Ivy: Interest-based Data Delivery in VANET through Neighbor Caching

Tan Yan and Guiling Wang

Department of Computer Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA; e-mail: {ty7,gwang}@njit.edu

Received 20 May 2013; Accepted 15 July 2013

Abstract

In this paper, we study the problem of interest-based data delivery in Vehicular Ad Hoc Networks (VANETs), which is to efficiently forward data to a vehicle that owns the given interest without knowing its ID beforehand. Such problem is generally challenging in large-scale distributed networks, because it usually requires to query a huge number of nodes in the network to find a node of interest out of them, which is costly. To tackle the problem, we design an interest-based data delivery (*Ivy*) scheme through neighbor caching to boost the efficiency in querying information of the vehicle with given interest. We let each vehicle cache the information of all the neighbor vehicles it meets during driving, such that by just querying a few vehicles, a vehicle can retrieve the information of a large amount of vehicles, which greatly reduces the message overhead. To further reduce the message cost in route establishment, we calculate an estimated current location of the interested vehicle and forward data towards the calculated location, which avoids broadcasting blindly to search for the vehicle. Simulation result shows *Ivy* is both efficient and effective, and outperforms existing data delivery schemes with higher delivery ratio, lower delay and smaller message overhead.

Keywords: Interest-based, data delivery, neighbor caching.

Journal of Cyber Security and Mobility, Vol. 2, 127–149.
doi 10.13052/jcsm2245-1439.222

1 Introduction

A Vehicular Ad hoc Network (VANET) is an on-road network where vehicles exchange and deliver data through multi-hop communication. Data delivery in VANET is envisioned to be applied into many promising applications for both safety [23] and non-safety [19] proposes. To improve the delivery efficiency, quite a few routing and data dissemination schemes have been designed to effectively forward data to vehicles with known identity [26] or in a given location/area [35, 20].

However, the current data delivery building blocks are not sufficient to support many applications in VANETs, which generally require to deliver data to vehicles based on their interests. For example, in data dissemination services [32], a disseminator wants to find a vehicle interested in such services (e.g., subscribes to the services) to relay the dissemination. In a rural area far from hospitals, a driver in a crashed car wants to find a vehicle interested in providing on-road rescue (e.g., the driver is a doctor or a mechanical technician) and ask for help. We call such applications, *interest-based* data delivery, in which, the data source needs to deliver data to a vehicle that has the given interest, and does not know its ID or location beforehand. Traditional data forwarding mechanisms fail to effectively support interest-based data delivery, since they require the source to have a priori knowledge of either the ID or the location of the destination. Moreover, even the source knows the ID and does not know the location of the destination, it still has to broadcast intensively to all the directions to build a route with destination, which is very costly.

In addressing the problem of a source delivering data to a destination with given interest, the primary objective is to reach the destination with small overhead and low delay. A solution direction could be first querying and obtaining the ID and the current location of the interested vehicle, and then making the source only forward data towards the obtained location to reach the vehicle, which avoids broadcasting blindly to the network and greatly reduces the message overhead. However, the costly query procedure brings us a challenge in making the solution feasible in distributed networks, especially when the network has a large number of vehicles and only a few of them are with the given interest. The source vehicle has to propagate the query message to almost the entire network in order to find the information of an interested vehicle. In addition, as vehicles are of high mobility, obtaining an accurate location of an interested vehicle is even more challenge. For example, to get the most up-to-date location of a vehicle, the only way is to ping this vehicle and obtain its current location, which simply brings us back to the original problem: how to reach this vehicle? Therefore, in reaching a vehicle with given interest, an efficient method that obtains the ID and the current location of the interested vehicle is needed.

This paper focuses on designing a distributed interest-based data delivery protocol, which delivers data to a vehicle that has given interest with small message overhead and low delay. To be more specific, our goal in this paper is to solve the following problem: *In a network where each vehicle has a collection of interests, given a specific interest, how to efficiently identify and reach a vehicle that owns the given interest without knowing its ID or location beforehand?*

We propose ***i***nterest-based data delive***ry*** (*Ivy*) to deliver data to a vehicle with given interest through neighbor caching. *Ivy* first efficiently queries and obtains the information, such as the ID and the current location, of the vehicle with given interest, and then uses such information to establish a route to reach the vehicle with conserved message overhead. To facilitate vehicle information query, we let each vehicle include its information into redesigned beacon messages, through which, neighbor vehicles can exchange and store the information for each other. To boost the query efficiency, we design a Cached Neighbor List for each vehicle to cache the beacon information received from all the vehicles it meets during driving, so that each vehicle in the network can store information of many other vehicles. In this way, we can retrieve the information of a large amount of vehicles by just querying a few of them, which greatly reduces the message overhead in obtaining the information of the interested vehicle. To reduce the message cost in route establishment, we estimate the current location of the interested vehicle by analyzing the location, the speed and the driving direction in the cached information, and we only forward data towards the estimated location to build a route with the interested vehicle. To cope with the vehicle mobility, we keep querying and updating the information of the interested vehicle during data forwarding, and use such information to dynamically adjust the estimated location and the data forwarding direction. Simulation result shows *Ivy* is both efficient and effective in reaching a vehicle with given interest, and outperforms existing data delivery schemes with higher delivery ratio, lower delay and smaller message overhead.

The rest of this paper is organized as follows. We introduce the assumption and the overview of the *Ivy* in Section 2. The detailed protocol of *Ivy* is presented in Section 3. Section 4 describes the enhanced scheme with location refinement. An evaluation of *Ivy* is given in Section 5. Section 6 presents the related work. Finally, we conclude the paper in Section 7.

2 Assumptions and Scheme Overview

2.1 Assumptions

We assume vehicles communicate with each other through a dedicated short-range wireless channel (DSRC [1]) (100 to 250 m) in VANET. Each vehicle is preloaded with a digital street map, and knows its location that can be obtained through GPS device or various on-road localization mechanisms [33]. Vehicles can find their neighbors through periodical exchange of beacon messages, which can be done efficiently with various of protocols such as S-Aloha [12, 13].

