot, and there is no need for the cluster head to send a parking occupancy map to it. While the parking occupancy is higher than a certain threshold, the cluster head begins to perform parking navigation.

**4.4. Parking Availability Data Dissemination.** To help a vehicle moving on a road find an available parking space conveniently, we organize the parking lot clusters within a specified area into a distributed parking database, and provide near real-time parking data query services for the vehicle users. According to the strategy used, this process could be divided into two phases: parking data sharing among different

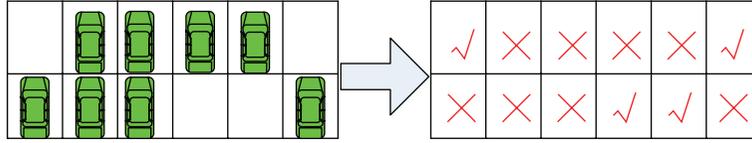


FIGURE 5: An example of parking lot occupancy map.

parking lot clusters and parking data dissemination. We will describe them in detail in the following part.

**4.4.1. Parking Data Sharing among Different Parking Lot Clusters.** We assume that one parking lot cluster knows the location of other parking lot clusters within a certain range. This could be realized through a simple mechanism with the help of intermediate vehicles, for example, each parking lot cluster periodically broadcasts its location (the location of cluster head) to the parking clusters within two hop (the TTL is set as 2), and adjacent parking lot clusters exchange the information (similar like  $\langle clusterID, location \rangle$ ) they obtain with each other. This process is similar to Link-State Broadcast [18]. Due to the high occupancy of urban parking lots, a long broadcast cycle is enough.

Under this assumption, we abstract the parking lot clusters within a specified area as a weighted connected graph  $G(V, E)$ , where  $V$  is the set of parking lot clusters,  $E$  is the set of communication links between adjacent parking lot clusters (which might be intermittently connected) and weight  $D_{ij}$  on  $E$  is the transmission delay of the link between adjacent parking lot clusters. Figure 6 shows one such weighted connected graph. We let adjacent parking lot clusters periodically send a delay probe packet to each other and estimate the transmission delay according to the history record. As the transmission delay between two parking lot clusters is affected by their mutual distance, the traffic density, and other factors that changes slowly, this approximation is reasonable.

After this weighted connected graph  $G$  is acquired, each parking lot cluster periodically broadcast the parking availability information to other parking lot clusters over the minimum spanning tree MST of graph  $G$ . The minimum spanning tree MST could be easily acquired at each parking cluster through the classic Kruskal's algorithm or Prim's algorithm, both of which are of polynomial complexity:

$$MST = \arg \min_{TCG} \sum_{i,j \in T} D_{ij}. \quad (2)$$

With this parking data sharing scheme, each packet only needs to be replicated while new tree branch appears, which greatly decreases the transmission overhead. Moreover, routing along the minimum spanning tree could guarantee that total estimated data sharing delay is minimized. Although routing along any one spanning tree could make sure that the packet could be received by each parking cluster, routing along the minimum spanning tree could realize the same goal in a shorter time.

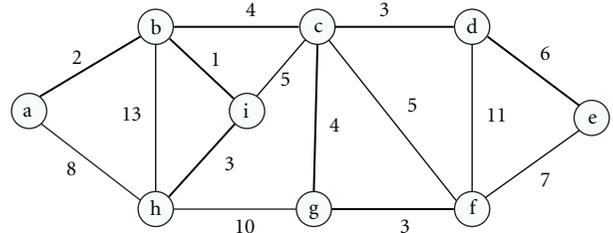


FIGURE 6: A minimum spanning tree.

**4.4.2. Parking Data Dissemination.** After the distributed parking database is established, it is responsible to deal with the parking data query request and disseminate the parking availability data to the end user.

