

# GPSFR: GPS-Free Geographic Routing Protocol for Intelligent Vehicles

Qing Yang, Alvin Lim, *Member, IEEE* and Prathima Agrawal, *Fellow, IEEE*

(Invited Paper)

**Abstract**—Intelligent vehicles can improve safety by communicating critical road hazard and traffic information among vehicles in the roadway. In this paper, we propose a novel routing protocol called *GPS-Free Geographic Routing (GPSFR)* for inter-vehicle communication. It does not require position information (e.g. from GPS) but instead rely on relative position that can be determined dynamically. The proposed method focuses primarily on the complexity of rural highways and solves problems that arise when vehicles are near interchanges, curves, and merge or exit lanes of highways. GPSFR greedily chooses the best next hop neighbor based on a *Balance Advance (BADV)* metric that balances between proximity and link stability due to relative velocity. Our simulation results show that by taking relative velocity into account, GPSFR reduces link breakage frequency to only 27% that of GPSR in the best case, and 70% in the worst case. Furthermore, the average path length of GPSFR is similar to that of GPSR about 80% of the time.

**Index Terms**—Geographic routing, GPS-Free, Intelligent vehicles, Relative position, Vehicular networks

## I. INTRODUCTION

IN the past few years, there has been an increasing interest in the research on inter-vehicle communication protocol for vehicular ad-hoc network [1, 2, 3, 7]. In fact, the FCC has already allocated 75MHz of spectrum for vehicle-vehicle and vehicle-roadside communication, called dedicated short range communications (DSRC). IEEE is also working on a standard for inter-vehicle communication [1]. In the future, large scale vehicular ad-hoc networks will be available to provide drivers with higher level of safety and convenience.

To enable this vision, several researches have been initiated. For example, the medium access control (MAC) problem was addressed in [4, 5], where directional [5] and omni-directional MAC protocol [4] are studied. In [6], a multi-channel MAC protocol was proposed to improve safety and commercial services. The routing issues were addressed in [1, 2, 7], where

position-base routing [7], mobility-centric data dissemination [2] and vehicle-assisted trajectory-based routing protocol [1] were studied. In addition, an application layer communication protocol VITP [3] was also recently investigated.

However, all the aforementioned routing protocols rely on the positions of nodes, which require each vehicle to be equipped with a GPS receiver. Typically, GPS receiver will use a different spectrum from DSRC. Thus, an extra antenna is necessary in each car to enable communication between vehicles. However, in this paper, we present a GPS-free geographic routing protocol by using only one directional antenna with two beams. Hence, the cost of adding a directional antenna will involve comparable cost. Although the GPS receiver and wireless transceiver may become standard components of vehicles in the future, some older cars may not have them. Our work attempts to provide a cheaper alternative for all types of vehicles and enables inter-vehicle communication even for those older ones.

The rest of this paper is organized as following. In Section II, we summarize some related work. Then, we discuss the motivation and assumption of this paper in Section III. The Relative Position Maintenance (*RPM*) and routing algorithms are described in Section IV. We set up the simulation environment and present the simulation results in Section V. Section VI gives the conclusion and future work.

## II. RELATED WORK

For a certain vehicular networking, there will be three potential scenarios: 1) all the vehicles are equipped with GPS receivers, so they can determine their position whenever four or more GPS satellites are available, 2) some vehicles have the GPS devices whereas others do not, and 3) no vehicle has a GPS device or all of them suffer from persistent GPS outage.

Many routing protocols have been developed for the first scenario [1, 2, 7], as discussed in Section I. The second scenario may exploit many approaches to find the location of nodes, such as Receiving Signal Strength Indicator (RSSI), Angle of Arrival (AOA), Time of Arrival (TOA), and Time Difference of Arrival (TDOA). All of these methods require nodes to have the ability to measure the distance to their neighbors. Therefore, the effectiveness of these approaches will rely heavily on the accuracy of distance measurement. They must also solve errors caused by absence of line of sight, where

Q. Yang is with the Computer Science and Software Engineering Department, Auburn University, AL 36849 USA (phone: 334-844-6324; e-mail: yangqin@auburn.edu).

A. Lim is with the Computer Science and Software Engineering Department, Auburn University, AL 36849 USA (email: limalvi@auburn.edu).

P. Agrawal is now with the Department of Electrical and Computer Engineering, Auburn University, AL 36849 USA (email: pagrawal@eng.auburn.edu).

additional hardware or software is required for signal processing. As a result, it will increase the cost of intelligent vehicles. On the other hand, absence of anchor nodes will cause a fatal failure of routing protocol.

For the last scenario, many research results exist [8, 9, 10]. A distributed, infrastructure free positioning algorithm for mobile ad hoc network has been proposed in [8] that do not rely on the GPS. However, it is not suitable for vehicular networks for two main reasons. First, after determining the relative positions of neighbors, each node has to change their local coordinate system to the network coordinate system. Thus, such updating overhead will become larger as the network size increases. In fact, it has been proved in [9] that the volume of message exchanged in [8] increases exponentially with the node density. Secondly, in high mobility vehicular networks, the overhead of updating location reference group, which is composed by nodes with lower moving speed, will be very significant. Although cluster-based method in [9] can generate less communication overheads compared to [8], the number of message exchange is still very high because the coordinate translation of all master nodes will occur throughout the entire network.

