Virtual Landmarks Assisted Routing Protocol in Vehicular Ad Hoc Networks

Jang-Ping Sheu
Dept. of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.

Guey-Yun Chang, Yu-Wen Chen and Wei-Hua Chen
Dept. of Computer Science and Information Engineering
National Central University
Jhongli, Taiwan, R.O.C.

Abstract—**How to route packets efficiently and reliably is an important issue in Vehicular Ad hoc Networks (VANETs). Due to the high mobility of vehicles, the communication topology of vehicles in VANET changes rapidly. In this paper, a Virtual Landmarks-based approach is suggested. Under the assistant of Virtual Landmarks, an efficient path is guaranteed even if the roads topology is complex. Our approach is applicable to routing packets between a vehicle and static destination. Simulation results shows that in comparison with previous work, our approach has much lowest communication cost and guarantees a reliable path.**

Keywords: Routing protocols; VANETs; wireless networks

I. INTRODUCTION

With the advancement of wireless networks technology and the growing population of vehicles, to provide more convenient driving life, a wireless ad hoc network formed by vehicles called Vehicular Ad Hoc Networks (VANETs) is being a new research topic nowadays. Many researchers hope that vehicles can exchange their information through the VANET system to provide more safety and comfortable driving environment [3, 4, 12]. VANETs deal with not only short distance information exchange between vehicles, but also for long distance communication, for example, checking the parking spaces around the static destination, booking a restaurant or movie theater, even to remote control intelligent household appliance, providing more convenience to driver.

To achieve those above-mentioned applications in VANETs, a robust and efficient packet delivery is necessary. In VANETs, vehicles on the road are used to deliver packets. But the communication range of microwave will be affected by buildings beside the road, and the communication range will be reduced. Further, VANETs have high mobility and frequent change in connected topology. Therefore, packet delivery is more unreliable in VANETs than in Mobile Ah Hoc Networks (MANETs). Besides, there is another problem that after a vehicle sent out a request packet to static destination, the static destination cannot reply information to the requesting vehicle because the requesting vehicle is no longer on the position which is recorded in the request packet.

In nowadays VANET environment, routing approaches can be classified into two categories: infrastructure based [9, 13], and ad-hoc based [2, 10, 15]. The former requires Road-Side-Units (RSUs) to assist the packet exchange and vehicles' communication in VANETs. By the aids of RSUs, packets route to the specific area can be pre-determined, and the packets can be stored in the buffer of RSU to wait for the best time to cast. Besides, RSUs can use its outstanding hardware performance to deal with the packet scheduling and packet delivery in VANET to reduce the collision rate. However, setting up and maintaining RSUs takes high cost and overhead and the network will be disconnected when these infrastructures are out of order. On the other hand, the vehicles can exchange information with their neighbors without additional devices or infrastructures. Ad-hoc based approaches do not take cost and overhead to set up hardware devices and have better adaptability of the changing topology.

In this paper, we propose a novel routing protocol for VANET. Our protocol works for communication between vehicles and static destinations in a city environment. The main idea of our protocol is based on the concept of virtual landmarks. Virtual landmarks (VLs) are pre-selected intersections in city map by some restrictions. These VLs connect with each other by roads with high traffic volume and formed a map called virtual landmark map (VLM). When a vehicle wants to send a request packet to a certain static destination, a robust routing path could be determined by the aid of VLM. The request packet is forwarded greedily from one VL to another which is on the routed path. After the static destination received and finished the request, the static destination will send a reply packet in the same way to the source vehicle.

With the virtual landmarks assisted, our protocol will have two main benefits: first is we can take the road segments' traffic load into account while selecting a path to routing packets. Therefore, the traffic load of each road segments of a routing path will be guaranteed; second, we can select a routing path to forwarding packet from source vehicle to destination in a global view. That is said, the routing path will be defined before sent out a packet, so every forward step on the routing path is good for the delivery form source vehicle to destination.

