

Research Article

Intelligent Information Dissemination Scheme for Urban Vehicular Ad Hoc Networks

Jinsheng Yang,¹ Changqing Liu,¹ and Shih-Lin Wu²

¹School of Electronics and Information Engineering, Tianjin University, Tianjin 300072, China

²Department of Computer Science and Information Engineering, Chang Gung University, Taoyuan City 33302, Taiwan

Correspondence should be addressed to Shih-Lin Wu; slwu@mail.cgu.edu.tw

Received 3 August 2014; Accepted 23 February 2015

Academic Editor: Anna Pandolfi

Copyright © 2015 Jinsheng Yang 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), a hotspot, such as a parking lot, is an information source and will receive inquiries from many vehicles for seeking any possible free parking space. According to the routing protocols in literature, each of the vehicles needs to flood its route discovery (RD) packets to discover a route to the hotspot before sending inquiring packets to the parking lot. As a result, the VANET nearby an urban area or city center may incur the problem of *broadcast storm* due to so many flooding RD packets during rush hours. To avoid the broadcast storm problem, this paper presents a *hotspot-enabled* routing-tree based data forwarding method, called the intelligent information dissemination scheme (IID). Our method can let the hotspot automatically decide when to build the routing-tree for proactive information transmissions under the condition that the number of vehicle routing discoveries during a given period exceeds a certain threshold which is calculated through our developed analytical packet delivery model. The routing information will be dynamically maintained by vehicles located at each intersection near the hotspot if the maintaining cost is less than that of allowing vehicles to discover routes themselves. Simulation results show that this method can minimize routing delays for vehicles with lower packets delivery overheads.

1. Introduction

Vehicular ad hoc networks (VANETs) have recently emerged as a promising research area for improving transportation safety and efficiency [1, 2]. For example, VANETs can be used to provide drivers with information related to traffic, weather, parking availability, and local shopping and amenities. They can also provide entertainment services and advertisements.

Data forwarding schemes are a key research issue in VANETs and have received considerable attention recently [3–7]. VADD [3] proposes a representative data forwarding protocol for VANETs, which takes advantage of road layout and traffic statistics to select the next road segment with the smallest estimated end-to-end delay. At street intersections, the forwarding algorithm selects the next hop by choosing either the vehicle that is closest to the road segment or a vehicle that is moving towards the road segment. SADV [4] gives better data delivery performance with the help of static nodes, for example, road side units (RSUs), deployed

at intersections. In [5], the authors deliver the idea that uses vehicular trajectory information to enhance routing efficiency between RSUs and vehicles. References [4, 5] need the support of fixed facilities for data forwarding. In this paper, we do not take the availability of RSUs into consideration because the fixed facilities incur increased costs and are unavailable in the most areas currently. RBVT [6] designs an intersection-to-intersection based routing protocol. When a source vehicle needs to send data, RBVT creates a route discovery (RD) packet to initiate a route discovery process. RBVT is a source routing protocol so the routing information (i.e., the sequence of intersections) is contained in the header of every data packet forwarded by intermediate nodes between intersections. To reduce the overloading problem by beacon exchanges, [7] proposes another intersection-to-intersection based routing protocol which designs a method for forwarding packets in an urban area that selects a next relay vehicle at each road intersection based on distance to destination, signal strength, and direction of movement.

All of the above references only consider the individual routing problem which only focuses on how to build peer-to-peer routes for data transmissions. During rush hours in an urban area or city center, many drivers need to know where they can park their vehicles such that many vehicles need to flood RD packets to a parking lot to build routes before they can inquire the parking information. As a result, the problem of *broadcast storm* [8] will happen due to so many RD packets in the network. The problem motivates us to design a better method to disseminate information efficiently.

This paper presents a data forwarding scheme by which a hotspot can automatically disseminate information to nearby vehicles to avoid the broadcast storm problem incurred by vehicles' flooding RD packets. A hotspot is an information source which needs to broadcast message to nearby vehicles, for example, a parking lot with free spots, a vehicle blocked at traffic jam roads, a shopping mall with advertisement messages, and so on. The data forwarding scheme enables the hotspot to dynamically determine when to flood routing-tree building (RB) packets to build a routing-tree for transmitting information to vehicles. The route information of the routing-tree can be dynamically maintained by passing vehicles at each intersection near the hotspot for a time period. There are two problems needed to be discussed. The first problem is to determine when the hotspot should proactively broadcast the RB packets. To solve this problem, we propose an analytical model to calculate the expected packet delivery overhead for a random vehicle in the network and thus flood RD packets to the hotspot. We can also calculate the total packet delivery overhead of each hotspot broadcasting RB packets throughout the network. If a hotspot receives more than γ RD packets during period τ , it should proactively broadcast the RB packet, where γ is the ratio between the total and the expected packet delivery overhead. The other problem is how to use the broadcasting RB packets to build and maintain the routing-tree at intersections by passing vehicles. To solve this problem, we develop an algorithm that allows vehicles at each intersection to maintain the routing information for a period of time. The vehicles can then update the routing-tree if they find better routes through exchanging information with neighboring intersections. In addition, vehicles can decide to discard the routing information if the cost of maintaining the routing-tree is higher than that of allowing vehicles to discover routes themselves.

The rest of this paper is organized as follows. Section 2 describes the problem formulation. Section 3 describes the design of the packet delivery overhead model. Section 4 explains the design of our routing protocol including the building and maintenance of the routing-tree and data forwarding process. Section 5 evaluates the proposed scheme. Section 6 concludes this paper.

2. Problem Formulation

During working hours, a road network with a hotspot, such as a parking lot, will receive inquiries from many vehicles seeking available parking spaces. With current data forwarding schemes proposed for VANETs [3–7], each of the vehicles needs to flood its RD packets to discover its individual

route to the hotspot before sending data packets. As a result, the VANET may be overloaded by too many flooding RD packets. In this situation, a better solution is to have the parking lot proactively broadcast RB packets to build a routing-tree for the delivery packets between vehicles and the parking lot, illustrated as the blue-dashed lines in Figure 1. The information of the routing-tree can be maintained at each intersection through the cooperation of passing vehicles. When a vehicle needs the information of parking spaces from the parking lot, it can quickly obtain the routing information from vehicles located at the intersections in the routing-tree.

The research problem is to provide routes with minimum delays for vehicles while minimizing packet delivery overhead. Our work is based on the following assumptions.