Unlike the above-mentioned approaches, [10] proposed a method which do not require any signal processing mode but can also localize nodes through message exchanges alone. Besides the GPS-free positioning mechanism, another important contribution of [10] is that they introduce the concept of geographic routing method using only virtual coordinate system instead of the absolute position of nodes. Inspired by this paper, we propose the GPSFR protocol where routing is based only on the relative positions of nodes. Method in [10] will be very suitable for ad-hoc network with less mobility, such as sensor networks. However, their method must be carefully modified before applying it to the vehicular networks. High mobility in vehicular networks will result in large network overhead using the method in [10] due to periodically bootstrapping beacons. According to [10], the number of flooding nodes during the bootstrapping phase will increase as network size increases. Therefore, this method may not be suitable for vehicular networks. The similar concept of relative position has also been proposed in [11], but it was only used for cooperative collision warning. Instead, GPSFR focuses on how to use the relative position of neighbors to optimize the routing task in vehicular networks.

In summary, the existing schemes for GPS-free positioning and routing either incur too much network overhead or require additional signal processing cost or suffer from both of them. Therefore, they are not suitable for vehicular networks. Unlike the aforementioned work, using directional antenna and local periodical beaconing, each node can determine the relative positions of nodes in its vicinity.

### III. MOTIVATIONS AND ASSUMPTIONS

The recent report [12] by the National Highway Traffic Safety Administration (NHTSA), shows that in U.S. alone, vehicle crashes on highways resulted in the loss of as many as

40,000 lives and an overall economic losses of more than \$230 billion. Therefore, in this paper, our goal is to design and implement an inter-vehicle routing protocol for intelligent vehicles in highways to exchange information to improve safety. For instance, by taking advantage of multi-hop communication between vehicles, the ACC (adaptive cruise control) systems can be enabled early to adapt longitudinal control in response to traffic incidents a few hops in front of it. It can also early anticipate braking maneuvers when an "invisible" vehicle further in front is braking.

However, the routing problem in vehicular networks is hard. Currently, we focus primarily on vehicular networks in rural highways. A rural highway provides a link between urban areas, e.g. the Inter-state 85 in U.S. The reason to choose rural highway is that most part of the inter-state highway system in U.S. is rural highway. Another benefit of rural highway system is that vehicles moving on the highway form a linear network, so the routing decision can be easily divided into those in forward or reverse directions. To build up and maintain the neighbors' relative position, each node needs a compass and a directional antenna with two beams. One beam is for sending/receiving data to neighbors in front, and the other for those behind. The interested reader can find an intensive study of directional antennas in [13]. Benefits of directional antennas are: 1) longer radio range, 2) solution to the *exposed station* problem, and 3) reduction of co-channel interference. We also assume that the coverage area of each antenna element is a semi-circle, thus the area covered by both should be a complete circle. Since the direction of the vehicle can be obtained from a compass, we can determine if a vehicle is turning or veering by checking the change in the direction.

### IV. ROUTING PROTOCOL

We now describe the GPS-Free Geographic Routing (GPSFR) protocol. This protocol consists of two parts: *relative position maintenance*, which is used to compute and maintain the relative positions of nodes without GPS support; and *routing algorithm*, which will greedily transmit packets based on the link metric called *balanced advance (BADV)*. The most difference between GPSFR and GPSR [17] is that 1) GPSFR does not need to know the absolute locations of nodes, and 2) GPSFR takes the velocities into account when routing packets.

#### A. Relative Position Maintenance

When moving at the same direction on highways, all vehicles will compose a linear network, as shown in Fig. 1. While problems involving vehicles at interchanges or ramps will be discussed later, for now we only focus on linear networking. In such networks, packets delivery can be categorized as *forward* or *backward*. That means, on each intermediate node, the routing algorithm only needs to choose the next hop from neighbors that are in the same direction as packets being delivered. In order to achieve this, each node has to compute and maintain the relative positions of all its neighbors. As shown in Fig. 1 (a), suppose all nodes from 1 to 6 are neighbors, then from node 3's perspective, node 5 should have a closer

relative position than node 6. Also, the relative position of node 1 should be further than node 2's.

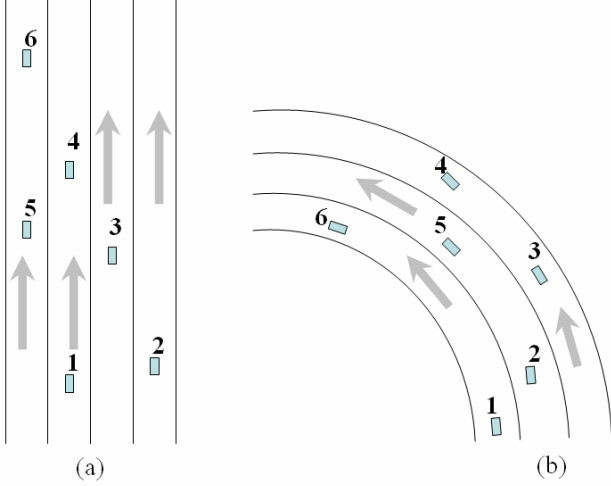


Fig. 1. Illustration of nodes' relative positions on straight and curve road.

As stated above, we assume the directional antenna consist of two antenna elements ( $f\_antenna$  and  $b\_antenna$ ). So if the message is received from one's  $f\_antenna$ , then it must be sent by a node in front of the receiver; otherwise, it must come from a rear node. Therefore, each node can divide its neighbors into two groups ( $fgroup$  and  $bgroup$ ) by checking from which antenna the messages were received. As shown in Fig. 1 (b), we note that even on a curve, vehicles can also divide neighbors into two groups (assume that curves should not be very sharp, which is reasonable on the highway systems). In addition, we can easily distinguish packets received from the opposite moving vehicles. Since if such message was sent from  $f\_antenna$  (or  $b\_antenna$ ) of one node moving at the opposite direction, then the receiver will obtain it also from its  $f\_antenna$  (or  $b\_antenna$ ). Therefore, the vehicles moving at the opposite direction will not interfere with the network in the forward direction.