II. RELATED WORKS

The position-based routing approaches may be the most well-known ad-hoc based schemes [5-8]. The main idea of position-based approaches is that packet is forwarded to the vehicle which is nearest to the packet destination. Position-based approaches have well efficiency if the road topology is not complex, like high-way. But in the city scenario, it may result in local-maximum problem or packet delivery loops. Greedy Perimeter Stateless Routing for Wireless Networks (GPSR) [5], which is the most popular positioned-based routing ap-

---

This work was supported by the National Science Council of R.O.C. under grants NSC 98-2221-E-008-054 and NSC 97-2221-E-007--039-MY3.

proach, provided perimeter forwarding to recover from local-maximum. In [8, 15], not only the distance between the vehicle and the static destination, but also vehicle direction are the criteria for choosing the forwarding vehicle. In [14], the authors just count a shortest path between source and destination to be the packet routing path. An important part of packet delivery in VANET is that sending reply packet from static destination to source vehicle. The challenge of this task is after a source vehicle sent out the request packet; it keeps on moving, so when the destination receives its request packet, the source vehicle is no longer at the position that recorded in request packet. So, the destination cannot reply to the source vehicle.

Connectivity-Aware Routing (CAR) in VANET [10] designed a strategy called Guard to assist the packet delivery between source vehicle and the static destination. When a source vehicle wants to send a request packet to the destination and didn't know about the destination position, it broadcasts the request packet [11], otherwise, if it has the destination position, it will use greedy forwarding protocol to route request packet to destination. When the static destination receives the request packet and wants to send a reply packet to the source vehicle, the reply packet can be forwarded by the aids of the coordinates recorded in the request packet. After the source vehicle broadcast a request packet to the destination, upon its direction or velocity vector has changed, it can activate a guard. Vehicles who receive a guard activating message, it adds the guarding information (source vehicle's direction or velocity information) into its own guard-table and rebroadcast the guarding information. When the guarding vehicle receives the reply packet for the source vehicle, it modifies the information of the reply packet to find the source.

Many protocols have a characteristics that every selection of next hop vehicle is based on the condition of local view, such as the neighbors' direction or position, for example, greedily pick up the neighbor who is nearest to the destination. Though this vehicle is the closest to destination, it may run into a dead end. In these kinds of protocols, a reliable packet delivery is not ensured. To address this issue, we propose a novel protocol to route packets with a global view in the following section.

III. THE PROPOSED PROTOCOL

In this section, packet delivery between a source vehicle and static destination is considered. Clearly, there are two main parts of our protocol: (1) source vehicle sends a request packet to a static destination like a restaurant or movie theater, and (2) the static destination sends a reply packet to the source vehicle. The processing steps of our protocol are depicted as follows: when a source vehicle wants to send a packet to the destination, computer of the vehicle will route a path to the static destination by the aid of the Virtual Landmark Map (VLM) which is a graph whose nodes are Virtual Landmarks (the detail of VLM is explained later). Besides, the destination could send back a reply packet in the same way.

We assume that every vehicle has equipped vehicular computer system with Global Positioning System and digital map in it. Virtual landmarks (VLs) are pre-selected intersections in city map by some restrictions, for example, the intersections on main roads. These VLs connect with each other by roads with high traffic volume and formed a map called virtual landmark map (VLM). Formally, VLM is a graph whose nodes are VLs and links are roads segments which have rank higher than a threshold (Road segments are ranked according to the average density of vehicles passing through them. Road segments with high average vehicle density rank high). Notice that VLM is a connected graph (i.e., for any two VLs, say *A* and *B*, there exists a path between *A* and *B* in VLM).

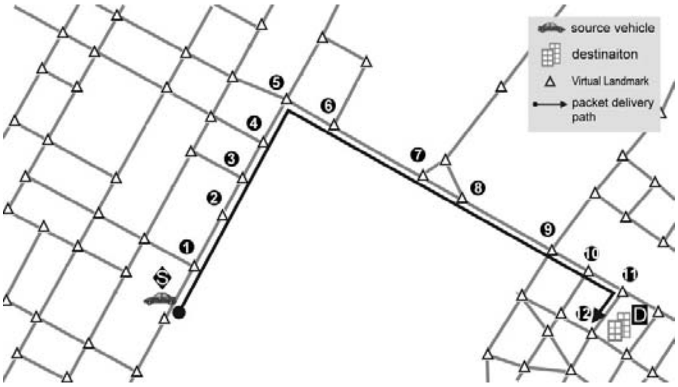


Figure 1. A city map with virtual landmarks which is marked as Δ.