In regards to the end user, there is no determined destination for query dissemination. However, some query request may occur between vehicles and parking lot cluster within mutual radio range. Since the vehicles can monitor nearby beacons, query messages can be directly sent to the parking lot cluster and the query results are returned immediately. In regards to the vehicles outside the radio range, we need a proper strategy to spread queries over VANETs. Epidemic routing [19] ensures eventual message delivery via random pair-wise exchanges of messages between mobile hosts, which has decent performance, but often causes heavy resource consumption in the network. GPSR [20] provides a location-based scheme that all packets transmitted into the network are marked by the originator with their destination's location. In greedy forwarding, intermediate nodes simply choose to forward the packet to the neighbor located closest to the destination, which is efficient and robust in straight roads, but often causes extra transmission and delay in complex topologies. In order to respond to a parking space query quickly and efficiently, we develop an enhanced GPSR to disseminate a query in a neighboring area around a vehicle.

When a driver issues a parking space query request, it is disseminated as follows. In the straight mode (as shown in Figure 7(a)), this query message is forwarded to the furthest neighbor along the driving direction. If the intermediate forwarder encounters a parking lot cluster, it sends the query message to the parking lot cluster, and the parking lot cluster returns the query results along a reversed path as the query message using GPSR. If no parking lot cluster encounters, the intermediate forwarder sends the query request to the next intermediate node until a parking lot cluster is encountered. In the intersection mode (as shown in Figure 7(b)), the

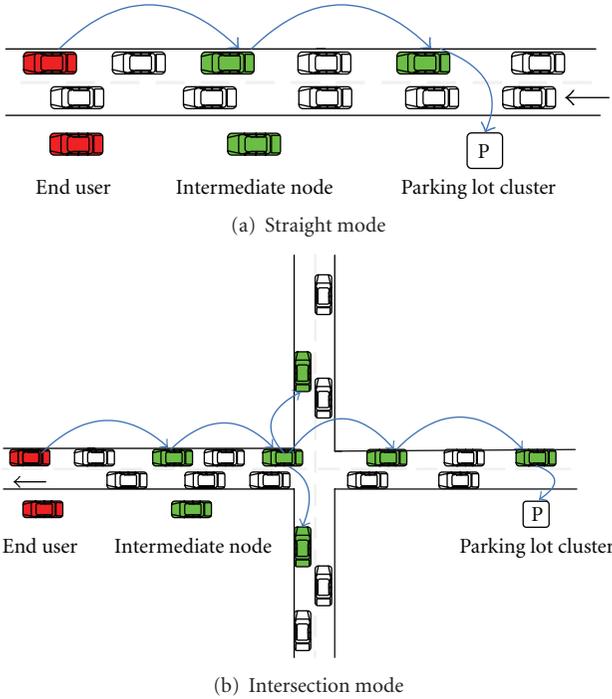


FIGURE 7: Query dissemination in IPARK.

query message is replicated so that there is one copy at each possible driving direction. While receiving a query request, the parking lot cluster returns the query results along a reversed path as the query message.

By this means, we could assure that the vehicle user could get a reply with small delay regardless of which path it would take at the crossing. Considering the fact that some other vehicles driving within the same area might be also seeking for parking spaces, we further improve our scheme as follows: while the query message is delivered using Improved GPSR, if the intermediate forwarder encounters a vehicle which has already received the response message recently, the results will be returned and the response message will be dropped.

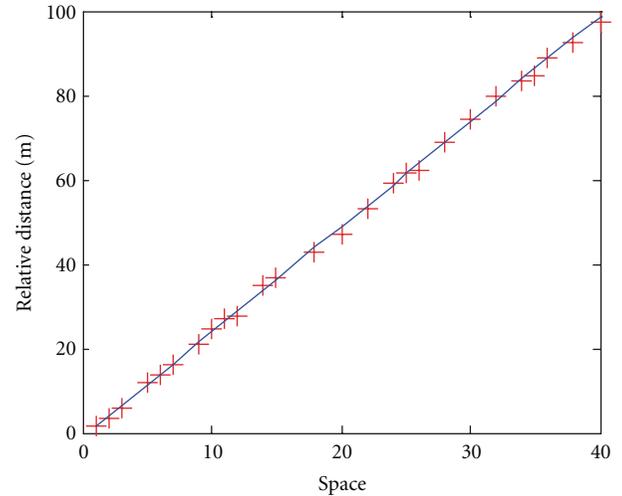
## 5. Performance Evaluation

In order to evaluate IPARK accurately, we collect realistic parking data in a large parking lot and perform a series of numerical experiments on these data. Furthermore, we also examine the performance of IPARK in NS-2.33.