As shown in Fig. 1, we note that the  $fgroup$  of node 3 should be  $\{4, 5, 6\}$  and the  $bgroup$  is  $\{1, 2\}$ . After collecting all neighbors' information, each node will beacon such group information periodically. This beacon message called  $group\_update$  has the format:  $\langle bgroup, id, velocity, direction, fgroup \rangle$ . If such a message is not received from one neighbor for longer than the interval  $T$ , we consider this neighbor to have failed or gone out-of-range and deletes it from the neighbor list. In fact, we could make GPSFR's beacon mechanism fully reactive, in which nodes will solicit beacons with a broadcast message only when they have data to transmit. However, we have not felt it necessary to take this step since the one-hop beacon overhead does not cause congestion.

Computing the relative positions of nodes in linear networks is straightforward. However, there is minor problem. Suppose vehicle A has two neighbors B and C in front, but B and C are not neighbors; then we can say B and C are the *hidden neighbors* of A. For example, as shown in Fig. 2, node 2 and 4 are the *hidden neighbors* of node 1.

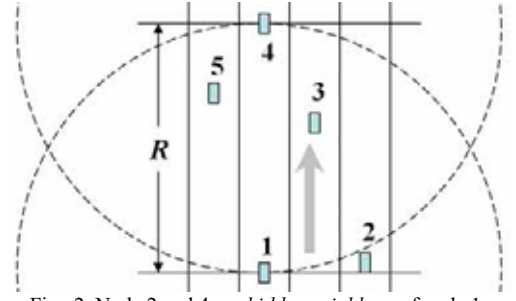


Fig. 2. Node 2 and 4 are *hidden neighbors* of node 1.

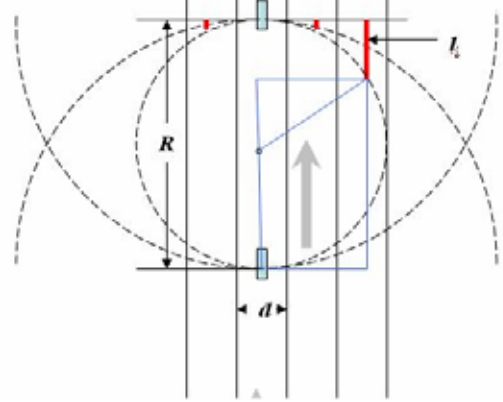


Fig. 3. There are only small pieces where *hidden neighbors* may exist.

Since the *hidden neighbors* are actually not neighbors, then it is impossible to obtain their relative positions through the directional antenna. However, the *hidden neighbor* problem occurs because the distance to *hidden neighbors* is larger than the radio range. As illustrated in Fig. 3, let the communication range of node be  $R$  and the width of each lane be  $d$ . We can find that on each lane, there is only a small area where *hidden neighbors* may exist. We denote the length of such area as  $l_i$ , calculated as follows:

$$l_i = R/2 - \sqrt{(R/2)^2 - i^2 \times d^2} \quad (i = 1, 2, \dots, m-1) \quad (1)$$

$m$  is the number of lanes which is usually from 2 to 6;  $R$  is 250 meters, which is the standard radio range in 802.11 WLAN;  $d$  is 3.6 meters, which is the typical width of highways [14]. Then the length of each piece will be very short, so the *hidden neighbor* problem will be an unlikely event in vehicular networks. Note that we also have the *hidden neighbor* problems when two vehicles are side by side, when vehicles are near an interchange or when vehicles try to leave the highway. In our protocol implementation, the hidden neighbors are not arranged in the neighbor list, since they only hidden for a short time.

#### 1) Relative Position Maintenance (RPM) Algorithm

In a certain neighborhood, except for those *hidden neighbors*, each node can determine whether or not a neighbor is in front or not through the directional antenna. So based on the message exchanges of  $group\_update$ , we derived the Relative Position Maintenance (RPM) algorithm to compute and maintain nodes' relative positions.

TABLE I

**Algorithm:** Relative Position Maintenance (*RPM*)

**Input:** Message  $m_j \langle GF_j, ID_j, V_j, D_j, GB_j \rangle$  received from  $n_j$ 
**Output:** Ordered link list  $F_i$  and  $B_i$  for current node  $n_i$ 
 $C$ : Cache for all recently received messages

 $F_i$ : Ordered link list of neighbors located in front of  $n_i$ 
 $B_i$ : Ordered link list of neighbors located behind  $n_i$ 
 $e$ : Temp variable holding the element of the ordered link list

 $GF_i$ : Neighbors located in front of  $n_i$ 
 $GB_i$ : Neighbors located behind  $n_i$ 
 $f\_antenna$ : whether message received from  $f$  antenna