A simple example is used to introduce our protocol steps by steps. In our protocol, while source vehicle *S* attempts to send a request packet to static destination *D*, source vehicle determine the VL closest to it, say V_S, and the VL closest to *D*, say V_D. By the aid of Dijkstra's algorithm, a minimum weighted path from V_S to V_D in VLM is determined. In order to choose a stable path, road segments with higher vehicle density have higher priority, i.e., lower weight. We define the weight of a link (or a road segment) *l* is as below:

$$W(l) = 1/(\text{vehicle density which link } l \text{ has})$$

We assume that the W(*l*) can be obtained from VLM. Notice that, the information of the routed path is not appended to the packet. Only the following VL on the routed path is recorded in the packet. So, packets in our protocol have fixed size. In Fig. 1, there are twelve VLs on the routed path from source vehicle *S* to static destination *D*. The format of request packet is shown in Fig. 2. There are seven fields in a packet. Type field records packet types. There are three packet types: request, reply and notification. The source ID and source location fields record the ID and location of source vehicle *S*, respectively. The destination ID and destination location fields record the ID and location of the destination *D*, respectively. Previous forwarder location field records the location of previous vehicle forwarding the request packet. Next field records the position of the right next VL on the routed path. Clearly, the request packet will be forwarded to the position recorded in Next field. When packet arrives at the VL, say *A*, recorded in Next field, the forwarding vehicle determines the next VL of *A* on the path. In order to reduce the times of re-compute the next VL, only the VL at the next turning point on the routed path is updated on the Next field. In Fig. 1, there are originally twelve VLs (i.e., VL 1, VL 2, …, VL 12) on the routed path. When the packet arrives at VL 1, its Next field will be updated to VL 5.

1 byte	4 bytes	8 bytes	4 bytes	8 bytes	8 bytes	8 bytes
Type	Source ID	Source location	Destination ID	Destination location	Previous forwarder location	Next

Figure 2. The format of request packet.

While a packet is transmitted through the path, each vehicle receiving the packet has to check whether it is at the VL recorded in Next field (in simulation, it is checked whether the vehicle is located within a radius 50 m of the VL which is recorded in Next field). If it is not, the vehicle should compete for forwarding the packet. Otherwise, the vehicle re-determines and modifies the content of Next field according to the destination position. In order to successfully choose a forwarder (e.g., avoid collision), each vehicle which participates the competition should wait for a back-off time according to (1).

Vehicles that are farther apart from the sender have shorter back-off time and could send packets early, and other vehicles could simply stop attempting to forward the packets while receiving the same packets. Therefore, we can select the vehicle which is the farthest to the sender to forward the packet. We calculate the back-off time W_{backoff} by using the following formula:

$$W_{backoff}(dist)=\left[\frac{R-dist}{l}\times 10+Random\_number\right]\times time\_slot \qquad (1)$$

where *dist* is the distance between the forwarding vehicle and the packet's sender; R is the maximum radio communication range (in our simulation is set to 250 m); l is a constant (in our simulation is set to 50 m); *Random_number* is a random number between 0 and 9; *time_slot* is the time for broadcasting a message and receiving it success (it is set 1 ms in simulations). The vehicle will drop the received packet if it receives the same packet before random back-off time is expired. Otherwise, the packet is forwarded. Note that, if a vehicle cannot hear its forwarding packet which is retransmitted by other vehicle after waiting $W_{timeout}$ period of time, the vehicle will retransmit the forwarding packet again. The $W_{timeout}$ is calculated as follows:

$$W_{timeout}=\left[\frac{R}{l}\times 10+Max\_random\right]\times time\_slot \qquad (2)$$

where *Max_random* is equal to 10 in our simulations.