We assume each vehicle tags itself with labels of interests showing the services it is interested in or the applications it subscribes, which is widely used in many publish/subscribe and social systems [28]. Vehicles discloses their interests to the network in order to provide/receive the services they subscribe.

2.2 Overview of *Ivy*

Ivy is a distributed protocol in VANET to make the source vehicle efficiently identify and obtain the information of a vehicle that has the given interest, estimate the current location of the vehicle, and deliver data to it with conserved message overhead. As the network may only have a small number of vehicles with given interest and they are randomly distributed in the network, to obtain their information, literally we have to search and check the information of a large number of vehicles before finding the information of an interested vehicle. Moreover, since each vehicle in the network only has the knowledge of limited number of vehicles in the network, we need to send messages to query multiple vehicles in order to hit the one that has the information of the interested vehicle. Thus, to reduce the message overhead in obtaining the information of an interested vehicle, we need to make each vehicle store the information of as many vehicles as possible, so that we can retrieve the information of a large number of vehicles by only querying a few of them.

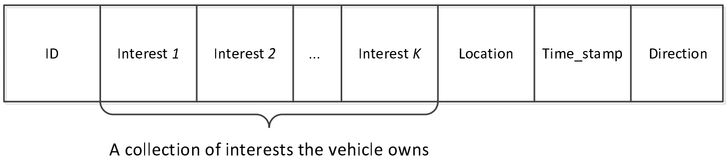
We observe that a vehicle driving around an area meets different vehicles. Through beacon message exchange, the vehicle can obtain the information of the vehicles it meets. The more vehicles it meets, the more vehicle information it can possibly obtain. Inspired by this, we let each vehicle embed its information, such as ID, interests, location and driving direction, into the beacon messages and exchange it with neighbors. We design a Cached Neighbor List for each vehicle to cache the beacon messages received from neighbor vehicles it meets when driving around the area, which stores the information of each neighbor vehicle at the time they meet. When a source needs to reach a vehicle with given interest, it broadcasts query messages looking for information of the vehicle that has the given interest. Upon receiving the message, each vehicle searches in its Cached Neighbor List, and replies to the source if it finds any vehicle whose interests contains the given interest. After collecting all the reply messages, the source obtains the ID of the interested vehicle, as well as the location and the driving direction that are recorded at the time the vehicle meets the reply sender. To calculate the current location of the interested vehicle, we first analyze the traffic of the area and calculate a speed range of the vehicle. Then we compute a region, which is a collection of all the locations the interested vehicle can reach under the calculated speed range. The calculated region is used as the estimation of current location of the interested vehicle. We further propose several heuristics to refine the region, such as by applying driving direction of the vehicle and by aggregating information from multiple reply messages. Finally, we route the message towards the estimated region and broadcast in the region to reach the vehicle.

As vehicles are highly mobile, it is possible that when the message reaches the estimated region, the interested vehicle has already moved out of the region and cannot receive the broadcasted message. To mitigate this issue, we enhance *Ivy* with location refinement (*Ivy-LR*), which dynamically adjusts the estimated region and the data forwarding direction by keeping querying the information of the interested vehicle when forwarding data.

In the following, we present the detailed design of *Ivy* and *Ivy-LR* in Sections 3 and 4, respectively.

3 Ivy: Interest-based Data Delivery through Neighbor Caching

In this section, we present Interest-based Data Delivery (*Ivy*) through Neighbor Caching. We first introduce the structure design of the beacon message and the proposed Cached Neighbor List, through which vehicles exchange their information and cache the received neighbor information. Then, we present interest-based information query for the source vehicle to efficiently



ID	Interest *1*	Interest *2*	...	Interest *K*	Location	Time_stamp	Direction

Figure 1 Structure of the beacon message.

obtain the information of a vehicle that has the given interest. After that, we design location estimation to calculate the current location of the interested vehicle based on the obtained information. Finally, we propose data delivery to the estimated location to reach the vehicle.

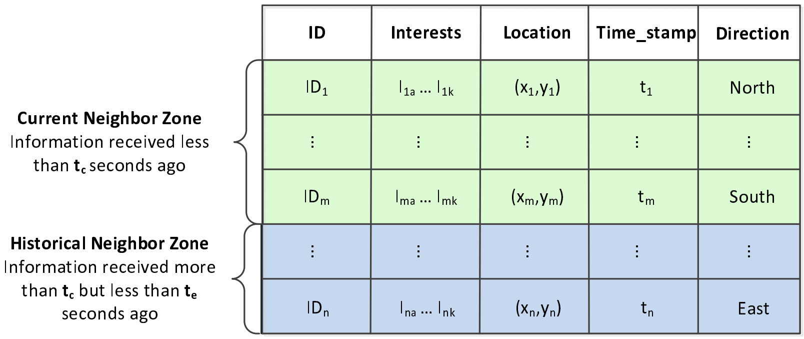
3.1 Structure of Beacon Message and Cached Neighbor List

Vehicles obtain their neighbor information through the exchange of beacon messages. To facilitate querying the vehicle with given interest, we redesign the structure of the beacon message by adding the *ID*, *Interests*, *Time-stamp*, *Location*, and *Driving direction* fields to it. Figure 1 shows the structure of the beacon message. The detailed description of each field is shown as follows:

- *ID*: the ID of the vehicle that sends the beacon message.
- *Interests*: the list of interests the vehicle wants to disclose to public, which can contain multiple elements. For example, a driver may want to add ‘Game’ and ‘Spanish’ into this field to subscribe to game and Spanish services/channels.
- *Time_stamp*: the time when the beacon message is sent out.
- *Location*: the position of the vehicle when the beacon message is sent out.
- *Direction*: the driving direction of the vehicle, i.e., North, South, East or West.

During driving, each vehicle periodically broadcasts the beacon messages and exchanges with its neighbor vehicles.