 $Interval$ : Period of beaconing message

```

1. if( $m_j$  is not in  $C$ ) then
2.   if(size of  $m_j == 2 \&\& D_i == D_j$ )
3.     if( $f\_antenna$ ) then
4.       add  $ID_j$  into  $GF_i$ ;
5.     else
6.       add  $ID_j$  into  $BF_i$ ;
7.     endif
8.   exit
9. endif
10. if (size of  $m_j \geq 3 \&\& D_j$  is on clock-wise direction of  $D_i$ )
11.   drop this msg.
12. endif
13. add/update the entry of  $\langle GF_j, ID_j, V_j, D_j, GB_j \rangle$  in  $C$ 
/*  $n_j$  overtake  $n_i$  */
14. if( $ID_j$  is in  $B_i \&\& f\_antenna$ ) then
15.    $e \leftarrow$  remove element corresponding to  $ID_j$  from  $B_i$ 
16.   set  $e$  as an anchor and reset  $e$ 's life time
17.    $e.position \leftarrow 0$ ;
18. endif
/*  $n_j$  move backwards of  $n_i$  */
19. if( $ID_j$  is in  $F_i \&\& !f\_antenna$ ) then
20.    $e \leftarrow$  remove element corresponding to  $ID_j$  from  $F_i$ 
21.   set  $e$  as an anchor and reset  $e$ 's life time
22.    $e.position \leftarrow 0$ ;
23. endif
24. if( $f\_antenna$ ) then /* add  $n_j$  into the corresponding list */
25.    $Insert(F_i, GF_j, GB_j, ID_j, V_j)$ 
26. else
27.    $Insert(B_i, GF_j, GB_j, ID_j, V_j)$ 
28. endif
29. update  $GF_i$  and  $GB_i$  by new  $F_i$  and  $B_i$ 
/* received messages from all neighbors */
30. if(one  $Interval$  passed) then
31.    $RPU(V_i, F_i, true)$  and  $RPU(V_i, B_i, false)$ 
32. endif
33. else
34.   reset the life time of element related to  $ID_j$  in  $F_i$  or  $B_i$ 
35. endif
    
```

As shown in the TABLE I, each node  $n_i$  will maintains two link lists  $F_i$  and  $B_i$ .  $F_i$  is used to save the neighbors located in front;  $B_i$  is for rear neighbors. Elements of each link list are ordered by the nodes' relative positions. Besides, there will be a cache on each node, which is applied to cache the message received in the last beaconing period.

If a message is received from  $n_j$ , then node  $n_i$  will first check

whether or not this message is in the cache. If this message matches an entry, then there will be no change about  $n_j$  in the list; so we just update the lifetime of node  $n_j$  in the list. If it is a new message,  $n_i$  will firstly add/update the message in the cache and then arrange  $n_j$ 's new relative position in the list. Line 2-9 is used to deal with the scenario of new vehicles merging into the networks, which will be examined later. Line 14-28 aims to arrange node  $n_j$  in the corresponding list. We notice that, if node  $n_j$  overtook  $n_i$  during the last beaconing period, then currently the distance between those two nodes be approximately equal to 0. From  $n_i$ 's perspective,  $n_j$  will be an anchor node since its relative distance is exactly known now. In the later beacon periods, we can estimate  $n_j$ 's new relative distance from the length of beaconing period and the velocity gap between  $n_j$  and  $n_i$ . Clearly, the relative distances of anchor nodes are more accurate. To calculate others' relative distances, we propose the relative position update (*RPU*) procedure which will be discussed later.

TABLE II

INSERT A NODE INTO THE ORDERED LINK LIST AT THE PROPER POSITION

**Procedure:**  $Insert(L_i, GF_j, GB_j, ID_j, V_j)$ 
 $L_i$ : Ordered link list of node  $n_i$ 
 $GF_j$ : Group information of node  $n_j$ 's front neighbors

 $GB_j$ : Group information of node  $n_j$ 's behind neighbors

 $ID_j$ : ID of node  $n_j$ 
 $V_j$ : Velocity of node  $n_j$ 
 $added$ : bool value indicating whether  $n_j$  is added successfully

```

1.  $added \leftarrow false$ 
2. for all element  $e$  in  $L_i$  do
/*  $n_j$  and node  $e.ID$  are hidden neighbors */
3.   if( $e$  is not in  $GF_j$  or  $GB_j$ ) then
4.     exit
5.   endif
/*  $n_j$  and node  $e.ID$  are not hidden neighbors */
6.   if(( $e \in GF_j \&\& L_j$  is  $F_i$ ) || ( $e \in GB_j \&\& L_j$  is  $B_i$ )) then
7.     insert  $\langle ID_j, V_j \rangle$  into  $L_i$  at the position just before  $e$ 
8.      $added \leftarrow true$ 
9.   endif
/*  $L_i$  is empty or  $n_j$  is the farthest one */
10. if( $added != true$ ) then
11.   append  $\langle ID_j, V_j \rangle$  in  $L_i$ 
12. endif
13. reset the lifetime of  $\langle ID_j, V_j \rangle$  in  $L_i$ 
    
```

As described in TABLE II, If the message received from  $f\_antenna$ , then node  $n_j$  will be put into list  $F_i$ ; otherwise, into list  $B_i$ . So as shown in the above table, assume  $L_i$  is  $n_i$ 's  $F_i$ , then node  $n_i$  will check every element  $e$  on the link list  $F_i$ . If  $e$  is not in group  $GB_j$  or  $GF_j$ , then it means node  $e.ID$  and  $n_j$  are *hidden neighbors*. Hence it will be ignored. If  $e$  belongs to group  $GF_j$ , that means  $e$  is in front of node  $n_j$ ; thus, all elements from  $e$  to the tail of  $F_i$  are in front of  $n_j$  because all the other elements in  $F_i$  are in front of  $e$ . Because all the checked elements are behind  $e$  and  $n_j$ , we can put  $n_j$  into the list at the position just before  $e$ . If we can not find the proper position of  $n_j$  after checking all elements in the list, then it means either such list is null or  $n_j$  is the farthest node compared with others. In this case, we just append such node onto the list.

### 2) Relative Distance Update

Till now we can establish the relative positions of nodes for the entire network. However, we do not know the relative distance between them. As described above, some neighbors may become anchor nodes that have more accurate position. Thus, we can use those anchor nodes' relative distance to estimate other distances. If there is no anchor node in the list, then nodes' distances will be estimated to be evenly distributed between each neighbor. If there is at least one anchor node, we will divide the list into segments and update the relative distance of each node in each segment. Also, before divide the list into segments we need to update the anchor nodes' relative positions based on the old distance, velocity difference and the beacon period. Although anchor nodes' relative distances are more accurate than others', there may still be errors. Therefore, currently GPSFR is only suitable for applications which do not require accurate location information of vehicles. For example, the adaptive cruise control systems as mentioned above.