When destination D received the request packet from source S, D will send a reply packet to S along the same routing path from S to D. The format of reply packet is shown in Fig. 3. The source location field and moving direction field record the latest location of the source vehicle S and its moving direction, respectively. The reply packet will keep forwarding to the source vehicle along the moving direction after it reach the source location. The definition of previous forwarder location field and Next field are the same as in Fig. 2.

1 byte	4 bytes	8 bytes	8 bytes	8 bytes	8 bytes
Type	Source ID	Source location	Moving direction	Previous forwarder location	Next

Figure 3. The format of reply packet.

In order to trace the updated location of the source vehicle, a notification packet will be sent to destination D when the source vehicle S changes its moving direction in the intersections. Since the reply packet sent by destination D will route back to the source vehicle S along with the same path from S to D. Some vehicles may receive both the notification packet and reply packet at some places of the routing path. The vehicles receiving the reply and notification packets will update the source location and moving direction fields of the reply packet by the notification packet. The notification packet format is shown in Fig. 4. The destination field is the source vehicle position of sending the last notification packet. In the first notification packet, this field will be filled with the location of the source vehicle sending a request packet. The source location field and moving direction field are recording the latest location and moving direction of the source vehicle.

If a notification packet arriving its destination location and cannot meet the corresponding reply packet, this notification packet will stay at the destination with a TTL time (in simulation we set to 3s). The staying packet will be rebroadcasted by the vehicles until the TTL reach to 0 or matched the reply packet.

1 byte	4 bytes	8 bytes	8 bytes	8 bytes	8 bytes	1 byte
Type	Source ID	Source location	Moving direction	Previous forwarder location	Destination	TTL

Figure 4. The format of notification packet.

For example, in Fig. 5, source vehicle S sends a request packet at position X to destination D and keeps on moving. The destination D will send a reply packet to source S with destination X after it received the request packet. Assume that the source vehicle S takes a right turn in location Y. Then source S will send a notification packet at location Y to destination X to inform its change of moving direction as shown in Fig. 5. For the convenient of description, we only show some fields of the notification packet and reply packet in Fig. 5. The fields from right to left of notification packet are: the destination, the moving direction of S; the latest location of S; and the source car ID. The reply packet fields from right to left are: the moving direction, destination of source vehicle S, and the source car ID.

In Fig. 5, assume the reply packet and notification-1 packet meet at location X. The reply packet will update its source location S and moving direction according to the information stored in notification-1 packet. If the source vehicle S is taking another turn in intersection Z, it will send notification-2 to destination X to inform its current location and moving direction. The vehicles between intersections Y and Z may receive both the reply packet and notification-2 packet and forward the updated reply packet to the source vehicle S.

Every notification packet will be sent to location X. Each vehicle at location X will update their notification packet information by the latest received notification packet. When the reply packet arrives the location X, it can obtain the latest source vehicle location and moving direction information from notification packet. Therefore, the reply packet will use VLM and the latest source vehicle location to calculate a routing path formed by VLs and route to that location. If the source vehicle S receives a reply packet from destination D, it will broadcast a cancel packet to notify its neighbors to delete this reply packet from their own buffers.

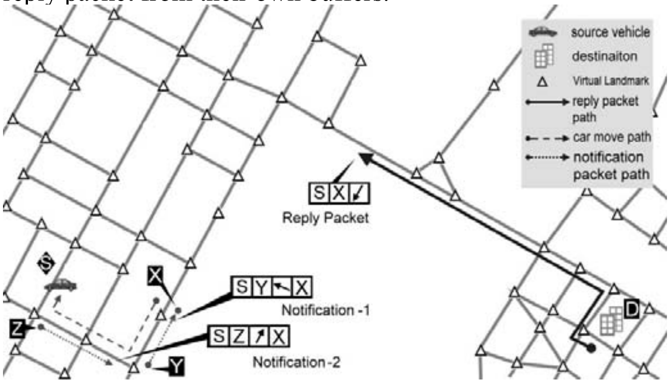


Figure 5. S sends the notification-1 at position Y and the notification-2 at position Z.