We design a Cached Neighbor List for each vehicle to parse the received beacon messages and store the information of the neighbor vehicles it meets during driving. The Cached Neighbor List is a queue, which is divided into two zones according to the time the vehicle receives the information: (1) *Current Neighbor Zone* that contains the information of the recently discovered neighbor vehicles, considering these vehicles may still be its neighbors, and (2) *Historical Neighbor Zone* that contains the information of the neighbors



	ID	Interests	Location	Time_stamp	Direction
Current Neighbor Zone Information received less than t_c seconds ago	ID_1	$I_{1a} \dots I_{1k}$	(x_1, y_1)	t_1	North
	⋮	⋮	⋮	⋮	⋮
	ID_m	$I_{ma} \dots I_{mk}$	(x_m, y_m)	t_m	South
Historical Neighbor Zone Information received more than t_c but less than t_e seconds ago	⋮	⋮	⋮	⋮	⋮
	ID_n	$I_{na} \dots I_{nk}$	(x_n, y_n)	t_n	East

Figure 2 Structure of the Cached Neighbor List.

that are discovered in the past, and these vehicle are no longer its neighbors. Figure 2 shows the structure of the Cached Neighbor List, where t_c and t_e are system parameters.

Basically, each vehicle caches its neighbor information for up to t_c seconds. Each time when a vehicle receives a new beacon message, the vehicle first parses and stores the information into the Current Neighbor Zone of its Cached Neighbor List. Then, the vehicle moves all the information that was received t_c seconds ago into the Historical Neighbor Zone. After this, it deletes the information that was received t_e seconds ago.

3.2 Interest-based Information Query

When a source wants to obtain the information of a vehicle that has the given interest, it broadcasts query messages to nearby vehicles to check whether they have the information of the interested vehicle in their Cached Neighbor List. A vehicle replies to the source vehicle with such information if it finds any.

In detail, when the query procedure starts, the source vehicle first specifies the interest it needs, and encloses the interest information into the head of the query message. After that, the source sets the Time-to-Live (TTL) of the query message and broadcasts to its neighbors. Upon receiving the query message, each vehicle searches its Cached Neighbor List to see whether it contains the information of a vehicle with the given interest. The vehicle first checks the Current Neighbor Zone. If it finds an entry whose *Interests* field contains the given interest, it collects the vehicle ID from the *ID* field of this entry, and then simply unicasts one-hop to reach the vehicle, considering they are currently neighbors. Upon receiving the message, the vehicle with the given interest replies to the source vehicle to establish a communication route. Up on receiving the query message, if a vehicle cannot find any information containing the given interest in the Current Neighbor Zone, the vehicle further checks its Historical Neighbor Zone. If the Historical Neighbor Zone contains such entries, the vehicle replies to the source with the entries it finds. After completing the search in its Cached Neighbor List, the vehicle reduces the TTL value of the message by one and further propagates the query message to its neighbors. A query message is dropped if its TTL value reaches zero.

After the source vehicle sends out the query messages, it sets a timer waiting for the reply. Upon timer firing, if the source cannot receive any reply, it resets the timer, increases the initial TTL value of the message by two, and rebroadcasts the message.

The detailed procedure of the Interest-based Information Query is illustrated in Algorithm 1.

3.3 Location Estimation

After receiving all the reply messages, the source vehicle parses the information from the messages, based on which, it obtains the ID and calculates an estimated current location of the interested vehicle. More specifically, from each reply message, the source obtains the ID of the vehicle that owns the given interest, and the location, driving speed and driving direction of this vehicle at the time the vehicle meets the sender of the reply message (i.e., value of *Location*, *Speed*, and *Direction* field at the time in *Time_stamp* field). Since vehicles are mobile, the current location of the interested vehicle is not the same as that in the reply message, and the speed of the vehicle may also vary over the time. To calculate the current position of the vehicle, we first estimate the minimum speed $v_{\min}$ and maximum speed $v_{\max}$ of the vehicle. Then we compute a region, a collection of all the locations the vehicle can reach under a speed within $[v_{\min}, v_{\max}]$. We further refine the calculated region by applying vehicle's driving direction and aggregating information from multiple reply messages. The refined region is the estimated current position of the vehicle.

3.3.1 Speed Estimation

To calculate the speed range of the interested vehicle, we adopt the widely applied Gaussian Distribution to model vehicle speed [34]. The speed v of a

Algorithm 1 The Interest-based Information Query

```
Notations:
    Interest_given: the given interest specified by the source node.
    i.X: the field X of entry i in a Cached Neighbor List.
    Qualified_List: a list of entries that contain information of qualified vehicles.

    At the source vehicle
1:  Set the TTL value;
2:  Set a timer;
3:  Broadcast the query message;
    Upon the timer fires
4:  if receive no reply then
5:      TTL=TTL+2;
6:      GOTO STEP 2
7:  else
8:      Exit;
9:  end if
    At the vehicle whose interests contains the given interest
    Upon receiving the query message
10: Send reply to the source node;
    At the vehicle whose interests does not contains the given interest
    Upon receiving the query message
11: for i in Current Neighbor Zone do
12:     if Interest_given is in i.Interest then
13:         Forward the query message to vehicle with i.ID;
14:         Exit;
15:     end if
16: end for
17: for i in Historical Neighbor Zone do
18:     if Interest_given $\in$ i.Interest then
19:         Add entry i to Qualified_List;
20:     end if
21: end for
22: if Qualified_List != $\emptyset$ then
23:     Reply Qualified_List to source vehicle;
24: end if
25: if $--$TTL!=0 then
26:     Broadcast the query message;
27: end if
```

Table 1 Confidence interval under different probabilities.