### 3) Vehicles at Interchanges

On rural highways, there will be many interchanges which are road junctions that typically utilize grade separations, and one or more ramps, to permit traffic on at least one road to pass over or under the highway. Vehicles near an interchange do not need to stop. However, some vehicles may slow down if they are leaving the highway, which will be discussed in following section. Here, we only focus on vehicles at interchanges that are not exiting. Let a node,  $n_i$ , moves through an overpass over a road  $road_1$ . At the interchange, it may receive packets from nodes moving on the over-passed  $road_2$ . However, since the directions of those nodes will be different from that of node  $n_i$ , these packets are dropped. To safely drop packets, a precondition is that all vehicles are moving on the original road, that means there is no steering information in the last few seconds. To implement this, the packet header will contain a flag field that is set if the vehicle steers in the last few seconds.

### 4) Vehicles Joining and Leaving Highways

Vehicles frequently join and leave the highways, making vehicular networks more dynamic than the other cases above.

#### a) Vehicles Joining in Highways

Vehicles can join the highways by only two ways: entrance paths or ramps of the interchange. However, there are many types of ramps and entrance paths. Regardless of the type, the entrance will be located besides the highway at a certain angle. That means vehicles on the entrance will have a different moving direction compared with those already on the highway.

Let us assume node  $n_j$  is trying to join in the network, and  $N$  denotes the set composed by nodes which were already in the networks. To join into the networking,  $n_j$  will periodically send out a *join-in* message in the form of  $\langle id, direction \rangle$  until it receives a response—a *group\_update* message, which contains information about itself, from nodes in the existing networks. After receiving the *join-in* message, each node  $n_i \in N$  will first check whether it is moving at the same direction. If so, they add  $n_j$  into the corresponding group (*fgroup* or *bgroup*) by checking which antenna element the message was received and then sent

out the new *group\_update* message to its neighbors as usual; otherwise, just drop this packet. So after fully merged into the highway,  $n_j$  can receive all *group\_update* message from nodes in  $N$ . Hence by running the *RPM* algorithm, node  $n_j$  can build up the relative positions and the group information of the new neighbors. Then after one beaconing period, all nodes in  $N$  will be able to arrange  $n_j$  into the correct link list.

In conclusion, joining vehicle will participate in networking if and only if they are already in the network. That means vehicles on the entrance path or ramp will not participate in the routing task and will not cause packet to be routed out the highway networks.

#### b) Vehicles leaving Highways

In this scenario, if the packet is propagated backward through the network, there will be no problem since the exiting node will not become the next hop. However, if the packet need to be forwarded to the front and the forwarder (vehicle) is trying to leave the highway, then this packet will go out of networks. To avoid this problem, the best choice is to equip each vehicle with an exit detection sensor. Although such kind of image sensors may become popular in the future, we can not only rely on it mainly for the reason of cost increase. Another method is to backup the sent message to another rear node, if the current forwarder is trying to leave. In the following, we will discuss the details.

First, we assume that typically exits are located on the right side of highways, because left-hand exits are contrary to the concept of driver expectancy when intermixed with right-hand entrances and exits [14]. In another word, vehicles trying to make a left steering will be regarded as a *Safe* node (that means it is safe to deliver packets to it). But if the vehicle makes a right turn, two cases may occur: this vehicle is on a right-turn curve of the highway, or it is leaving the highway. To avoid routing packet out of networking when the forwarder is leaving the highways, we need to backup the message; if this neighbor is also leaving, backup will be executed again. On the other hand, if the vehicle is moving on a curve, the backup process will keep on running until there is no vehicle approaching the curve. Suppose there are  $k$  nodes participated in the backup process, and then the problem is how to choose optimal value of  $k$ .

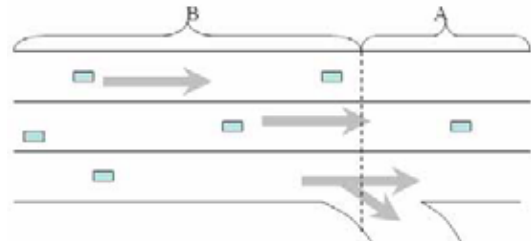


Fig. 4. Vehicles moving around an exit of the highway.

Assume that on the highway, around the exit, there are  $m$  cars connected through wireless links and one of them, e.g.  $q$ , is located at the junction between the highway and exit. Then there will be  $m + 1$  possible deployment of these  $m$  nodes. For example, one is in the *A* area and  $m - 1$  in the *B* area, as shown in Fig. 4. If there is one or more nodes in the *A* area, then the

packet will pass the exit successfully. A problem arises only if there is no vehicle in the  $A$  area, so the probability of this case will be  $\frac{1}{(m+1)}$ . Now suppose all  $m$  nodes are in area  $B$ , and

then the probability of  $k$  continuous nodes leaving off the highway will be  $\frac{1}{C_m^k}$ . Therefore, we can obtain that after  $k$

times of backup, the probability of packet being routed out of the network is:

$$P_{out} = \frac{1}{m+1} \cdot \frac{1}{C_m^k} \cdot p^k \quad (2)$$

$p$  is the probability that the node leaves the highways — the largest value for this is 0.5. Now assume we are trying to forward a packet through a routing path with  $n$  exit junctions. Then the probability  $p$  of the last exit is 0.5, while the probability of the vehicle leaving at the first exit is  $\frac{1}{(n+1)}$ .