For the aid of Virtual Landmark, the routing path from source to destination will pass through the main roads with more number of vehicles than other roads. The advantages of using main roads to forward packets are: the connection between vehicles is more reliable and can avoid the dead end (local maximum) problem in position-based routing scheme such as GPRS. Based on the VLM, we can divide the routing path into several straight road segments and route the VLs one by one until reach the destination.

Our protocol is summarized as follows.

Routing Protocol with Virtual Landmarks:

For source vehicle:

1. With the aid of VLM, a request packet sending to the destination is forwarded to the first VL determined by the Dijkstra's algorithm. The source vehicle will rebroadcast the request packet if it cannot hear any neighbor forwarding this packet within $Wtimeout$.
2. When the source vehicle changes its moving direction at intersection, it will send a notification packet with the current location and moving direction to the location which is sending the previous notification packet or request packet.
3. When source vehicle received a reply packet from the destination, it will broadcast a cancel packet to inform its neighbors to delete the reply packet from their buffers.

For normal vehicles:

1. If a normal vehicle receives a request packet and the vehicle is located within the range *r* of the VL recorded in the received packet then the vehicle will update the VL field of the received packet.
2. If a normal vehicle receives a reply packet and a notification packet with the same source vehicle ID then update the source location and its moving direction of the reply packet by the notification packet.
3. Put the received packet into buffer and use formula (1) to calculate the back-off time $W_{backoff}$. The vehicle will drop the packet from its buffer if it receives the same packet before random back-off time is expired.
4. The packet will be rebroadcasted if the normal vehicle cannot hear any neighbor forwarding this packet within $W_{timeout}$.

For keeping notification packet vehicles:

1. Vehicle receives a notification packet, check if in the range of the destination of packet or if is a latest notification packet.
2. If vehicle is in the range, then keeps this notification packet in its buffer, or if it is a latest notification packet, then update the notification packet information in buffer.
3. While vehicle moving out of this range and the TTL value of the notification packet is bigger than 0, broadcast this notification packet forward to its destination and minus 1 of the TTL value. Otherwise, if the TTL value of notification packet reaches to 0, delete the packet from buffer.
4. If a vehicle who has a notification packet in buffer and receives a reply packet which has the same source vehicle ID with the notification packet. It will rewrite the data of reply packet by notification packet and rebroadcast the reply packet to the new destination.

For destination vehicle:

When the destination receives the request packet, it will broadcast a reply packet to the source vehicle.

IV. SIMULATION RESULTS

In this chapter, we evaluate the performance of our protocol through simulations. We use ns2 to simulate the task of wireless communications and VanetMobiSim [1] to simulate the navigation of vehicles. We can set the vehicle velocities, distance between vehicles, time of traffic lights, and road topology in the VanetMobiSim. The VanetMobiSim can output a vehicle navigation scenario data for ns2 and ns2 can simulate the wireless communications among vehicles in the output scenario.

We compare the performance of our protocol with a method using flooding to route packets and the previous work CAR [10]. We design a protocol who using flooding to forward the request packet to the destination. We will use this protocol to be the best case of delivery delay time and compare the throughput of our protocol and CAR, because using flooding can always find a shortest existing path from source vehicle to destination. To prevent forwarding loop, vehicles would not deal with a request packet twice. The main method of CAR protocol is using greedy forwarding to forward the request packet to destination while it knows the destination position and record each turning point position into the packet. Therefore, the destination can obtain a path from source vehicle to it when received the request packet. Note that, the vehicles in GPSR and CAR need to exchange hello beacons periodically to obtain the neighbors information.

In our simulations, we will compare the packet delivery delay time, routing overhead in packets, and the packet routing success rate from source vehicle to destination. The communication range of each vehicle is set to 250 m in our simulations and the speed of every vehicle in scenario is between 0 and 70 km/hr randomly navigating on the roads. We have two different simulation maps: one is a Manhattan-type map of 2500 m x 1800 m, formed by 4 rows and 7 columns as shown in Fig. 6. In this scenario, there are 3 main roads whose traffic load is 4 times than others roads. Another one is a city-like map of area 2500 m x 1800 m. There are two separate regions in the scenario and connected by a bridge as shows in Fig. 7. In the city-like map, we have 5 main roads whose traffic load is 4 times than other roads. In our simulations, the number of vehicles is varied from 100 to 400. In each simulation, the sources S_1 and S_2 send request packets to destinations D_1 and D_2, respectively.