- (i) Each vehicle and hotspot is equipped with a short range wireless communication device, such as an IEEE 802.11p device [9]. The development of IEEE 802.11p was driven by the allocation of the dedicated short range communications (DSRC) [10] spectrum band in the United States. Many automakers, such as GM and Toyota, have plans to install DSRC devices in their vehicles.
- (ii) Vehicles are equipped with a global positioning system (GPS), which is already available in most new cars, allowing vehicles to know their own positions and speed and synchronize with each other by GPS.
- (iii) Vehicles are equipped with digital road maps and a navigation system, which provide the positions of the neighboring intersections and driving directions. Such systems are already commercially available (e.g., [11]).
- (iv) GPS and digital road maps can determine traffic statistics such as the average vehicle speed v of a road segment at any given time.

The following section explains our packet delivery overhead model and the intelligent routing-tree based data forwarding scheme.

3. Packet Delivery Overhead Model

This section analyzes the packet delivery overhead model which is defined as the forwarding times for a packet transmitted over the road network. Without loss of generality, we consider a diamond-shaped grid road network with a hotspot located in the center because the number of road segments from any farthest intersection (as shown in black circles) to the center has the same value, as shown in Figure 2. Packets generated from the farthest intersections can be delivered to the hotspot through a number of road segments, denoted as n . The length of each road segment is denoted as l . If a road segment has only one shortest route to the hotspot (i.e., the blue roads shown in Figure 2), vehicles on this route do not need to flood RD packets. If vehicles are located on other road segments (i.e., the yellow roads shown in Figure 2), they need to directionally flood RD packets to discover a route with

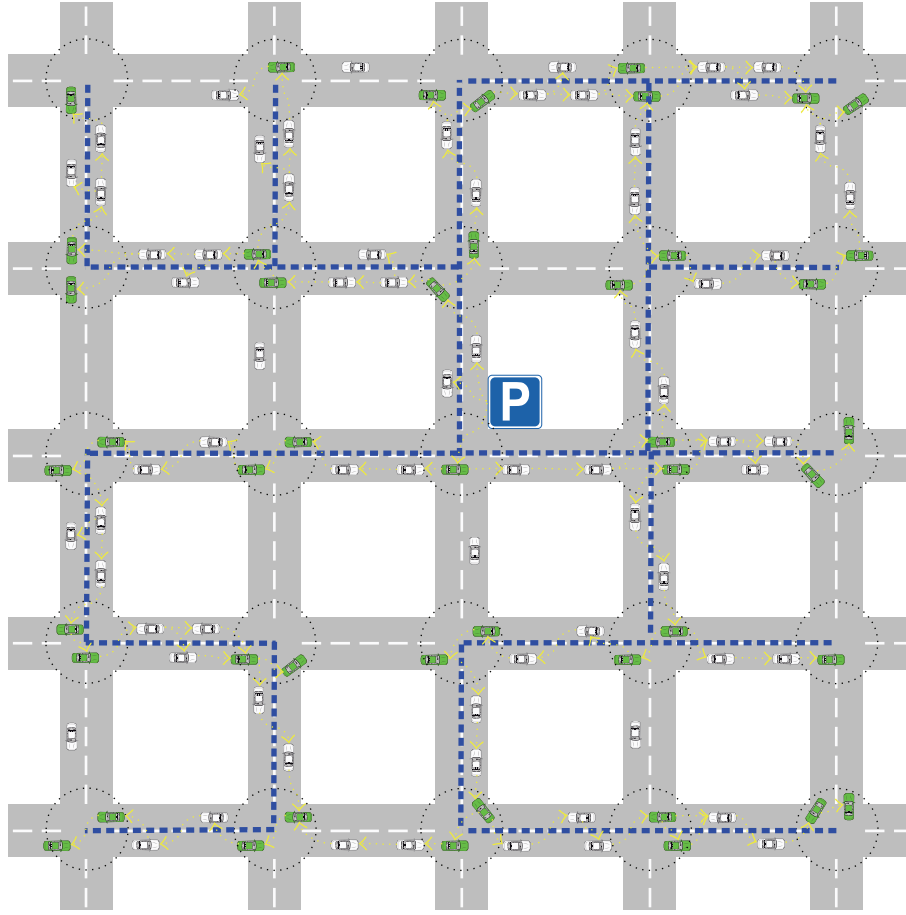


FIGURE 1: The parking lot broadcasts RB packets to build the routing-tree for the VANET (as shown in blue-dashed lines), and the routing information is maintained by vehicles at each intersection (as shown in green vehicles in black-dashed circles). Whenever a vehicle needs the parking information from the parking lot, it can inquire the information from vehicles located in its nearby intersection.

minimum delivery delay. The directional flood means that the flooding packets are only flooded into all possible routes with shortest distance to the hotspot. In this model, we assume that the wireless transmission range of DSRC devices is fixed so the packet delivery overhead can be approximated as the packet delivery distance which is defined as the number of road segments to the hotspot. In the following, we first derive the probability that a vehicle needs to flood RD packets when located on a particular road segment and then we give the expected packet delivery overhead.

Assume that vehicles are uniformly distributed in the road network. The packet delivery distance between a vehicle and the hotspot is a random variable, denoted as X , and k is a positive constant integer. In Figure 2, for example, the packet delivery distance between the red car and the hotspot X is equal to $2.5 \times l$; that is, $2 \times l \leq X \leq 3 \times l$. Note that, in the diamond-shaped road network, the number of road segments with a distance X is $4k + 4(2k - 2)$ if $k \times l \leq X \leq (k + 1) \times l$, but only $4(2k - 2)$ of these (located in the yellow roads) need to flood the RD packets while $4k$ of these (located in the blue roads) do not need to flood according to the above assumption. Thus, the probability that a vehicle needs to flood

RD packets when located on a road segment with the distance X between $k \times l$ and $(k + 1) \times l$ can be calculated as

$$P_k = \frac{4(2k - 2)}{\text{total_road_length}}, \quad 1 < k < n, \quad (1)$$

where the denominator total_road_length is the total number of all road segments in the diamond-shaped road network and is equal to $4n^2l$. Note that k is larger than 1, because the road segments with a distance X between l and $2l$ do not need to flood RD packets. The *expected delivery length* (EDL) for an RD packet flooded by a vehicle located on a road segment with a distance X between $k \times l$ and $(k + 1) \times l$ is denoted as EDL_k . Since the RD packets are directionally flooded by the vehicles and require a reply from the hotspot, EDL_k is calculated as

$$\text{EDL}_k = \int_{kl}^{(k+1)l} [2x + F(k)l] \times \frac{1}{l} dx, \quad 1 < k < n, \quad (2)$$

where dx is a very short distance and $F(k)$ is the additional number of road segments over which a packet is delivered because of the directional flooding. Since the vehicle needs to receive a route reply (RR) packet from the hotspot,