Therefore, we can calculate the probability of the packet successfully reach the destination as:

$$P_{suc} = \prod_{i=1}^{i=n} \left( 1 - \frac{1}{m_i+1} \cdot \frac{1}{C_{m_i}^k} \cdot \left( \frac{1}{i+1} \right)^k \right) \quad (3)$$

It should be noted that sometimes  $m_i$  might be smaller than  $k$ . In this case, the packet will be routed out of the range. In fact, this failure is caused by a network partition, so it will not be counted into the performance of routing protocol. Most likely  $m_i$  will be bigger than  $k$ , so the minimal value of  $P_{suc}$  can be obtained when  $m$  is equal to  $k$ . Now we have

$$P_{suc} > \prod_{i=1}^{i=n} \left( 1 - \frac{1}{k+1} \cdot \left( \frac{1}{i+1} \right)^k \right) \quad (4)$$

Let  $k$  equals to 3, then the  $P_{suc}$  will be within [0.95, 0.96]. In fact, the value of  $P_{suc}$  will increase when  $k$  increased. However, we believe the value of 0.95 was already larger enough for our network routing protocol. So, in the implementation we choose 3 as the  $k$ 's value. The detailed routing algorithm will be shown in the next section.

## B. Routing Algorithm

### 1) Distance Advance

In GPSR [17], the current node  $n_i$  greedily selects one neighbor that is closet to the destination as the next hop. The implicit goal of such strategy is to maximize the distance advance and eventually minimize the total hop numbers. Let us denote such *distance advance* ( $ADV$ ) of a neighbor  $n_j$  as

$$ADV_j = \begin{cases} \frac{d_{ij}}{r_i} & (\text{if } d_{ij} \leq r_i) \\ 0 & (\text{if } d_{ij} > r_i) \end{cases} \quad (5)$$

$d_{ij}$  is the relative distance between node  $n_i$  and  $n_j$ . No matter whether  $n_j$  is behind or in front of  $n_i$  the value  $d_{ij}$  is always larger or equal to zero. Clearly, the conventional geographic routing protocols try to maximize  $ADV$  of next hop.

### 2) Balanced Advance ( $BADV$ )

*Balanced Advance* ( $BADV$ ) aims to avoid choosing an unstable node as the next hop while gaining as much distance

advance as possible. The goal of  $BADV$  is to the balance between the large distance advance and good link stability.

$$BADV_j = \alpha \cdot ADV_j + (1 - \alpha) \cdot e^{-\frac{\Delta v_{ij}}{d}} \quad (\text{if } \Delta v_{ij} < 0, e^{-\frac{\Delta v_{ij}}{d}} = 1) \quad (6)$$

$\Delta v_{ij} = v_j - v_i$  is the velocity difference between  $n_j$  and  $n_i$ , and  $d$  is the distance from  $n_j$ 's current position to the edge of  $n_i$ 's communication range. Therefore, suppose  $t_j$  is the time used by  $n_j$  to move out of  $n_i$ 's range; then a longer  $t_j$  implicates a more stable link between  $n_i$  and  $n_j$ . If  $\Delta v_{ij}$  is less than zero, that means node  $n_j$  is moving closer towards  $n_i$ . In this case, we consider the link stability as 1 because such link will become stronger and stronger in the future, until  $n_j$  move into the different neighbor group of  $n_i$ . Since the beaconing period will be 1 – 2 seconds, so the  $\Delta v_{ij}$  within such a short time will not change too much. Thus, we can trust this value for at least one beaconing period.

Although the concept of  $BADV$  is simple, it has many benefits in the wireless intelligent vehicles networks. First, since the link chosen by  $BADV$  will usually survive for a few seconds, we can adapt the idea in [15] to send burst message with only one pair of RTS/CTS. Second, fewer changes of next hop will reduce the channel switching overhead. For example, if [16] was adapted as the MAC protocol, then there will be a huge time slot allocation overhead due to the frequent next hop change. Third, since the data loss rate of sending packet to the node with the largest relative distance is always higher than the others, therefore by using  $BADV$  the total data loss rate among networking can be reduced. However, if the data is an emergency message which means it must be delivered as soon as possible, then GPSFR will use the maximal  $ADV$  policy to select the next hop. This is because, for such packets, there will be no benefit to choose a stable path compared with the shortest one to the destination. The detailed routing algorithm is shown below in TABLE IV.

TABLE IV  
ILLUSTRATION OF THE LOOK-AHEAD ROUTING

<p><b>Algorithm:</b> Look-Ahead Routing  <b>Input:</b> Current node ID <math>i</math> and packet deliver direction <math>dir</math>  <b>Output:</b> Next hop's ID <math>j</math>  <math>F_i</math>: Ordered link list of neighbors located in front of <math>n_i</math>  <math>B_i</math>: Ordered link list of neighbors located behind <math>n_i</math>  <math>e</math>: Temp variable holding the element of the ordered link list</p> <ol style="list-style-type: none"> <li>1. <b>if</b> (<math>dir == forwarding</math>) <b>then</b></li> <li>2.   <b>if</b> (<math>F_i == NULL</math>) <b>then</b></li> <li>3.     carry and forward the msg.</li> <li>4.   <b>if</b> (turning right during waiting) <b>then</b></li> <li>5.     <b>if</b> (counter of the message &lt; 3) <b>then</b></li> <li>6.       <math>e \leftarrow</math> the first element of <math>B_i</math></li> <li>7.       <b>if</b> (<math>e != NULL</math>) <b>then</b></li> <li>8.         increase the counter and backup msg. to <math>e</math></li> <li>9.       <b>else</b></li> <li>10.         carry and forward the msg.</li> <li>11.       <b>endif</b></li> <li>12.     <b>else</b></li> <li>13.       drop the msg.</li> </ol>
--

```

14.     endif
15.     endif
16.     endif
17.     if( $F_i \neq NULL$ ) then
18.          $e \leftarrow$  node with the largest  $BADV$ 
19.         forward message to  $e$ 
20.     endif
21. endif
22. if( $dir == \text{back warding}$ ) then
23.     if( $B_i \neq NULL$ ) then
24.          $e \leftarrow$  node with the largest  $BADV$ 
25.         forward message to  $e$ 
26.     else
27.         carry and forward the msg.
28.     endif
29. endif

```

## V. SIMULATION RESULTS AND EVALUATION

To measure how well we meet our design goals for GPSFR, we create a simulator using a CSIM18 Simulation Engine for the intelligent vehicle networks. We focus mainly on the two-lane highway scenario with different node densities and velocities in this paper. To compare the performance of GPSFR with the prior work for vehicle communication, we choose the well know GPSR [17] protocol.