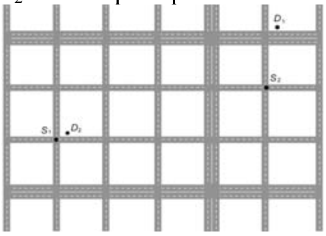


Figure 6. Manhattan-type map in simulations

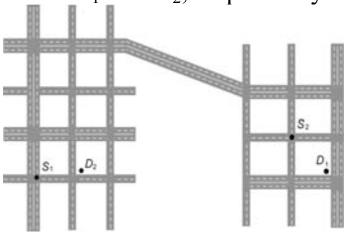
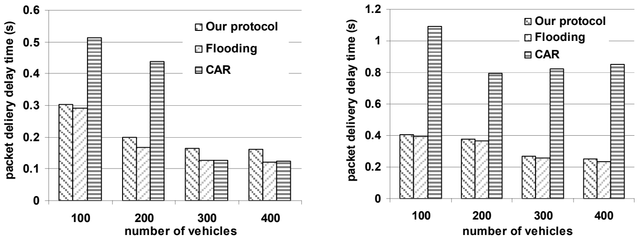


Figure 7. City-like map in simulations

Fig. 8(a) shows the results of packet delivery delay time for CAR, flooding and our routing protocols in Manhattan-type map. We can find that in the scenarios of 100, 200 vehicles the CAR protocol has an obviously long delivery delay time than other two protocols. This is because the vehicles in the Manhattan-type map may not be connected as the vehicular density is low. Since the CAR protocol simply selects the vehicle nearer to destination to forward packet, the packet has bigger chance be routed to dead end in lower vehicular density before it can reach the destination. As the vehicular density is high, the communication topology of Manhattan-type map will be connected and the delivery delay time of CAR protocol will close to flooding and even better than our protocol.

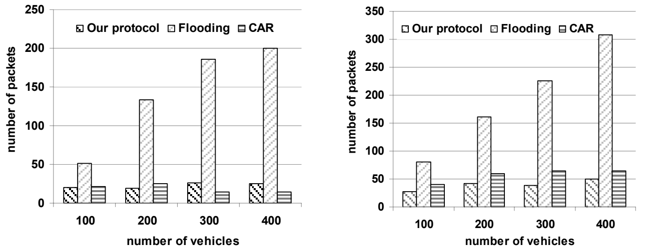
In Fig. 8(b), we found that the packet delivery delay time of our protocol and flooding protocol in city-like scenario will decrease with the increasing of the number of vehicles except CAR protocol. Since there is only one bridge to connect between the sources (S_1 and S_2) and destinations (D_1 and D_2) in the city-like map, the CAR with greedy forwarding scheme has higher probability to route packets to the dead end with the higher vehicular density. The increasing number of vehicles may make some disconnected lanes become accessible and rise the local-maximum problem times. So, the delivery delay time of CAR in 200 vehicles scenario will be shorter than that of in 300 and 400 vehicles scenarios. From the simulation results of Fig. 8, we have that the CAR protocol has bad packet delay time if the city map is complicated with various main roads and small lanes.



(a) Manhattan-type environment (b) City-like environment

Figure 8. Packet delivery delay time.

Fig. 9 shows the routing overhead of three protocols. First of all, vehicles in CAR protocols have to exchange information with neighbors by broadcast hello beacons periodically to keep a neighboring table. Since the communication overhead depends on the broadcast periods, we ignore the packet overhead of exchange hello beacons in CAR in our simulations. In Fig. 9 we can see that the number of packets sent by flooding is much larger than our protocol and CAR because it uses the flooding mechanism to find out a shortest path to the destination.