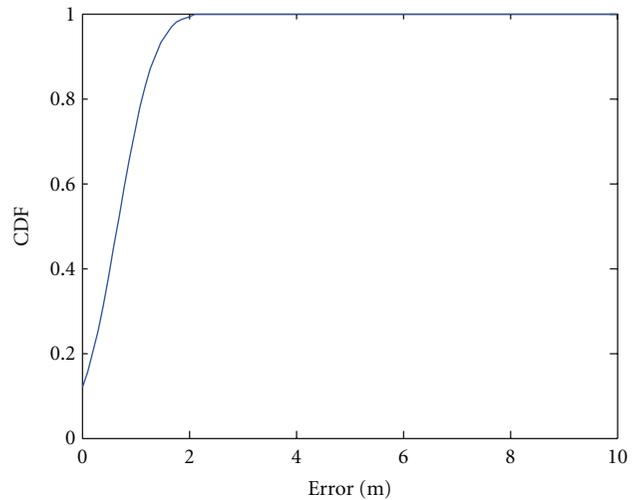
*5.1. Numerical Experiments and Results.* We collect realistic parking data at a large parking lot of Chengdu, a city in China. As shown in Figure 8, this parking lot has about 2000 parking spots. Due to the fact that this parking lot has a high occupancy most of the time, we only choose a relatively vacant subarea, as marked in Figure 8, for study. Our private car is quipped with a Garmin 18–5 Hz GPS, which has 12 channel receiver and provides 5 fresh GPS readings per second. During the experiment, we repeatedly drive into the parking lot. To simulate the real parking behavior of



FIGURE 8: Parking lot under study.



(a) Data distribution



(b) CDF of the estimation error

FIGURE 9: Error distribution of the samples collected in the parking lot.

different vehicles, we park at a different parking space in the chosen subarea each time. Specially, the corresponding GPS trajectory is recorded and the relative logical location (logical number relative to the reference point) of the parked space is recorded manually. Apart from this, the data collection process was not controlled in any other manner (e.g., speed,