### A. Simulation Environment

We simulate networks of 50, 112, and 250 nodes with 802.11 WLAN radios (250-meter radio range). Suppose the average velocity of vehicles moving on the left lane is 65 MPH and 45 MPH on the right lane. Then we randomly choose the speed of each vehicle within the range  $[V - 10, V + 10]$  where  $V$  is the average velocity of each lane. Usually, there are fewer vehicles on left lane than those on the right lane; so we give 1:2 as the ratio between them.

For the nodes deployment, we assume that all vehicles move on a straight highway, and only 5% of them leave the highway. For a certain vehicle, if it is deployed at position  $(x_i, y_i)$ , then the next vehicle will be located at position  $((x_i + i * d_s + rand(d_s)), y_i)$ .  $i$  is an integer number within range  $[0, 4]$  and  $d_s$  is the predefined distance (safe distance); and  $rand(x)$  is the function to generate a random number between 0 and  $x$ . Therefore, we will have 5 possible network densities, which are defined as the average number of neighbors moving on each lane. To simulate the high mobility attributes of intelligent vehicles, we give each node a possibility (10%, 20%, 50% and 100%) to change its velocity. The new velocity will be randomly chosen within the range  $[V - 5, V + 5]$  in each beaconing period. Finally, each simulation lasts for 300 seconds of simulated time, and all the result were obtained by running 100 times of the simulation with different nodes deployments. In our simulation, the location database overhead is ignored. Currently, we suppose all the source node knows whether the destination is behind or in front of it.

### B. Link Stability

Fig. 5 shows the ratio of link breakage to the next hop

between GPSFR and GPSR for varying network density, in which  $\alpha = 0.7$  (all the following simulations were done with the same parameter). Note that GPSFR always generate less link breakage than that of GPSR. However, in sparse networks, GPSFR performs almost the same as GPSR because only a few candidate nodes can be chosen as the next hop; so GPSFR may choose the same node as GPSR did.

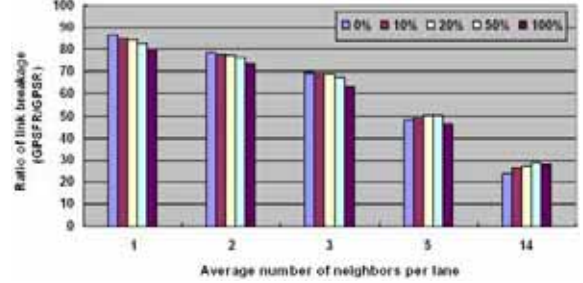


Fig. 5 Percentage of next hop changes various different network densities.

### C. Routing Path Length

Fig. 6 and 7 give the histogram of the number of hops beyond the length of path obtained by GPSR. The data are presented as percentages of all packets delivered across randomly distributed networking of GPSFR and GPSR. Here, the “0” bin counts packets delivered in the optimal, same number of hops as the path by GPSR; and the successive bins counts packets that took one hop longer, two hops longer, and so on. Note that no matter how dynamic the networking is, GPSFR always delivers the vast majority of packets in the optimal number of hops.

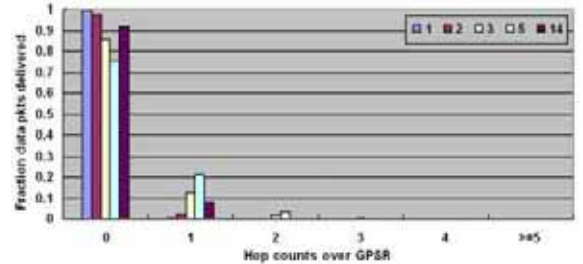


Fig. 6. Path length beyond GPSR when there is no velocity change.

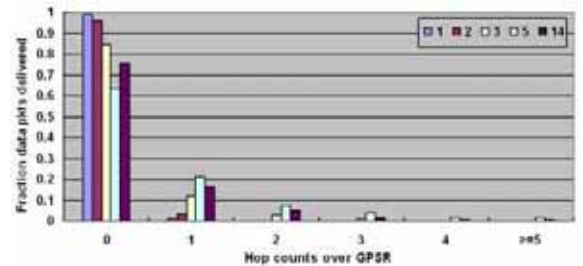


Fig. 7. Path length beyond GPSR when velocity changes at possibility 100%.

### D. Effect of Network Diameter

Fig. 8 and 9 present the path length beyond the optimal one for larger-scale (50-, 112- and 250-node) networks with identical node density (3 neighbors per lane). The intent of these simulations is to evaluate the scaling of GPSFR as network diameter increases. Although when routes become longer, the percentage of “0” bin will become smaller; we can find that GPSFR can still deliver majority of packets in very short paths.