Probability (p)	**Confidence Interval/Speed Range**
0.8	$\mu \pm 1.281\sigma$
0.9	$\mu \pm 1.645\sigma$
0.95	$\mu \pm 1.960\sigma$
0.98	$\mu \pm 2.326\sigma$
0.99	$\mu \pm 2.575\sigma$
0.995	$\mu \pm 2.807\sigma$

vehicle can be modeled as follows:

$$v \sim N(\mu, \sigma^2), \tag{1}$$

where μ is the mean and σ is the standard deviation. Given an area, the mean and standard deviation of the vehicle speed in the area can be obtained through various traffic statistical services, such as Google Maps [2] and Microsoft Bing Maps [3].

Given the vehicle speed in an area modeled with Gaussian distribution shown as in Eq. (1), we want to compute a speed range $[v_{\min}, v_{\max}]$, which is a confidence interval, such that the speed of a vehicle has a high probability (e.g., 0.9) to fall in this interval. We use quantile function to calculate the confidence interval of the vehicle speed under a given probability. The quantile function of a distribution is the inverse of the cumulative distribution function. For a Gaussian distribution with mean μ and standard deviation σ, given a probability p, the quantile function of the distribution can be written as:

$$F^{-1}p = \mu + \sigma\sqrt{2} \cdot \mathrm{erf}^{-1}(2p - 1), \tag{2}$$

where *erf* is the error function shown as follows:

$$\mathrm{erf}(x) = 2/\sqrt{\pi} \cdot \int_0^x e^{-t^2} dt. \tag{3}$$

Table 1 shows the calculated confidence interval (speed range) under different probabilities.

For an interested vehicle, we specify a high probability value (e.g., 0.9 or 0.95), and then calculate a speed range of this vehicle to ensure the actual speed of the vehicle has a high probability to fall in the calculated range. For example, in an area, when the mean of the vehicle speed is 30 m/s and the standard deviation is 5m/s , given the probably of 0.9, the corrsponding speed range of a vehicle in the area is $30 \pm 1.645 \times 5 = [21.775$ m/s, 38.225 m/s].

3.3.2 Calculation of the Possible Locations

We compute the speed range of the interested vehicle. In addition, from each received reply message, we can obtain the time the interested vehicle meets the reply sender (value in *Time-stamp* field) and the location at the time it meets the reply sender (value in *Location* field). By comparing the time recorded in the reply message and the current time, we can further calculate the time duration the vehicle travels after meeting with the reply sender. Combining all such obtained information, we can estimate the current region of the interested vehicle, which is a collection of all the possible locations this vehicle can reach after meeting with the reply sender, under the calculated speed range.

In detail, when the source vehicle receives a reply message from the reply sender containing the information of an interested vehicle, it first computes the speed range $[v_{\min}, v_{\max}]$ of the vehicle using the method introduced in Section 3.3.1. Then, it calculates the traveling time t_{tr} of the vehicle by computing the time difference between the value in $Time\_stamp$ field of the reply message and the current time. t_{tr} is the time the vehicle travels after exchanging a beacon message with the reply sender. After that, the traveling distance d_{tr} of the vehicle can be computed by multiplying the speed of the vehicle and the calculated traveling time, which is the range shown as follows:

$$[v_{\min} \times t_{tr}, v_{\max} \times t_{tr}]. \tag{4}$$

Thus, the possible current location of the interested vehicle is a region, a collection of locations whose distance to the location the vehicle meets the reply sender falls in the range shown in Eq. (4). For example, given the location where the interested vehicle meets the reply sender as shown in Figure 3, the estimated region is the area of a circle with the meeting location as the origin and $v_{\max} \times t_{tr}$ as radius subtracting the area of a circle with the meeting location as the origin and $v_{\min} \times t_{tr}$ as radius.

As vehicles are driving rationally, a vehicle may always drive on a road segment and does not go to the off-road area, and it may always try to drive in the direction towards the destination and will not drive back and forth through it can do so. For the interested vehicle, after departing from the location it meets the reply sender, it is very likely that the vehicle will keep driving in the direction same as the one when it meets the reply sender, i.e., the direction recorded in the *Direction* field of the reply message. Thus, the estimated region of the interested vehicle can be further refined, which is the collection of subregions that meet the following two conditions:

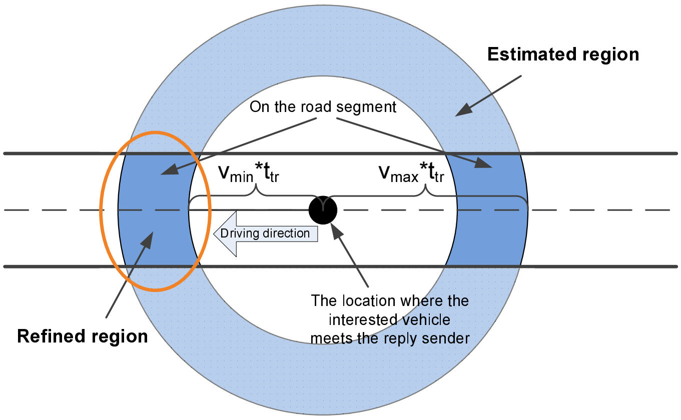


Figure 3 Region of the interested vehicle.

1. on a road segment;
2. in the driving direction (i.g., *Direction* field of the reply message) of the location where the interested vehicle meets the reply sender (e.g., *Location* field in the reply message).

For example, in Figure 3, since the vehicle drives to the left of the road, the area on the left-hand side of the figure is the refined region the interested vehicle.

As a vehicle driving in an area meets multiple vehicles, it is very likely that the information of the interested vehicle is cached by multiple vehicles, and the source vehicle may receive multiple reply messages containing the information of the same interested vehicle. We aggregate the information from multiple reply messages to provide more accurate location estimation for the interested vehicle. When the source receives multiple replies containing the information of the same vehicle, for each received message, the source applies the aforementioned method to calculate a region of possible locations for the vehicle. The aggregated region is the one that intersects all the calculated regions, considering the vehicle has a very high probability to present in this region. For example, a source vehicle receives three reply messages for the same vehicle. After applying the location estimation method, it calculates three sets of subregions $\{R_a, R_b, R_c, R_d\}$, $\{R_b, R_c, R_d\}$, and $\{R_a, R_b, R_c\}$. Thus, the final aggregated region of the vehicle is $\{R_b, R_c\}$.