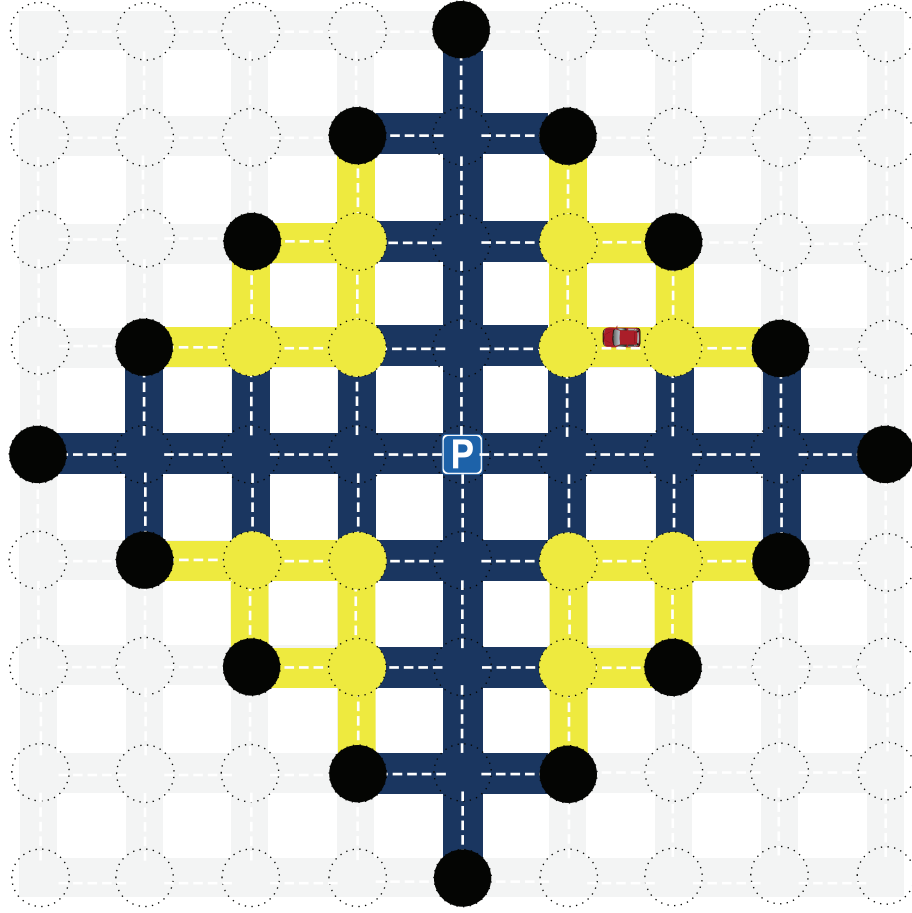


FIGURE 2: Intersections with black circles form a diamond-shaped grid road network with a hotspot located at the center (marked with “P”). These black intersections have the same number of road segments to the center and the number is defined as a network scale, n (in this example, $n = 4$).

the variable x is multiplied by a factor of 2. As can be seen from Figure 2, packets can be flooded into different network topologies even if they are generated from road segments equidistant from the hotspot. Different network topologies result in different degrees of additional packet delivery distance. Thus, we have

$$F(k) = \begin{cases} \frac{2}{k-1} \sum_{i=1}^{k/2-1} [2i(k-i)] + \frac{1}{k-1} \left[k \left(\frac{k}{2} + 1 \right) - k \right], & k \text{ is even,} \\ \frac{2}{k-1} \sum_{i=1}^{(k-1)/2} [2i(k-i)], & k \text{ is odd.} \end{cases} \quad (3)$$

For example, when $k = 3$, there are 28 road segments with a distance X between $3l$ and $4l$, but only 16 (i.e., the yellow roads with red borders in Figure 2) need to flood RD packets. Packets generated on these 16 road segments will be flooded into 1×2 shaped road networks. The additional packet delivery length in a 1×2 shaped road network is 4

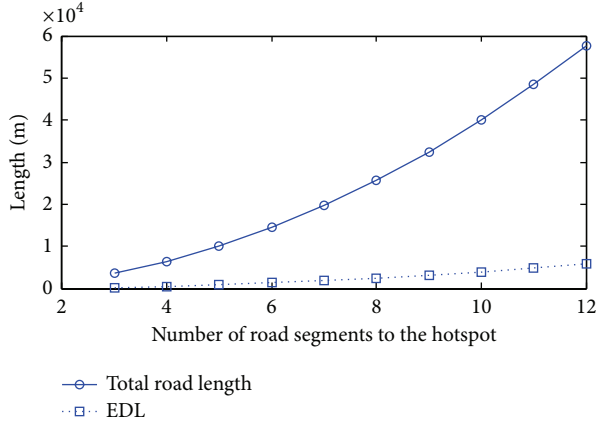
road segments, which is the same as bringing $k = 3$ into (3), obtaining $F(3) = 4$.

Thus far, we have explained how to calculate the EDL_k and the total road length of the specified road network. The EDL of the whole road network is calculated as

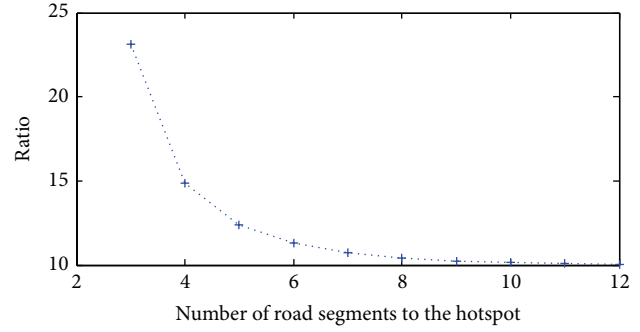
$$EDL = \sum_{k=2}^{n-1} EDL_k \times P_k. \quad (4)$$

Figure 3(a) shows the total road length of all road segments and the EDL versus different network scales.