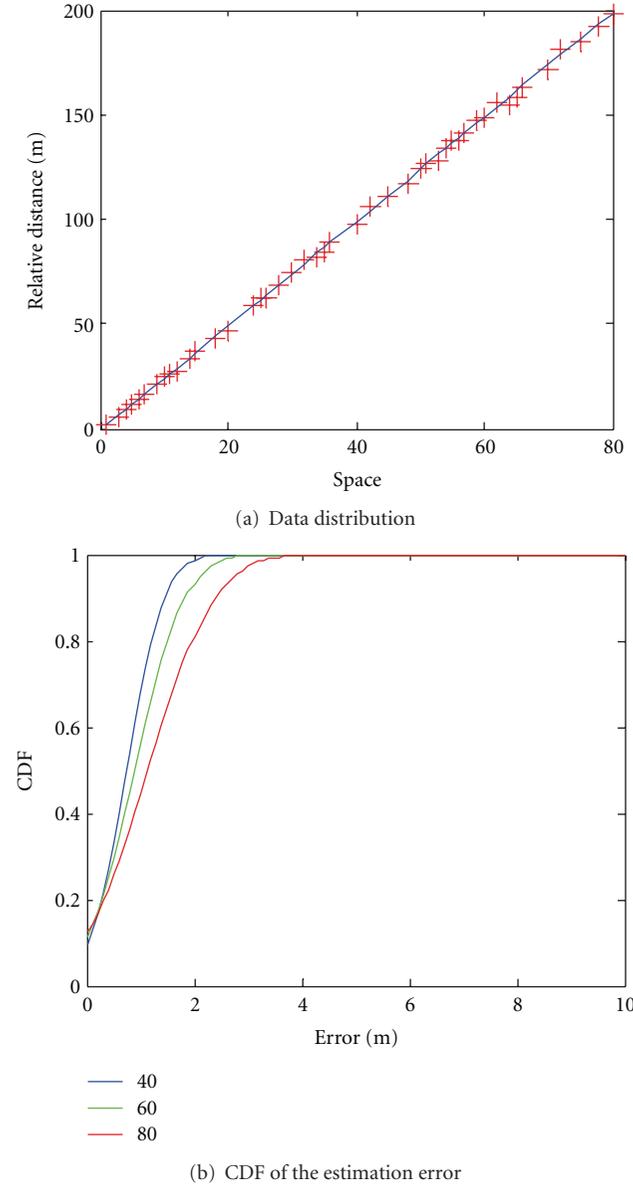


FIGURE 10: Error distribution of the samples collected in the virtual parking lot.

obstacles, etc.). After the survey, the data provided by GPS is accessed via an USB serial port and processed on a computer.

To verify the effectiveness of the proposed logical location estimation algorithm, we analyze the collected data in the following way.

Let  $V = [v_1; v_2; \dots; v_N]$  denote the set of  $N$  locations on a road corresponding to the mid-point of  $N$  different parking spaces along this road segment, and  $X_V = [X_1; X_2; \dots; X_N]$  the corresponding random variables describing the estimated distance relative to the reference point at these locations. Assume  $\text{err}(i)$  is a certain error function measuring the distance between the real value and the estimate at the  $i$ th location, which is denoted by  $\text{err}(i) = X_i - \bar{X}_i$ . We assume  $\text{err}(i)$  to be Gaussian and perform analysis.

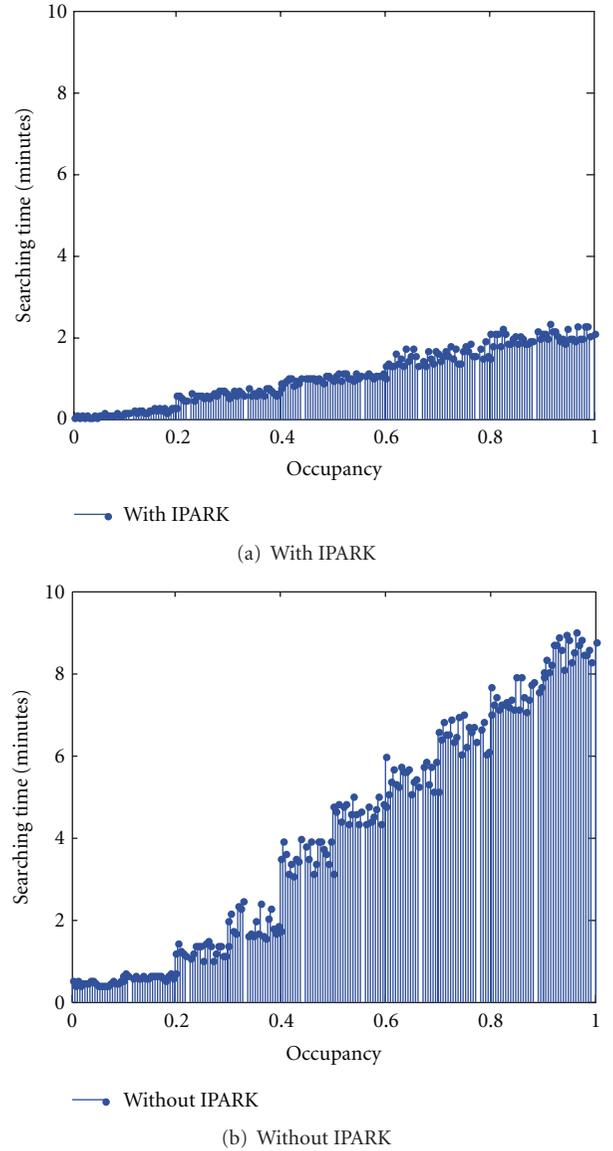
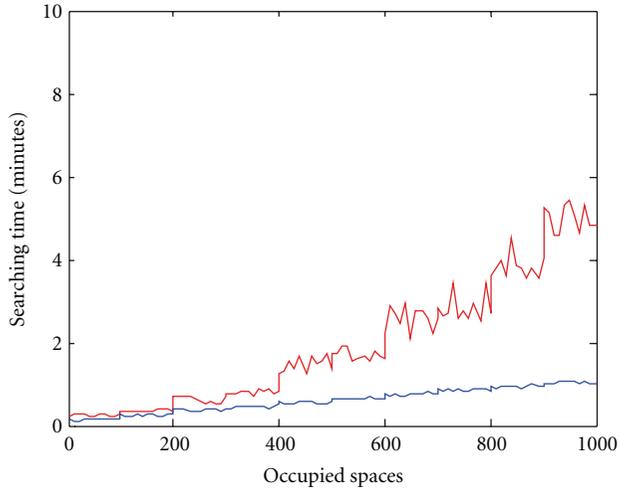