It is also possible that the source vehicle will receive the information of multiple vehicles, each having the given interest. In this case, the source calculates the region for all these vehicles and picks the vehicle whose region is closest to the current location of the source as the interested vehicle.

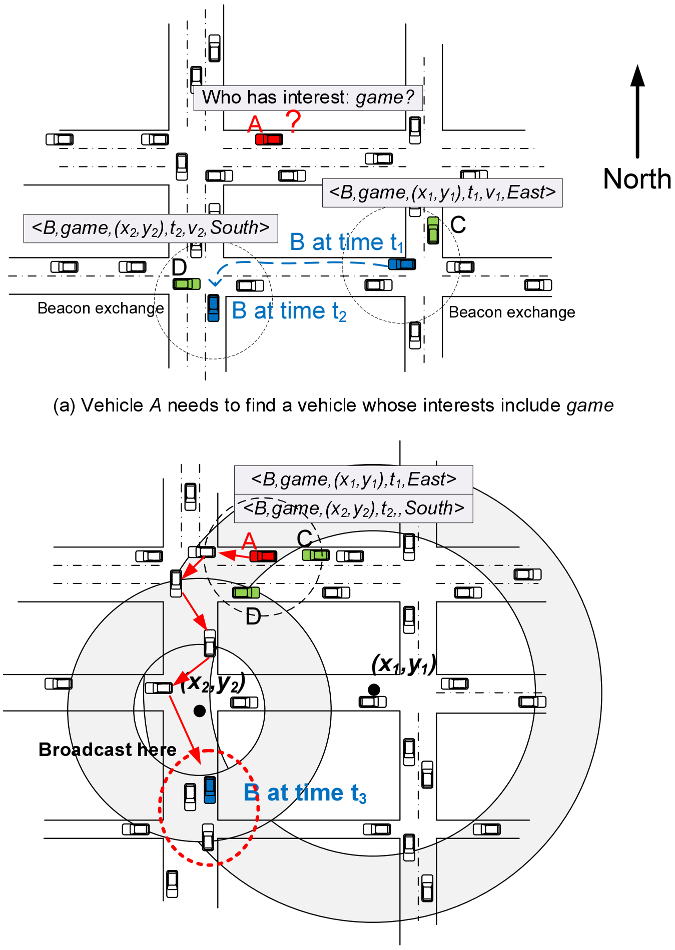
3.4 Data Delivery

After *Interest-based Information Query* and *Location Estimation*, we obtain the ID and the estimated region of the vehicle that has the given interest. The estimated region is not the exact current location of the interested vehicle, but a set of subregions containing all the locations where the vehicle currently has a high probability to present. To deliver the data to the interested vehicle, we build a route from the source vehicle towards the calculated region and broadcast in the region to reach the interested vehicle. The routing and broadcasting can be carried out by various location-based data dissemination mechanisms, such as Geocast [11]. If the source vehicle cannot reach the interested vehicle after broadcasting, it doubles the size of the region and broadcasts the message to the expended region. If the source still cannot reach the destination, it cancels the broadcasting and starts sending the query messages to redo the whole data delivery procedure.

In Section 3.5 we will show an example of the whole procedure of the *Ivy* protocol.

3.5 An Example of the *Ivy* Protocol

Suppose the source vehicle A wants to find a vehicle whose interests includes *game* and deliver data to this vehicle as shown in Figure 4(a). Suppose vehicle B is a vehicle that has interest *game*. During driving, at time t_1, vehicle B meets vehicle C. Through beacon message exchange, C caches the information of B into its Cached Neighbor List. At time t_2, B meets another vehicle, D. Through beacon message exchange, D caches the information of B into its Cached Neighbor List. At time t_3, vehicle A broadcasts query message as shown in Figure 4(b). Upon receiving the query message, vehicle C and D reply to A with their cached information of vehicle B. Upon receiving reply messages, vehicle A obtains the ID of vehicle B and calculates an estimated region for each received message shown as the circle areas in Figure 4(b). According to the driving direction of vehicle B recorded in reply messages, vehicle A only considers the left half area of the circle with origin (x_1, y_1) and the lower half area of the circle with origin (x_2, y_2). After that, it computes the



(a) Vehicle *A* needs to find a vehicle whose interests include *game*

(b) At time t_3 (current time), vehicle *A* broadcasts query message and calculates the location of the interested vehicle, vehicle *B*

Figure 4 An example of the *Ivy* protocol.

aggregated region shown as the dashed oval in Figure 4(b), and then routes the data and broadcast it in the region.

4 *Ivy-LR*: Ivy with Location Refinement

As vehicles are of high mobility, after the source vehicle routes the data to the estimated region, it is possible that the interested vehicle may have already moved out of the region, which makes it fail to be reached by the source vehicle.

To mitigate this issue, in this section, we design an enhanced *Ivy* with location refinement (*Ivy-LR*) which dynamically adjusts the estimated region and the forwarding direction by keeping querying the information of the interested vehicle during data forwarding to receive more up-to-date information. The design of *Ivy-LR* is inspired by the following observation. For the information (e.g., Location and Driving direction) of a vehicle cached by another vehicle, the more recently the two vehicles meet, the more up-to-date the cached information is, and the more accurate the estimated region will be. Thus, the estimated region calculated using the information cached in vehicles that are close to the current location of the interested vehicle is more accurate than that using the information cached in vehicles far from that location. Therefore, when forwarding messages from the source vehicle to the estimated region, if we keep broadcasting queries in each hop we move closer to the region, we will receive more and more up-to-date locations and driving directions of the interested vehicle, which helps us update the estimated region and dynamically adjust the forwarding direction to reach the interested vehicle.