Figure 3(b) shows the ratio between total road length and EDL versus road network scale. We can see that the ratio approaches 10 as the network scale increases. The ratio is an important factor, because it is used as a criterion by the hotspot to decide when to proactively broadcast RB packets to build a routing-tree, which will be discussed extensively in the next section. Although the analysis result is based on diamond-shaped grid network, it can be extended to any irregular network. Any hotspot can know its nearby road topology by the digital road map and check every road segment to know which road segments are needed to flood the RD packets. Although it takes some time to check every



(a) Total road length and EDL versus road network scale ($l = 100$ m)



(b) Ratio versus road network scale

FIGURE 3: Validation of the packet delivery overhead model.

Routing-tree building (RB) packet	
Source location	Sending time
Relay node location	Route list

FIGURE 4: RB packet format.

Routing table		
Destination address	Delay	Route list
Timestamp	Inquiry count	

FIGURE 5: Routing table entry.

segment, the analysis procedure can be performed and the analysis result can be saved in advance.

4. Intelligent Information Dissemination Scheme (IID)

This section explains the design of our data forwarding scheme in three parts: the building of the routing-tree, the maintenance of the routing-tree, and the data forwarding process.

4.1. Building of the Routing-Tree. Given a road network as described in Section 3, we can calculate the ratio of total road length and EDL, which is denoted as γ . If the hotspot receives more than γ inquiries during a period of time τ , the hotspot will proactively broadcast RB packets. The period τ can be determined by the time needed to build the routing-tree or the rebroadcasting period at the intersection (described in Figure 4). *Source location*, *sending time*, *relay node location*, and *route list* are included in the RB packet header, as illustrated in Figure 4, and we explain the transmission of RB packets on road segments and the delivery of RB packets through intersection zones.

To reduce the effect of the broadcast storm problem [8], an improved flooding mechanism is used as in [5]. When a vehicle receives an RB packet, it checks whether the packet

has the same source location and sending time as a previously received packet. If so, the new packet is discarded. If not, the vehicle holds the packet for a period of time inversely proportional to the distance between itself and the sending vehicle. The coordinates of the sending vehicle are known from the RB packet’s relay node location entry. Once the waiting time expires, the vehicle rebroadcasts the packet only if the packet has not been rebroadcasted by other vehicles in the interim. In this way, vehicles further away can rebroadcast the packet first, thus ensuring faster forwarding progress and reduced network traffic.

The routing-tree is built gradually. The first time the RB packet is flooded into an intersection zone (i.e., a circular region at each intersection) and received by vehicles, the vehicles append the intersection ID to the packet’s route list, while calculating the delay to the hotspot and saving this delay information and route list to the routing table. They also update the timestamp of the routing table to the current time. The entries of the routing table are shown in Figure 5. The *timestamp* and *inquiry count* entry will be used to count the number of inquiries received by an intersection in a given time period (this is discussed in detail in next section). The routing table is maintained at the intersection, which will be described in detail in next section. The vehicles then compete to determine which one is to rebroadcast the RB in the same manner as in the road segment. This process continues until all intersections in the given network with $n = 4$ have received

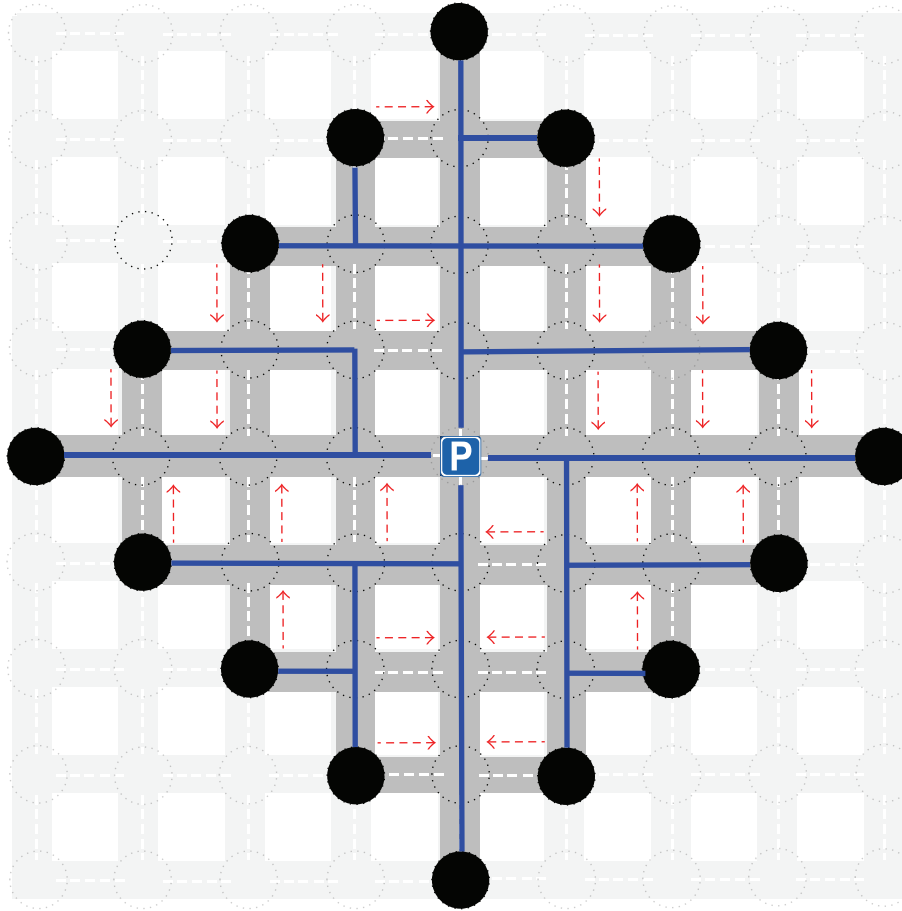


FIGURE 6: Any intersection in the constructed blue-line routing-tree has a red-dashed arrow pointed to a particular intersection which has the same shortest distance to the hotspot.

the RB packet, which means the routing-tree has been built (as the black parts shown in Figure 7).