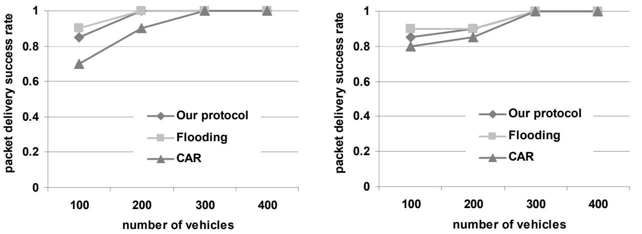
(a) Manhattan-type environment (b) City-like environment

Figure 9. The routing overhead in packets.

Fig. 10 shows the success rate of delivery packets from source vehicle to destination in three protocols. In the simulation results, the flooding protocol has the highest success rate in various vehicular densities. The flooding strategy can always find a path from source to destination if the topology is connected. The success rate of our protocol is better than CAR in low vehicle density. The packet delivery success rate of CAR is affected by the local-maximum problem, so in low vehicular density scenario its success rate is worse than other two protocols.

V. CONCLUSION

In this paper, we shows that with the Virtual Landmarks help, vehicles can easily find a reliable short routing path and have low routing packet overhead no matter the road topology is simply grids or complex. According to the simulation results, we can find that our protocol is efficient and reliable no matter in Manhattan-type map or City-like map. In the two different environments, our protocol can always route the packets through reliable paths with high vehicular density to the destinations. Our protocol has the better packet delay time, routing overhead and packet success rate than the previous work.



(a) Manhattan-type environment (b) City-like environment

Figure 10. The packet delivery success rate.

REFERENCES

[1] The Vehicular Mobility Simulator, http://vanet.eurecom.fr/

[2] Z. D. Chen, H. T. Kung, and D. Vlah, "Ad hoc relay wireless networks over moving vehicles on highways," in *Proceeding of Mobile Ad Hoc Networking and Computing*, pp. 247–250, 2001.

[3] M. D. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-aware services over vehicular ad-Hoc networks using car-to-car communication," *IEEE JSAC*, pp. 1590–1602, 2007.

[4] D. Hadaller, S. Keshav, and T. brecht, et.al. "Vehicular opportunistic communication under the microscope," in *Proceedings of Mobile Systems, Applications, and Services*, pp. 206–219, 2007.

[5] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of MobiCom*, pp. 243–254, 2000.

[6] J. Li, J. Janotti, D. D. Coutu, and R. M. D. Karger, "A scalable location service for geographic ad hoc routing," in *Proceedings of MobiCom*, pp. 120–130, 2000.

[7] C. Lochert, H. Hartenstein, J. Tian, H. Fuessler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proceeding of IEEE Intelligent Vehicles Symposium*, pp. 156–161, 2003

[8] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol.9, pp. 69–72, 2005.

[9] A. Mansy, M. Ammar, and E. Zegura, "Reliable roadside-to-roadside data transfer using vehicular traffic," in *Proceedings of Mobile Adhoc and Sensor Systems*, pp. 1–6, 2007.

[10] V. Naumov and T. R. Gross, "Connectivity-Aware Routing (CAR) in vehicular ad-hoc networks," in *Proceedings of INFOCOM*, pp. 1919–1927, 2007.

[11] V. Naumov, R. Baumann, T. Gross, "An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces," in Proceedings of MobiHoc, pp. 108–119, 2006.

[12] J. Ott and D. Kutscher, "Drive-thru internet: IEEE 802.11b for automobile users," in *Proceedings of ICC*, pp. 362–373, 2004.

[13] B. Petit, M. Ammar, and R. Fujimoto, "Protocols for roadside-to-roadside data relaying over vehicular networks," in *Proceedings of WCNC*, pp. 294–299, 2006.

[14] W. Wang, F. Xie, M. Chatterjee, "TOPO: routing in large scale vehicular networks," in *Proceedings of Vehicular Technology Conference*, pp. 2106–2110, 2007.

[15] J. Zhao and G. Cao. "VADD: vehicle-assisted data delivery in vehicular ad Hoc networks," in *Proceedings of INFOCOM*, pp.1–12, 2006.