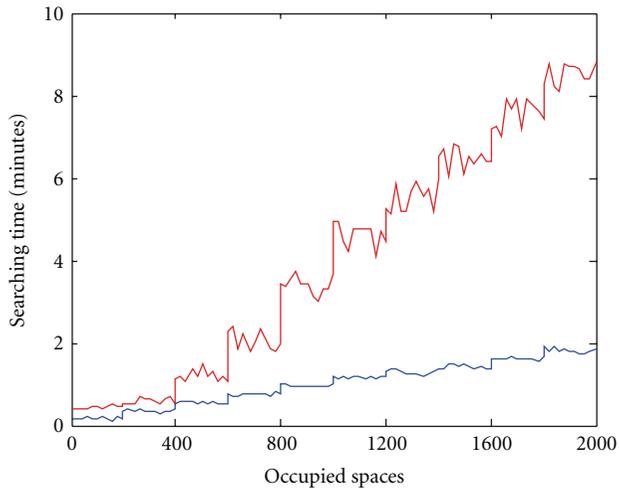
FIGURE 11: Searching time under different parking occupancy.

Denote the mean error at the  $i$ th location by  $\mu_i$ ,  $i = 1; 2; \dots; N$ , then, the sample mean is  $\bar{\mu} = (1/N) \sum_{i=1}^N \mu_i$ , and the sample variance is  $\sigma^2 = (1/N) \sum_{i=1}^N (\mu_i - \bar{\mu})^2$ . The possibility density function of  $\text{err}(i)$  is  $p(x) = (1/\sigma\sqrt{2\pi})e^{-(x-\bar{\mu})^2/2\sigma^2}$ , and the cumulative distribution function is  $F(x, \bar{\mu}, \sigma) = (1/\sigma\sqrt{2\pi}) \int_{-\infty}^x e^{-(t-\bar{\mu})^2/2\sigma^2} dt$ .

Based on the above analysis, we investigate the error distribution of the collected samples, with the results are shown in Figures 9(a) and 9(b). Figure 9(a) is a scatter plot showing the estimated relative distance against the actual relative distance of the sample spaces. Figure 9(b) is the cumulative distribution function of the estimation error. We find that about 90% of the estimation error is less than 1.5 m, which is small enough to uniquely determine a parking space.



(a) 1000 spaces



(b) 2000 spaces

FIGURE 12: Searching time under different capacity.

The above experiment illustrates that our logical location estimation scheme works extremely well in large parking lot. As our parking lot map generation scheme directly relies on the reported logical location of the parked vehicles, we could conclude that the parking lot map could be automatically establish over long periods of time using our scheme.

To determine the threshold for reference point selection, we carry out another group of experiment. As the above parking lot does not have more than 50 consequent spaces along the same road segment, we take a road with a length of 200 m as a line of parking spaces. We assume that there are 80 parking spaces along this road (with each parking space has a width of 2.5 m) and sample similarly like the way in the parking lot. Finally, we analyze the error distribution of