4.2. Routing-Tree Maintenance. The routing-tree is maintained by vehicles in intersections wherein the vehicles with the routing table periodically rebroadcast the routing information. The rebroadcasting period T_i of intersection zone i is determined by the size of the diameter of the intersection zone D_i and the average vehicle speed at the intersection and can be calculated as $T_i = (D_i/2)/v_i$. To avoid collisions among transmitting packets, each vehicle generates a random backoff period, $B_{\text{uni}} \times \text{Slot-Time}$, $0 \leq B_{\text{uni}} \leq \text{CW}$ (*contention window*), for a deferral time before transmitting, and the vehicle with the smallest backoff time can transmit its packet. In fact, this method is IEEE 802.11 MAC protocol [12] and B_{uni} is a uniform random variable, in which the probability of a selected slot b is defined as follows:

$$\Pr [B_{\text{uni}} = b] = \frac{1}{\text{CW} + 1}, \quad 0 \leq b \leq \text{CW}. \quad (5)$$

However, unlike in [12], we can assign CW a value according to the number of vehicles in the intersection. Each vehicle knows its own speed which can be used to estimate

the intervehicle spacing [13] which, in turn, is used to estimate the number of vehicles in the intersection.

It should be noted that there are several ways to maintain the routing table in the intersection, such as a notification scheme. For example, when a vehicle moves into an intersection, it will send a packet to request the routing-tree information that may be kept in vehicles currently located at the intersection. The vehicles with the routing table in the intersection that receive this kind of packet will wait for a time proportional to the distance between themselves and the sending vehicle and then the vehicle with the shortest waiting time will broadcast the routing information. This notification scheme may be appropriate for low vehicle density VANETs. However, this paper considers urban road networks with relatively high vehicle density, where the periodical broadcasting scheme provides greater cost savings.

We now describe how the routing-tree is locally updated. First, the delay information in the routing-tree can be kept up-to-date with the help of reply data packets from the hotspot. Vehicles located at any intersection of the routing-tree periodically send ping messages to *particular neighboring intersections* to check whether the routing-tree requires to be updated. As shown in Figure 6, any intersection in

Route discovery (RD) packet		
Source ID	Destination ID	Sequence number
Relay node location		Route list

FIGURE 7: RD packet format.

the constructed blue-line routing-tree has a particular neighboring intersection which has the same shortest distance to the hotspot. When a vehicle in a particular intersection receives a ping message, it will check whether it possesses delay information for the hotspot. If so, it will send a reply message containing the delay information to the intersection from which the ping message was originally sent. Otherwise, it sends a reply to tell the sending intersection to maintain its original routing information. In this way, each intersection can obtain the fresh delay of a second route to the hotspot. If this delay is smaller than that contained in its routing table, the intersection will update its delay information in its routing table using the new delay data.

We now explain when an intersection should stop locally maintaining and updating the routing-tree. Assume that the period of sending the ping message is T_p , and an intersection I_i receives N_i inquiries and its road distance from the hotspot is $k \times l$. The time to begin counting the inquiries can be obtained from the timestamp in the routing table, and N_i is recorded in the inquiry count entry of the routing table, as shown in Figure 5. Each time a vehicle receives an RD packet it increases the inquiry count by 1. Whether the routing information should be discarded depends on the following criterion:

$$\frac{2l/R_c}{T_p} + \frac{1}{T_i} < \frac{N_i(F(k) + 2k)l/R_c}{T_p}, \quad (6)$$

where R_c is the transmission range of DSRC and $F(k)$ is the same as in (3). The leftmost component is the cost of exchanging information with the neighbor to update the routing table. The other component in the left part of the inequality is the cost of holding the routing information in the intersection. The right part of the inequality is the cost of allowing vehicles to discover routes themselves. If this criterion is satisfied, the intersection continues to hold the routing table. If not, it discards the routing information.

4.3. Data Forwarding. Now we explain how vehicles can use the routing-tree to deliver data packets to the hotspot in the given road network.

When a vehicle needs to forward data packets, it first checks whether it is currently maintaining routing information. If so, it can insert the route into its data packet header

and send the data. If not, it creates an RD packet to initiate a route discovery process. The RD packet is flooded in the same way as the RB packet as described in Section 4.1. As shown in Figure 7, the RD header includes *source ID*, *destination ID*, *relay node location*, *sequence number*, and *route list*. When a vehicle in an intersection receives a new RD packet, it checks whether it is currently maintaining the routing information. If so, it sends a route reply (RR) packet to the vehicle that floods the RD. The routing information in this intersection and the route recorded in the RD packet are added to the RR header, which has the same entry as the RD packet header. The RR packet is forwarded along the road segments defined by the intersections stored in its header. If not, it appends the intersection ID to the route list of RD packet and holds the packet for a period of time proportional to the distance between itself and the hotspot. It will rebroadcast the packet only if this packet has not been rebroadcasted by other vehicles.

Upon receiving the RR packet, the vehicle starts sending data. Each data packet stores the route in its header and is geographically forwarded along the route. When one of the intersections in the route list receives the data packet, it checks whether it is currently maintaining a routing table. If so, it compares the routing information with the route list in the packet header and updates the route list with the routing information in the intersection if they are different. If the intersection is not currently maintaining a routing table, the packet will be forwarded along the route described in the data packet header.

Data packets are delivered between intersections using improved flooding [5]. This forwarding method is further optimized as follows: during the initial simulation process, we set the waiting time inversely proportional to the distance between the transmitting vehicle and the forwarding vehicle. However, since the roads have a certain width, the vehicle farthest from the forwarding vehicle may be not the one closest to the next intersection. Taking this into account, we optimize the waiting time to be inversely proportional to the dot product of two vectors: the vector from the forwarding vehicle to the receiving vehicle and the unit vector from the forwarding vector to the next intersection. In this way, we can ensure that the vehicle that is closest to the next intersection in the routing table will have the shortest waiting time.

5. Performance Evaluation

In this section, we evaluate the performance of IID. As we mentioned early, most of the current literature only considers the individual routing problem which only focuses on how to build peer-to-peer routes for data transmissions. Our proposed IID presents a smart data forwarding scheme by which a hotspot can automatically disseminate information to nearby vehicles instead of many individual routing requests to the hotspot by individual vehicles to avoid the broadcast storm problem during rush hours. RBVT [6] is one of the current representative routing protocols and belongs to a reactive routing protocol. When a source vehicle needs to send data, RBVT creates a route discovery packet to initiate a route discovery process. A route created by RBVT consists

TABLE 1: Simulation parameters.