In detail, after the source vehicle calculates the region for the interested vehicle, it routes data to a vehicle closer to the calculated region. Upon receiving the message, the vehicle first broadcasts a query message to its neighbors asking for the information of the interested vehicle. Upon receiving the query message, the neighbor vehicle replies to the query sender if it has such information. Upon collecting the information from its neighbors, the vehicle computes an estimated region using the collected information, aggregates this region with the region computed by the source, and generates a refined region that is the area intersecting both regions. After that, the vehicle picks its neighbor closest to the newly generated region and forwards the message to it. The whole procedure terminates if the message reaches the interested vehicle.

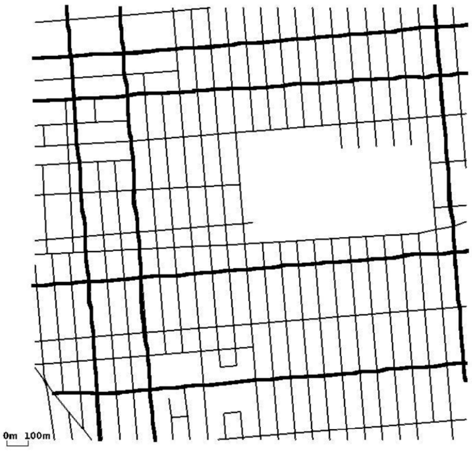


Figure 5 Simulation map.

5 Performance Evaluation

5.1 Evaluation Methodology

Ivy is evaluated through simulation implemented in the NS-2 simulator. The objective of the evaluation is three-fold:

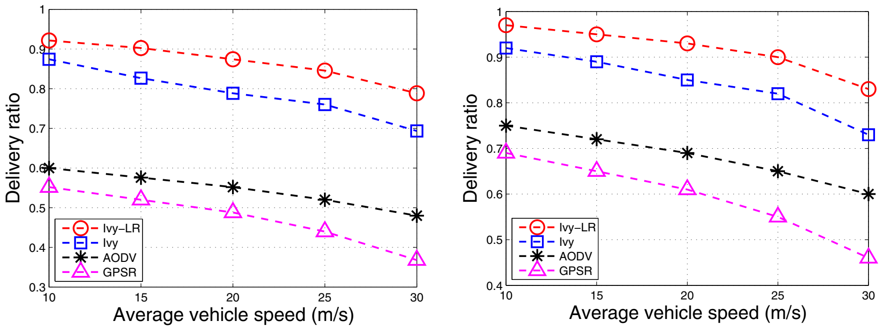
1. Test the performance of *Ivy* in reaching a vehicle that has given interest.
2. Evaluate the efficiency of *Ivy* to reach the vehicle with given interest.
3. Test the effectiveness of location refinement in *Ivy-LR*.

We employ three metrics to do the evaluation. We choose the *delivery ratio* and *delivery delay* as measures of the performance and choose and *message overhead* as measures of cost. We compare our scheme with AODV [26] and GPSR [17], two widely used routing protocols in VANET and Mobile Ad Hoc Networks (MANETs). The original AODV and GPSR do not include interest information in their beacon messages. To have a fair comparison, we extend AODV and GPSR by adding interest information into the beacon message of each vehicle. In a network, there may be multiple vehicles that have the given interest, the source vehicle is considered to successfully deliver the data if it can reach any of the interested vehicle.

In our simulation, the map is a 2000 m × 2000 m area centered at (Latitude: 40.6548, longitude: –73.942795) in Brooklyn, NYC, extracted from the TIGER/Line (Topologically Intergraded Geographic Encoding and Referencing) database of the US Census Bureau [4]. Figure 5 shows the

Table 2 Simulation setting.

Parameter	**Value**
Size of the simulation area	2000 m × 2000 m
Simulation time	5000 s
Number of vehicles	200
Number of vehicles with given interest	5, 15
Vehicle communication range:	200 m
Vehicle speed	8 ∼ 15 m/s



(a) Number of interested vehicles = 5

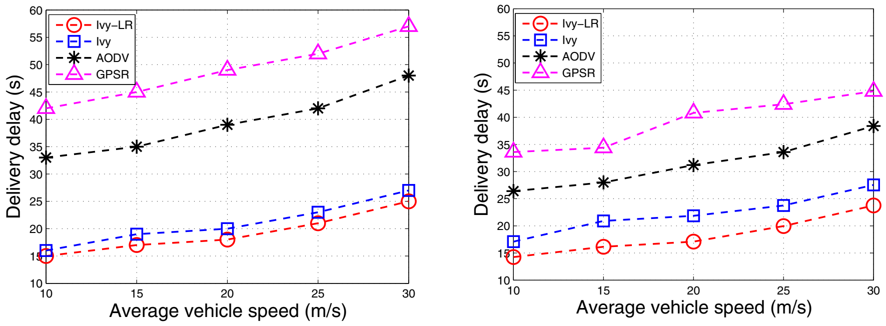
(b) Number of interested vehicles = 15

Figure 6 Data delivery ratio.

map we use. For the moving trace of vehicles, we employ the open-source, microscopic, space-continuous and time discrete vehicular traffic generator package SUMO [5] to generate the movements of vehicle nodes. SUMO uses a collision-free car-following model to determine the speeds and positions of the vehicles. After intergrading the SUMO trace into the map of TIGER, we discard the first 2000 seconds to obtain more accurate node movements. The output from SUMO is converted into input files for the movement of node in the NS-2 simulator. We configure the simulation according to the WAVE protocol [6]. Other system parameters are listed in Table 2.

5.2 Data Delivery Ratio

Figure 6 shows the data delivery ratio under different average vehicle speed. From the figure we can see that the delivery ratio of *Ivy-LR* is higher than that of *Ivy*, and they both outperform the compared schemes. When the average vehicle speed increases, the delivery ratio of all the schemes decreases. When



(a) Number of interested vehicles = 5 (b) Number of interested vehicles = 15

Figure 7 Delivery delay.