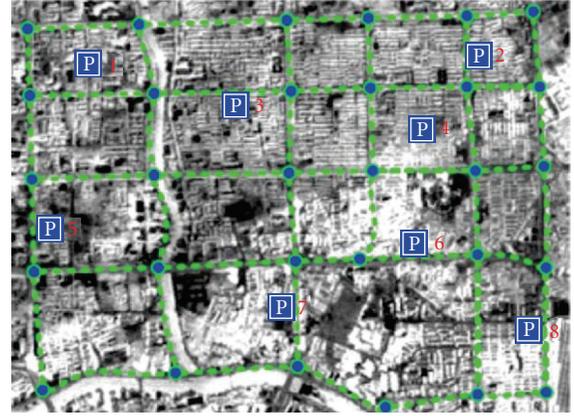


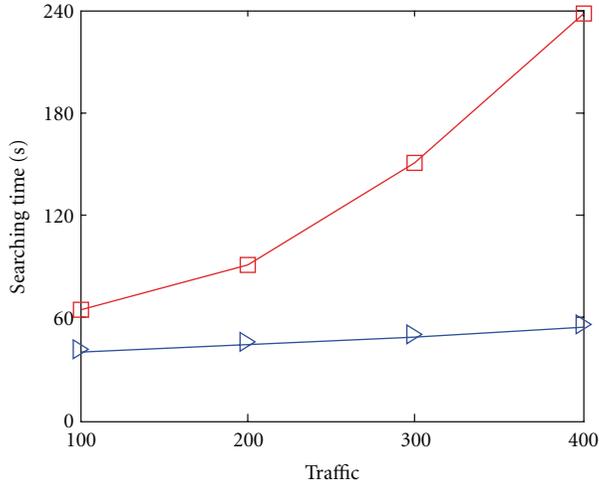
FIGURE 13: Map used in our simulation.

these samples, with the results shown in Figures 10(a) and 10(b). We find that within 40 spaces (100 m), 90% of the estimation error is less than 1.5 m. Within 60 spaces (150 m), only 80% of the estimation error is less than 1.5 m. Within 80 spaces (200 m), the situation becomes even worse. Therefore, we take 100 m as the threshold. Out of this threshold, the reference point has to be redefined.

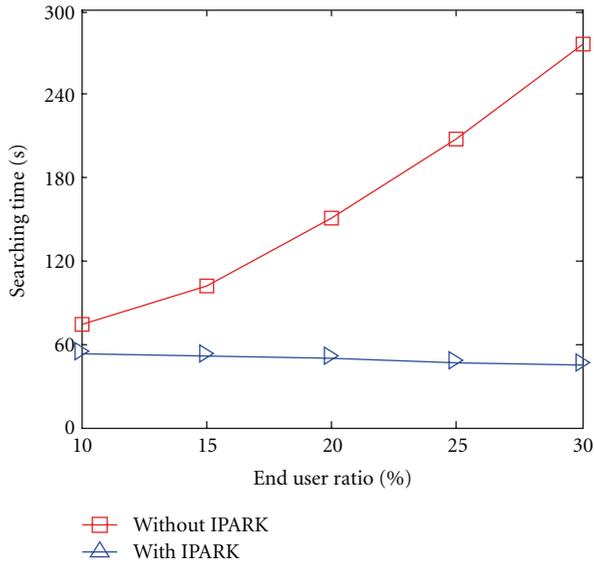
**5.2. Simulations.** In this section, simulations are conducted in NS-2.33 to verify the effectiveness of the proposed IPARK scheme, including the parking navigation inside the parking lot and the parking data dissemination outside the parking lot.

**5.2.1. Parking Navigation.** For the parking navigation, we investigate the searching time of a vacant parking space with and without the assistance of IPARK, respectively. The large parking lot used in our simulation is the one in Figure 8. For simplicity, we assume that a driver prefers to park in an available parking spot close to the entrance. After entering the parking lot, each vehicle is driving with a randomly fluctuated speed in a range of 10 percent centered at the parking lot speed limit. With SPARK, the driver will drive to a vacant parking space closest to the entrance directly. Without SPARK, the driver will be searching for a parking spot close to the entrance of the mall and keep circling around until finding the nearest available parking space.