Parameter	Value
Simulation area	1600 m × 1600 m
Number of vehicles	140
Number of inquiries	5–25
Transmission range	250 m
Maximum vehicle speed	8, 16 m/s
Data packet size	512 bytes
Simulation time	250 s

of connected intersection-to-intersection road segments and each road segment between two adjacent intersections has enough vehicular traffic to ensure network connectivity. RBVT is a source routing protocol so the routing information (i.e., the sequence of intersections) is contained in the header of every data packet forwarded by intermediate nodes between intersections. Thus we can use RBVT as the individual routing protocol before the hotspot decides to build a routing-tree to disseminate information. Note that RBVT can be replaced with any individual routing protocol [3–7]. To know how much performance IID can achieve, we evaluate the performance of IID by comparing with RBVT using the network simulator NS-2.35 [14]. Below we present the evaluation method, the metrics for comparing protocol performance, and an analysis of the simulation results.

5.1. Evaluation Method. We use an open source vehicular mobility model generator, called (MOVE) [14], which is built on top of an open source microtraffic simulator SUMO [15], to generate a road layout and vehicle movement. The vehicles' behavior follows the collision-free car-following model [13, 16]. The transmission range of wireless devices is set to 250 meters. The size of road network used in simulations is a 1600 m × 1600 m area including road segments, intersections, and a hotspot. Speed limits, the number of lanes, and traffic flows are inputted into MOVE. We assign different numbers of traffic flows along different road segments. At some intersections, we assign different probabilities to different turning directions. Thus, traffic densities differ between road segments, resulting in significantly different data delivery time along different roads. In this way, a routing-tree can be built. The first 2000 seconds of outputs of the MOVE is discarded to obtain more realistic vehicle movements.

To evaluate the performance for different inquiry volumes over a given period of time, we vary the number of concurrent vehicle inquiries from 1 to 25. During the simulation process, the data transmission rate is 0.1 packets per second. Road segments are assigned speed limits of 8 or 16 m/s. The detailed simulation parameters are shown in Table 1.

5.2. Performance Metrics. The performance metrics used here are average packet delivery overhead, average packet delivery delay, average packet delivery ratio, and average throughput. Performance is evaluated by varying the number of vehicle inquiries during a period of time along with vehicle density.

5.2.1. Average Packet Delivery Overhead. The number of routing packets needed to be transmitted and forwarded for a data packet successfully received at the destination. The overhead measures the additional packet delivery traffic that the routing protocol generates for data packets that were successfully delivered.

5.2.2. Average Packet Delivery Delay. The average delivery delay of all data packets successfully is delivered to the destination. The average delay characterizes the latency generated by the routing scheme.

5.2.3. Average Packet Delivery Ratio. The number of data packets successfully is delivered to the destination per the number of data packets sent by sources. The average delivery ratio shows the ability of the routing protocol to successfully deliver data on an end-to-end basis.

5.2.4. Average Throughput. The number of bits is successfully delivered to the destination during a given period of time. The average throughput characterizes the network's rate of successful data delivery.

5.3. Simulation Results

5.3.1. Average Packet Delivery Overhead. Figure 8 compares packet delivery overhead between IID and RBVT under two maximum vehicle speeds. Figure 8(a) shows that IID outperforms RBVT because each vehicle in RBVT needs to initiate a route discovery process before sending a piece of data while vehicles in IID can quickly obtain routing information once the routing-tree is built. Figure 8(b) shows the same trend as in Figure 8(a) for similar reasons. The packet delivery overhead of IID stays nearly the same (around 10 to 20 routing packets per data packet successfully received) for the two maximum vehicle speeds; however, the average delivery overhead of RBVT increases with the maximum vehicle speed. The reasons are that, in IID, the impact of vehicle speed on packet delivery overhead is relatively small because the number of routing packets is reduced. However, in RBVT, as vehicles move faster, the network topology changes more quickly and disconnectivity between nodes increases, which means the packet delivery overhead used for managing the connectivity increases. Thus, the packet delivery overhead of RBVT is larger than that of IID regardless of the speed limit.

5.3.2. Average Packet Delivery Delay. Figure 9 compares packet delivery delay between IID and RBVT under two maximum vehicle speeds. Figure 9(a) shows that IID has a smaller packet delivery delay than RBVT and the delay of IID is nearly constant for the various different inquiries. This is because, after the routing-tree is built, in IID vehicles that want to send data to the hotspot generally already possess routing information or can quickly obtain it from the nearby intersections in the routing-tree. Therefore, in IID, vehicles can send out data packets with only a very small delay. However, in RBVT, if vehicles want to send data to the hotspot, they need to first initiate a route discovery process and can send data only after receiving the route reply packet.

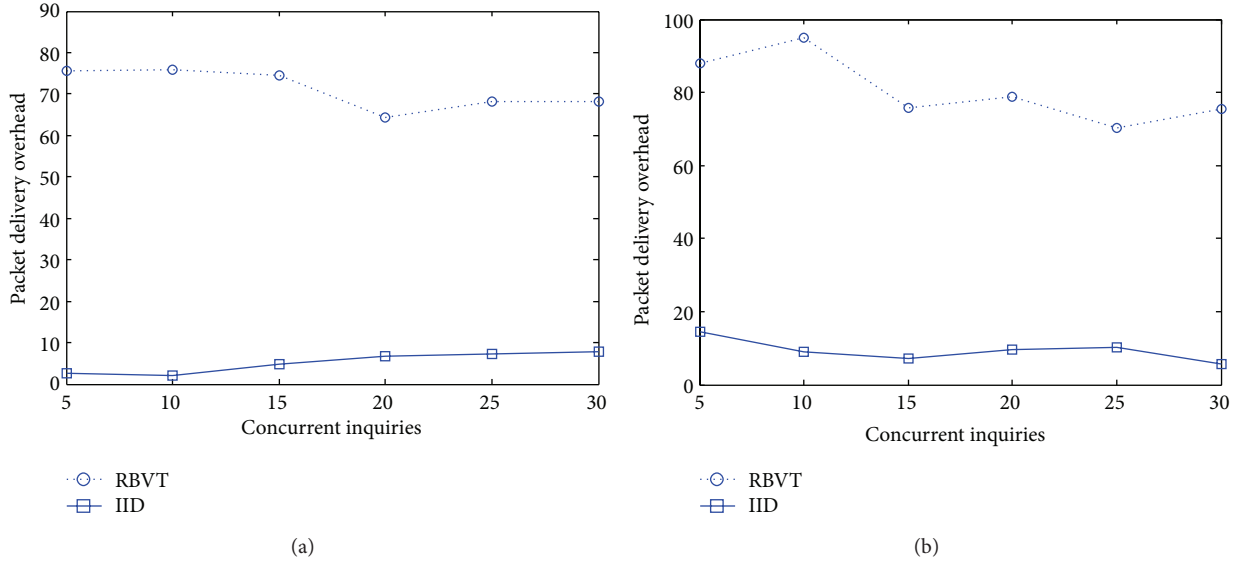


FIGURE 8: Average delivery overhead for IID and RBVT. (a) Max speed 8 m/s. (b) Max speed 16 m/s.