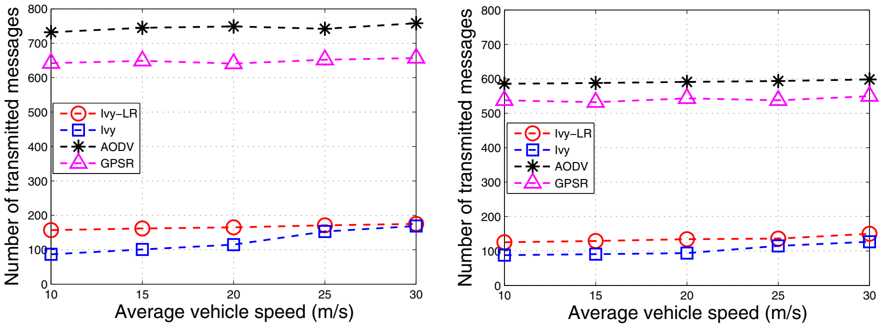
the number of interested vehicles in the network increase, the delivery ratio increases, which is because when the network has more vehicles with given interest, it is easier to identify and reach one out of them.

5.3 Data Delivery Delay

Figure 7 shows the data delivery delay under different average vehicle speed. From the figure we can see that the delivery delay of *Ivy-LR* is lower than that of *Ivy*, because *Ivy-LR* dynamically adjusts the estimated region to reach the destination, which improves the effectiveness of the data forwarding. Both of our schemes have lower delay than the compared schemes. When the average vehicle speed increases, the delivery delay of all the schemes increase because the forwarding link in the network may break more frequently, which increases the time in finding alternative routes. When the number of interested vehicles in the network increases, the delivery delay decreases, because the density of the interested vehicles increases, which reduces the routing length between the source vehicle and an interested vehicle.

5.4 Message Overhead

Figure 8 shows the number of transmitted messages under different average vehicle speed. From the figure we can see that the message overhead of *Ivy-LR* is higher than that of *Ivy*, because *Ivy-LR* needs to keep querying the information of the interested vehicle when forwarding data. Both *Ivy* and



(a) Number of interested vehicles = 5

(b) Number of interested vehicles = 15

Figure 8 Message overhead.

Ivy-LR have less message overhead than the compared schemes, because our schemes only forward data to the direction towards the estimated region, while AODV and GPSR need to broadcast to all the directions to build a route to the destination. When the average vehicle speed increases, the message overhead of *Ivy* increases, while the overhead of the other three schemes does not increase significantly. In *Ivy*, if the speed of the interested vehicle is high, when the source routes the data to the estimated region, it is very likely that the vehicle has already moved out of the region. The source has to broadcast to a bigger region to search for the interested vehicle, which increases the message overhead. For *Ivy-LR*, since it keeps updating the estimated region, the message can always be forwarded to the location where the interested vehicle currently is. Since AODV and GPSR does not estimate the destination location, their message overhead does not change much when the vehicle speed changes.

6 Related Work

Data delivery has been extensively studied in both Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs). Node centric routing protocols, such as AODV [26], DSDV [27], DSR [16] and OLSR [9], establish topological end-to-end paths for the source to reach the destination. Their performance is further improved in [10, 24, 29] by exploiting the knowledge of vehicles' relative velocities and their constrained movements. Geographical routing protocols, such as GPSR [17], GOAFR [18] and GFG [7], leverage node positions to forward messages. To improve the greedy forwarding according to node's position, recovery mechanisms are provided in [22, 8], which proactively detect the potential dead-end positions and overhears the communication channels to decrease the number of hops on the recovery paths. GSR [21] and SAR [30] adapts the concepts of anchor-based routing that was originally designed for sensor networks to vehicular network environments, in which a source vehicle uses a list of intersections to identify road segments and computes the shortest road-based path from its current position to the destination. A-STAR [50] improves GSR by giving preference to streets served by transit buses each time a new intersection is to be added to the source route. CAR [25] and GyTar [15] dynamically find connected paths between source and destination pairs by considering vehicular traffic and road topology. Opportunistic data forwarding mechanisms are designed in MDDV [31] and VADD [35]. VADD analyzes the historical traffic flow of an area to determine the best route to the destination. In addition to traffics, MDDV considers the number of lanes on each road segment to select a suitable road-based trajectory to forward data. In both protocols, when an intermediate vehicle temporarily cannot reach another vehicle to forward data, a carry-and-forward approach is used, which caches the data packets until a suitable relay is found. Huang et al. [14] presents a delay tolerant epidemic routing protocol, which improves the delivery ratio in the scenario with very sparse vehicle traffic.

The aforementioned schemes together improves the performance and efficiency for delivering data to a vehicle with known ID or in a known location. However, none of them considers reaching a vehicle that has given interest without knowing its ID or location beforehand. Without a delegated mechanism to efficiently obtain the ID and calculate the location of the interested vehicle, these mechanisms cannot be applied to achieve the goal of the paper.

7 Conclusion

This paper proposes *Ivy*, an interest-based data delivery scheme to deliver data to a vehicle that has a given interest. We increase the efficiency in querying and identifying the vehicle of interest by letting each vehicle in the network cache the neighbor vehicles it meets during driving, such that by querying only a few vehicles, we can retrieve the information of a large amount of vehicles and identify the interested vehicle out of it. We reduce the message overhead in route establishment by estimating the current location of the interested vehicle, and route message towards the estimated location. Simulation result shows *Ivy* is both efficient and effective, and outperforms AODV and GPSR with higher delivery ratio, lower delay and smaller message overhead.

References

[1] `http://grouper.ieee.org/groups/scc32/dsrc/index.html`. 5.9 GHz Dedicated Short Range Communications (DSRC).

[2] `http://support.google.com/maps/bin/answer.py?hl=en&answer=61454`. Traffic in Google Maps.

[3] `http://msdn.microsoft.com/en-us/library/jj136866.aspx`. Traffic in Bing Maps.