First, we investigate the impact of the occupancy ratio of the parking lot on the searching time. We test in the parking lot with IPARK navigation and without IPARK navigation. For each case, we test 20 times, and the average searching time is considered. As shown in Figure 11, without IPARK navigation scheme, with the increase of the occupancy ratio, the searching time (ST) for a vacant parking space increases significantly after the occupancy ratio reaches 40%. Especially, when the occupancy ratio is above 80%, the time it takes a driver to find a vacant parking space is too long. However, with the assistance of the proposed parking navigation scheme, the searching time (ST) for a vacant parking space becomes much lower. Second, we investigate the impact of the capacity of a parking lot on the searching



(a) ST under different traffic



(b) ST under different end user ratio

FIGURE 14: Searching time of an available parking lot.

time. We study the searching time under 1000 spaces and 2000 spaces, respectively. As shown in Figure 12, with the assistance of IPARK, the searching time just increase slightly with the capacity of the parking lot changing from 1000 to 2000. However, without the assistance of IPARK, the searching time increase obviously with the capacity of the parking lot varying from 1000 to 2000.

**5.2.2. Parking Data Dissemination.** For parking data dissemination, we study the searching time of an available parking lot with and without IPARK, respectively. As shown in Figure 13, we extract a regional urban area with the range of 3200 m  $\times$  2200 m from a real street map of Chengdu. We deploy different vehicle numbers, that is, 100, 200, 300, and 400, to the map, with the average speed ranges of 40 to 80

TABLE 1: Capacity of the eight parking lots.

$i$	1	2	3	4	5	6	7	8
Capacity	124	46	62	136	37	54	42	58

kilometers per hour. Among these vehicles, 20% of them have a random destination within this area and they would find a parking spot to park while arriving at the destination. For the other 70% vehicles, they move back and forth within this area. According to our survey, there are 8 parking lots within this area, with 6 roadside parking lots and 2 off-street parking lots, as are marked in Figure 13. The capacity of each parking lot is listed in Table 1. Specially, for the roadside parking lot, the initial occupancy ratio is set as a random value between 90–95%; for the off-street parking lot, the initial occupancy ratio is set as a random value between 80–85%. During the simulation, the occupancy ratio of each parking lot changes with the parking behaviors of the vehicles. The radio range is set at 250 m, and the MAC protocol is 2 Mbps 802.11. The beacon interval is 1 s, and all messages have a uniform size of 1 kb. We assume that the parking lot clusters are established at the beginning of simulation, and they share the parking data with each other at an interval of 60 seconds.

With IPARK, a vehicle could get parking data from a parking lot by the means presented in Section 4. Once obtaining the parking data, the driver would directly drives towards the closets parking lots with available parking spaces. Without IPARK, a vehicle would drive towards the closest parking lots while having parking demand. While arriving at a parking lot, if the current parking lot has no vacant parking space, the driver has to leave for the second closest parking lot. This process will be repeated until a parking lot with available parking spaces is founded.

The searching time is defined as the time interval between the instant when a driver has a parking demand and the instant when it arrives at one of the parking lots with available parking spaces. The average searching time is the mean value of that of all the vehicles searching for parking spaces. We test the average searching time of an available parking lot with and without IPARK under different vehicle density, with the results are shown in Figure 14(a). We notice that with the assistance of IPARK, it takes a driver less than 55 s to arrive at a parking lot with available parking space, which is tolerable to most of the drivers. However, without the assistance of IPARK, the searching time of an available parking lot is 65–238 seconds. In addition, we observe that while the vehicle number varying from 100 to 400, the searching time with IPARK increases slightly, while the searching time without IPARK increases obviously. Moreover, we investigate the impact of end user ratio on the searching time delay of the two schemes. As reported in Figure 14(b), with the growing of end user ratio, the searching time is decreased. This is because a vehicle could acquire the response more easily with more end user. Without IPARK, with the growing of end user ratio, the number of vehicle searching for parking spaces is increased, and the probability for the parking lot being full is also increased. Thus, the searching time becomes even longer.