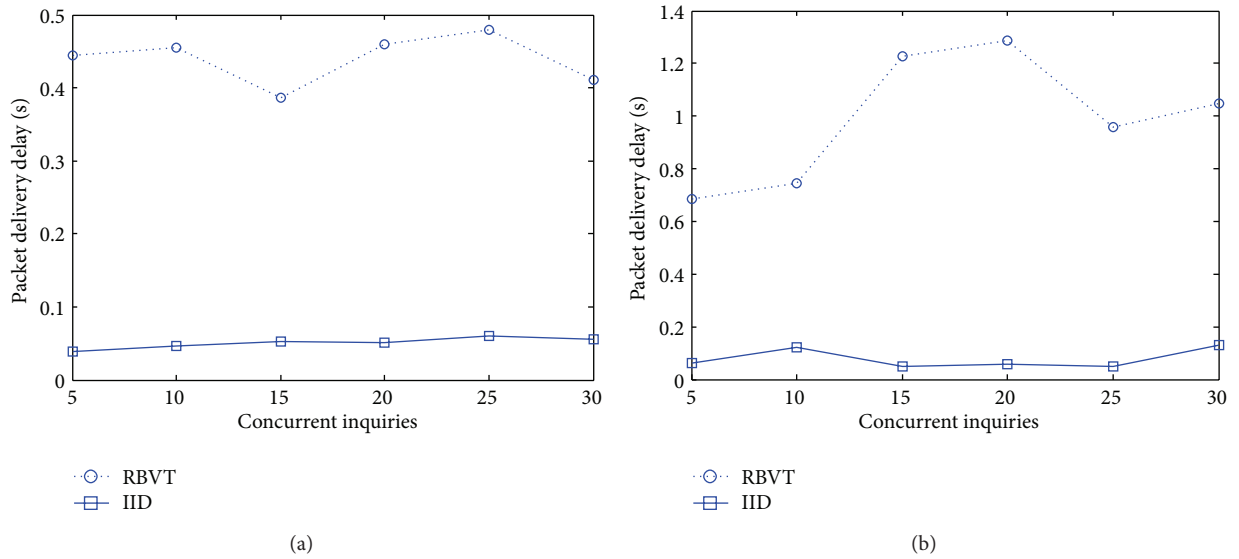


FIGURE 9: Average delivery delay IID and RBVT. (a) Max speed 8 m/s. (b) Max speed 16 m/s.

Figure 9(b) results are similar to those in Figure 9(a) for similar reasons. The packet delivery delay for IID is nearly constant (around 0.5 to 1.5 s) for the two maximum vehicle speeds, but the average delivery delay of RBVT increases with maximum vehicle speed. This is because the impact of vehicle speed on packet delivery delay is relatively small in IID because many data packets may have routes to hotspots. However, in RBVT, as vehicle speed increases, the network topology changes more quickly and disconnectivity between nodes increases, resulting in longer packet delivery delays to discover a new route. Therefore, the packet delivery delay of RBVT is larger than that of IID regardless of speed limit.

5.3.3. Average Packet Delivery Ratio. Figure 10 compares the packet delivery ratio between IID and RBVT under two

maximum vehicle speeds. Figures 10(a) and 10(b) show that the packet delivery ratios for both IID and RBVT decrease as the number of concurrent inquiries increases due to packet collisions. We also find that IID outperforms RBVT for any number of concurrent inquiries because, in the VANET, RBVT requires a large number of RD packets resulting in more packet collisions. IID, on the other hand, consumes fewer RD packets because the routing-tree is built proactively if the hotspot receives more than γ inquiries during a period of time τ .

5.3.4. Average Throughput. In Figure 11, we compare the average throughput of IID and RBVT under two maximum vehicle speeds. Figures 11(a) and 11(b) show that the average throughputs of both IID and RBVT increase with the number

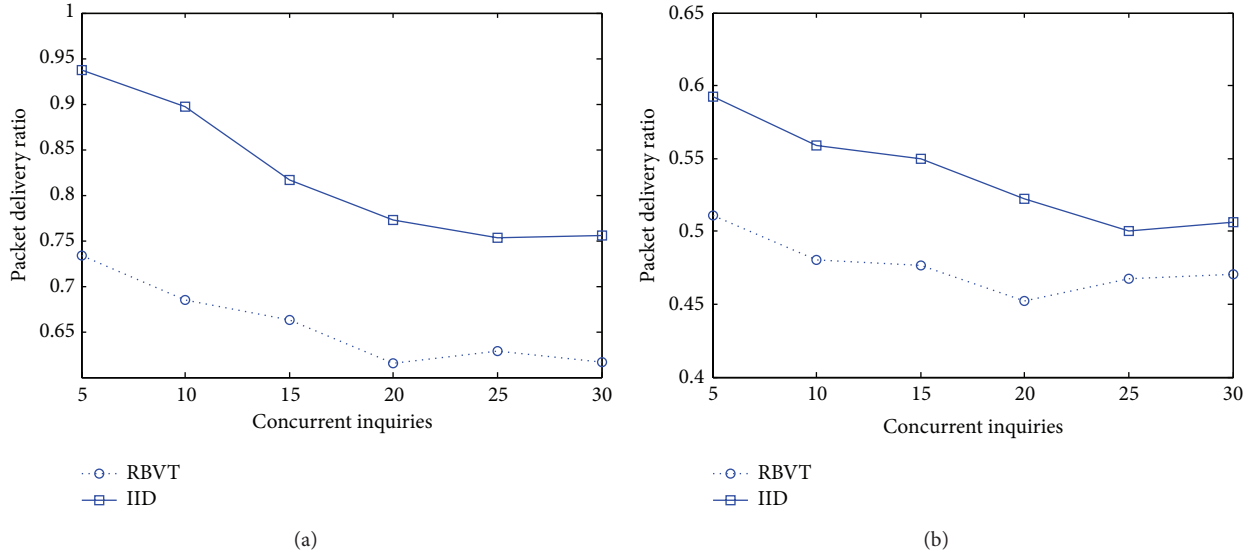


FIGURE 10: Average delivery ratio for IID and RBVT. (a) Max speed 8 m/s. (b) Max speed 16 m/s.