[4] `http://www.census.gov/geo/www/tiger/`. U.S. Census Bureau TIGER/Line 2009.

[5] `http://sumo.sourceforge.net/`. Centre for Applied Informatics (ZAIK) and the Institute of Transport Research German Aerospace Centre, SUMO – Simulation of Urban Mobility.

[6] IEEE 802.11p wireless access in vehicular environments (WAVE). In Proceedings IEEE 1609-Family of Standards for Wireless Access in Vehicular Environments (WAVE). U.S. Department of Transportation. January 9, 2006. Retrieved 2007-07-15.

[7] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. ACM Wireless Networks, 7(6):609–616, November 2001.

[8] C.-H. Chou, K.-F. Ssu, and H. C. Jiau. Geographic forwarding with dead-end reduction in mobile ad hoc networks. IEEE Transactions on Vehicular Technology, 57(4):2375–2386, July 2008.

[9] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). Internet Engineering Task Force, 2003.

[10] K.-T. Feng, C.-H. Hsu, and T.-E. Lu. Velocity-assisted predictive mobility and location-aware routing protocols for mobile ad hoc networks. IEEE Transactions on Vehicular Technology, 57(1):448–464, 2008.

[11] R. Hall. An improved geocast for mobile ad hoc networks. IEEE Transactions on Mobile Computing, 10(2): February 2011.

[12] Sanqing Hu, Yu-Dong Yao, and A.U. Sheikh. Slotted aloha for cognitive radio users and its tagged user analysis. In Proceedings Wireless and Optical Communications Conference (WOCC), 2012.

[13] Sanqing Hu, Yu-Dong Yao, A.U. Sheikh, and M.A. Haleem. Tagged user approach for finite-user finite-buffer S-Aloha analysis in AWGN and frequency selective fading channels. In Proceedings IEEE 34th Sarnoff Symposium, 2011.

[14] H.-Y. Huang, P.-E. Luo, M. Li, D. Li, X. Li, W. Shu, and M.-Y. Wu. Performance evaluation of suvnet with real-time traffic data. IEEE Transactions on Vehicular Technology, 56(6):3381–3396, 2007.

[15] M. Jerbi, R. Meraihi, S.-M. Senouci, and Y. Ghamri-Doudane. Gytar: Improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments. In Pro-

ceedings 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET), 2006.

[16] D. Johnson, D. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In Proceedings Ad Hoc Networking, 2001.

[17] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In Proceedings Mobicom, 2003.

[18] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: Of theory and practice. In Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing, Boston, MA, July 2003, pages 63–72.

[19] S-B. Lee, G. Pan, J-S. Park, M. Gerla, and S. Lu. Secure incentives for commercial ad dissemination in vehicular networks. In Proceedings Mobihoc 2007.

[20] I. Leontiadis and C. Mascolo. Geopps: Geographical opportunistic routing for vehicular networks. In Proceedings IEEE Workshop on Autonomic and Opportunistic Communication, 2007.

[21] C. Lochert, H. Hartenstein, J. Tian, H. Fubler, D. Hermann, and M. Mauve. A routing strategy for vehicular ad hoc networks in city environments. In Proceedings IEEE Intelligent Vehicles Symposium, 2003.

[22] X. Ma, M.-T. Sun, G. Zhao, and X. Liu. An efficient path pruning algorithm for geographical routing in wireless networks. IEEE Transactions on Vehicular Technology, 57(4):2474–2488, July 2008.

[23] Y. Mylonase, M. Lestas, and A. Pitsillides. Speed adaptive probabilistic flooding in cooperative emergency warning. In Proceedings of International Conference on Wireless Internet, 2008.

[24] V. Namboodiri and L. Gao. Prediction-based routing for vehicular ad hoc networks. IEEE Transactions on Vehicular Technology, 56(4):2332–2345, 2007.

[25] V. Naumov and T. R. Gross. Connectivity-aware routing (CAR) in vehicular ad hoc networks. In Proceedings IEEE INFOCOM 2007.

[26] C. Perkins, E. Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. Internet Engineering Task Force, 2003.

[27] E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In Proceedings Sigcomm, 1994.

[28] Mudhakar Srivatsa, Ling Liu, and Arun Iyengar. Eventguard: A system architecture for securing publish-subscribe networks. ACM Transactions on Computer Systems (TOCS), 29(4):10, 2011.

[29] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto. A stable routing protocol to support its services in VANET networks. IEEE Transactions on Vehicular Technology, 56(6):3337–3347, 2007.

[30] J. Tian, L. Han, K. Rothermel, and C. Cseh. Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks. In Proceedings IEEE Intelligent Transportation Systems, 2003.

[31] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: A mobility-centric data dissemination algorithm for vehicular networks. In Proceedings 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET), 2004.

[32] T. Yan, W. Zhang, and G. Wang. DOVE: Data dissemination to a fixed number of receivers in VANET. In Proceedings IEEE International Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012.

[33] T. Yan, W. Zhang, G. Wang, and Y. Zhang. GOT: Grid-based on-road localization through inter-vehicle collaboration. In Proceedings IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), 2011.

[34] Saleh Youse, Eitan Altman, Rachid El-Azouzi, and Mahmood Fathy. Analytical model for connectivity in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology, 57(6), 2008.

[35] J. Zhao and G. Cao. VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology, 57(3), May 2008.

Biographies

Tan Yan is a PhD candidate in Computer Science Department in New Jersey Institute of Technology, New Jersey, USA. He received his M.E in Electrical Engineering from New Jersey Institute of Technology in 2008, and B.E. in Electrical Engineering from Southeast University, Nanjing, China in 2007.

Guiling Wang (Grace) joined NJIT in fall 2006 and was promoted to Associate Professor with tenure in June 2011. She received her Ph.D. in Computer Science and Engineering and a minor in Statistics from The Pennsylvania State University in May 2006. She received her B.S. in Software from Nankai University in Tianjin, China.