## 6. Conclusion

We propose IPARK to achieve intelligent parking guidance over infrastructureless VANETs. The basic idea of IPARK is simple: if some of the parked vehicles are willing to provide PVA (Parked Vehicle Assistance) services, why not let them obtain the parking statistics in real time and offer guidance for vehicles searching for parking spaces? In this paper, we group individual parked vehicles within the same parking lot into one cluster, develop an automatic parking lot map generation scheme, and investigate efficient parking navigation and parking data dissemination based on the parking lot cluster. As a novel parking guidance scheme, IPARK successfully establishes the effective management and collaborative communication without the support of any infrastructure. At last, the analysis on real parking data illustrates the effectiveness of the proposed logical location estimation scheme and the simulation results demonstrate that IPARK could help a vehicle driver find a vacant parking space at a small delay.

## Acknowledgments

This work is supported by National Science Foundation of China under Grant nos. 61170256, 61103226, 61103227, 61173171, and 61272526 and the Fundamental Research Funds for the Central Universities under Grant nos. ZYGX2011J060, ZYGX2010J074, and ZYGX2011J073.

## References

- [1] D. Schrank and T. Lomax, "Annual urban mobility report," 2005, <http://www.pittsburghregion.org/public/cfm/library/reports>.
- [2] D. Shoup, "Cruising for parking," *Access*, vol. 30, pp. 16–22, 2007.
- [3] J. Chinrungrueng, U. Sunantachaikul, and S. Triamlumlerd, "Smart parking: an application of optical wireless sensor network," in *Proceedings of the International Symposium on Applications and the Internet Workshops (SAINTW '07)*, pp. 66–69, January 2007.
- [4] Y. Z. Bi, L. M. Sun, H. S. Zhu, T. X. Yan, and Z. J. Luo, "Parking management system based on wireless sensor network," *Acta Automatica Sinica*, vol. 32, no. 6, pp. 877–968, 2006.
- [5] <http://sfpark.org/>.
- [6] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: a new VANET-based smart parking scheme for large parking lots," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 1413–1421, April 2009.
- [7] R. Panayappan, J. M. Trivedi, A. Studer, and A. Perrig, "VANET-based approach for parking space availability," in *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET '07)*, pp. 75–76, September 2007.
- [8] *Advanced Parking Management Systems: A Cross-Cutting Study-Taking the Stress Out of Parking*, Federal Highway Administration, Intelligent Transportation Systems, U.S. Department of Transportation, 2007.
- [9] M. Caliskan, D. Graupner, and M. Mauve, "Decentralized discovery of free parking places," in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET '06)*, pp. 30–39, September 2006.
- [10] S. Mathur, T. Jin, N. Kasturirangan et al., "ParkNet: drive-by sensing of road-side parking statistics," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 123–136, June 2010.
- [11] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: stopped cars are not silent," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 431–435, April 2011.
- [12] S. Olariu, I. Khalil, and M. Abuelela, "Taking VANET to the clouds," *International Journal of Pervasive Computing and Communications*, vol. 7, pp. 7–21, 2011.
- [13] C. Morency and M. Trépanier, *Characterizing Parking Spaces Using Travel Survey Data*, CIRRELT, 2006.
- [14] A. Adiv and W. Wang, "On-street parking meter behavior," *Transportation Quarterly*, vol. 41, pp. 281–307, 1987.
- [15] B. Albanese and G. Matlack, "Utilization of parking lots in Hattiesburg, Mississippi, USA, and impacts on local streams," *Environmental Management*, vol. 24, no. 2, pp. 265–271, 1999.
- [16] Y. Bao, H. Xu, and Z. Liu, "Vector map geo-location using GPS tracks," in *Cartographic Theory and Models (Geoinformatics '07)*, vol. 6751 of *Proceedings of SPIE*, May 2007.
- [17] G. R. Jagadeesh, T. Srikanthan, and X. D. Zhang, "A map matching method for GPS based real-time vehicle location," *Journal of Navigation*, vol. 57, no. 3, pp. 429–440, 2005.
- [18] J. F. Kurose and K. W. Ross, *A Top-Down Approach Featuring the Internet*, 2012.
- [19] A. Vahdat and D. Becker, *Epidemic Routing for Partially Connected Ad Hoc Networks*, 2000.
- [20] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