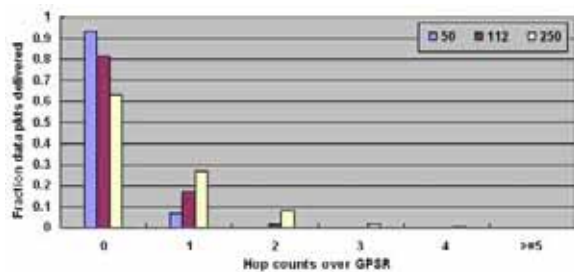


Fig. 8. Path length beyond GPSR's when no velocity changes.

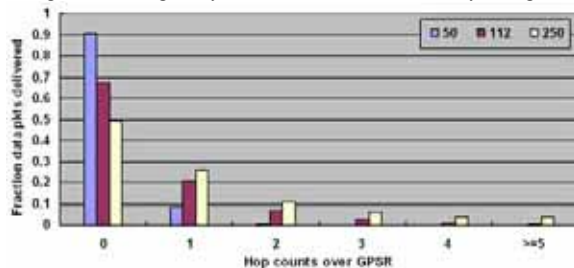


Fig. 9. Path length beyond GPSR's when velocity changes every second.

## VI. CONCLUSION AND FUTURE WORK

We have introduced GPS-Free Geographic Routing (GPSFR), a routing algorithm using relative positions to achieve small per-node routing state, robust packet delivery on densely and highly dynamic network made up of intelligent vehicles. To obtain the relative positions of neighbors and solve the hidden neighbor problem, we presented the Relative Position Maintenance (RPM) algorithm. Then geographic routing with *BADV* provides not only a better performance with link stability, but also routing paths that are very close to the optimal one generated by GPSR.

GPS-Free geographic routing with *BADV* benefits greatly from the routing decision for *forward* or *backward* propagation. That is because we assume all vehicles are moving on the rural highway. However, for the urban highway or the downtown streets, we still need more work. Thus, our future work is to design a new protocol based on GPSFR to meet the communication requirement of intelligent vehicles moving in the urban and even downtown environment. While we have shown herein the benefits of GPSFR as a routing protocol for intelligent vehicle system, measuring the combined behavior of GPSFR and a location database system will reveal more about the costs of using geography for routing. An efficient distributed location service would enable the network to be more useful and powerful.

## REFERENCES

- [1] J. Zhao and G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," in *IEEE INFOCOM*, Barcelona, Spain, Apr. 2006
- [2] H. Wu, R. Fujimoto, R. Guensler and M. Hunter, "MDDV: a mobility-centric data dissemination algorithm for vehicular networks," in *VANET '04*, Philadelphia, PA, USA, 2004, pp. 47–56
- [3] M. D. Dikaiakos, S. Iqbal, T. Nadeem and L. Iftode, "VITP: an information transfer protocol for vehicular computing," in *VANET '05*, Cologne, Germany, 2005, pp. 30–39
- [4] R. M. Yadumurthy, A. Chimalakonda, M. Sadashivaiah and R. Makanaboyina, "Reliable MAC broadcast protocol in directional and

- Omni-directional transmissions for vehicular ad hoc networks," in *VANET '05*, Cologne, Germany, 2005, pp. 10–19
- [5] M. Sadashivaiah, R. Makanaboyina, B. George and R. Raghavendra, "Performance evaluation of directional MAC protocol for inter-vehicle communication," in *VTC '05-Spring*, 2005 IEEE 61st Volume 4, 30 May-1 June 2005, pp. 2585–2589
- [6] T. K. Mak, K. P. Laberteaux and R. Sengupta, "A multi-channel VANET providing concurrent safety and commercial services," in *VANET '05*, Cologne, Germany, 2005, pp. 1–9
- [7] H. Füllner, M. Mauve, H. Hartenstein, M. Käsemann, and D. Vollmer, "Poster: Location-Based Routing for Vehicular Ad-Hoc Networks," in *Proc. of ACM MobiCom '02*, Atlanta, Georgia, Sept. 2002.
- [8] S. Capkun, M. Hamdi and J.P. Hubaux, "GPS-free positioning in mobile Ad-Hoc networks," in *Proc. of Hawaii International Conference on System Sciences*, Maui, HI, January 2001, pp. 3481–3490
- [9] R. Iyengar and B. Sikdar, "Scalable and Distributed GPS free Positioning for Sensor Networks," in *ICC '03*, IEEE International Conference on Volume 1, 11-15 May 2003, pp. 338–342
- [10] A. Rao, C. Papadimitriou, S. Shenker and I. Stoica, "Geographic routing without location information," in *Proc. of ACM MobiCom '03*, San Diego, CA, USA, 2003, pp. 96–108
- [11] V. Kukshya, H. Krishnan and C. Kellum, "Design of a System Solution for Relative Positioning of Vehicles Using Vehicle-to-Vehicle Radio Communications During GPS Outages," in *VTC '05-Fall*, 2005 IEEE 62nd Volume 2, 25-28 Sept., 2005, pp. 1313–1317
- [12] Report and Press Release – The Economic Impact of Motor Vehicle Crashes, 2000, posted on National Highway Traffic Safety Administration website
- [13] C. Liberti and T. S. Rappaport, "Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications," Prentice Hall, April 1999.
- [14] *Geometric Design of Highways and Streets*, American Association of State Highway and Transportation Officials, 2004
- [15] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," in *IEEE/ACM Trans. on Networking (TON)*, volume 12, Issue 3, June 2004, pp. 493–506
- [16] F. Borgonovo, L. Campelli, M. Cesana and L. Coletti, "MAC for ad-hoc inter-vehicle network: services and performance," in *VTC '03-Fall*, 2003 IEEE 58th Volume 5, 6-9 Oct. 2003, pp. 2789–2793
- [17] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. of ACM MobiCom '00*, Boston, Massachusetts, USA, 2000, pp. 243–254