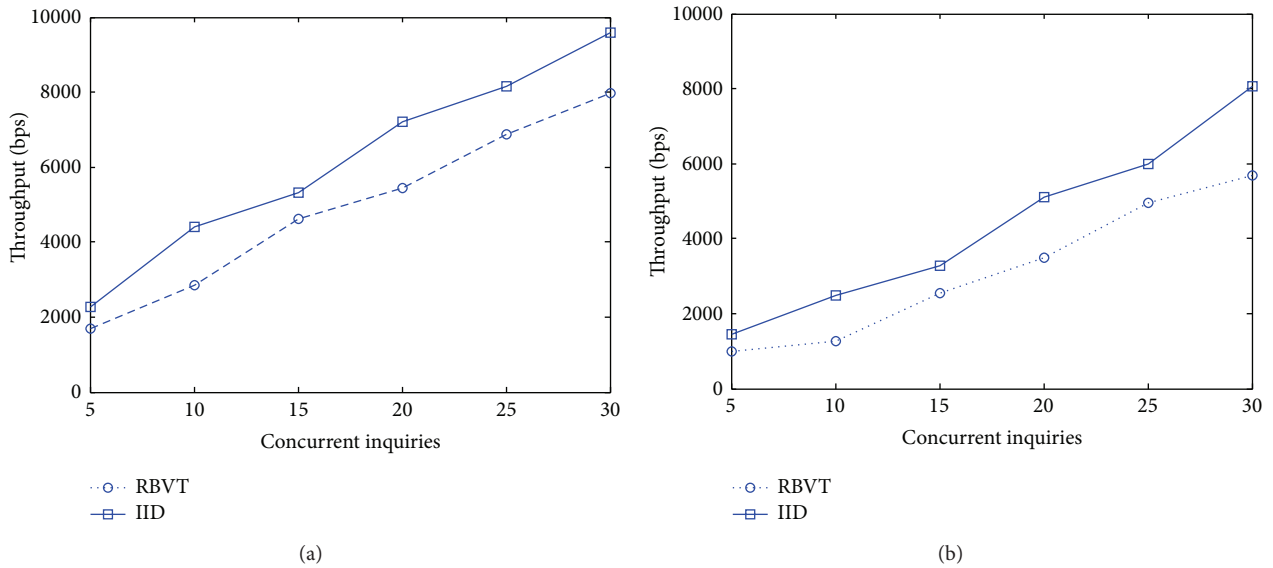


FIGURE 11: Average throughput for IID and RBVT. (a) Max speed 8 m/s. (b) Max speed 16 m/s.

of concurrent inquiries. IID is seen to outperform RBVT for any number of concurrent inquiries because IID incurs a smaller average delay and has a higher delivery ratio.

6. Conclusions

This paper proposes an IID scheme for hotspot-enabled road networks to avoid the broadcast storm problem incurred by so many flooding RD packets of the individual routing protocols [3–7] especially for rush hours. The hotspot can dynamically decide whether to build a routing-tree if it receives a critical number of inquiries over a particular time period. This critical value is calculated through the analytical traffic overhead model. The routing-tree is maintained by vehicles

periodically rebroadcasting in the intersections. Each intersection can determine whether to continue to dynamically hold and update the routing information. Since the routing-tree is built and maintained dynamically depending on packet delivery costs, this scheme produces cost savings. In addition, the routing-tree is built using real-time traffic information and is updated dynamically, providing routes with minimal delays. We can cooperate with any individual routing protocol [3–7] before our method takes over the mission to build a routing-tree to disseminate information. Simulation results show that this scheme outperforms similar schemes in terms of packet delivery overhead, packet delivery delay, packet delivery ratio, and throughput under different degrees of network loading.

Conflict of Interests

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

Acknowledgment

This work is cosponsored by the Ministry of Science and Technology, Taiwan, under Grants 102-2221-E-182-031, 103-2221-E-182-029, and 101-2923-E-182-001-MY3.

References

- [1] Q. Xu, R. Sengupta, D. Jiang, and D. Chrysler, "Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum," in *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference (VTC '03)*, vol. 4, pp. 2451–2455, April 2003.
- [2] A. Skordylis and N. Trigoni, "Delay-bounded routing in vehicular ad-hoc networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 341–350, May 2008.
- [3] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular Ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.
- [4] Y. Ding and L. Xiao, "SADV: static-node-assisted adaptive data dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2445–2455, 2010.
- [5] F. Xu, S. Guo, J. Jeong et al., "Utilizing shared vehicle trajectories for data forwarding in vehicular networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '11)*, pp. 441–445, 2011.
- [6] J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea, "VANET routing on city roads using real-time vehicular traffic information," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3609–3626, 2009.
- [7] K. Z. Ghafoor, K. Abu Bakar, J. Lloret, R. H. Khokhar, and K. C. Lee, "Intelligent beaconless geographical forwarding for urban vehicular environments," *Wireless Networks*, vol. 19, no. 3, pp. 345–362, 2013.
- [8] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2-3, pp. 153–167, 2002.
- [9] M. Amadeo, C. Campolo, and A. Molinaro, "Enhancing IEEE 802.11p/WAVE to provide infotainment applications in VANETs," *Ad Hoc Networks*, vol. 10, no. 2, pp. 253–269, 2012.
- [10] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [11] Garmin, "Garmin Traffic," <http://www8.garmin.com/traffic/>.
- [12] IEEE Std 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, 1999.
- [13] R. W. Rothery, "Car following models," in *Traffic Flow Theory*, N. H. Gartner, C. Messer, and A. K. Rathi, Eds., 1997.
- [14] F. Karnadi, Z. Mo, and K. Lan, "Rapid Generation of Realistic Mobility Models for VANET," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 2508–2513, 2007.
- [15] Centre for Applied Informatics (ZAIK) and the Institute of Transport Research German Aerospace Centre, Sumo-Simulation of Urban Mobility, <http://sumo.sourceforge.net/>.
- [16] Z. Wu, Y. Liu, and G. Pan, "A smart car control model for brake comfort based on car following," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 42–46, 2009.



Hindawi

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

