

Enhancing Junction-based Routing for Vehicular Ad-hoc Networks by Effective Routing Table Learning and Maintenance

Po-Jen Chuang and Ming-Chun Liu

*Department of Electrical and Computer Engineering, Tamkang University,
Tamsui, New Taipei City, Taiwan 25137, R. O. C.
pjchuang@ee.tku.edu.tw*

Abstract

We present a new routing protocol in this paper to enhance junction-based routing for vehicle-to-vehicle (V2V) vehicular ad-hoc networks (VANETs). Employing effective routing table learning and maintenance, the new protocol is able to accomplish better transmission stability and lower transmission cost. In practice, the mechanism of routing table learning will help a vehicle establish its own static path information by which to locate suitable relay vehicles in a more efficient way, and the practice of routing table maintenance may substantially reduce the probability of finding no suitable relay vehicles, to avoid unnecessary packet discarding. Extended simulation is conducted to evaluate the performance of our new protocol and related routing protocols. The results exhibit that our protocol performs constantly better than others in terms of packet delivery ratios, packet drop ratios and average delay time. It ensures more efficient transmission without additional control overhead in highly mobile V2V VANETs.

Keywords: *Vehicular ad-hoc networks (VANETs), vehicle-to-vehicle (V2V) environments, junction-based routing protocols, packet relay, experimental evaluation*

1. Introduction

We know that modern vehicles are often equipped with GPS [1] and that routing protocols using GPS in vehicular ad-hoc networks (VANETs) [2-3] are position-based [4-5]. According to the characteristics of the city scenario, possible direction changes for packet transmission in VANETs happen mainly at junctions, not on straight streets, because buildings in the urban environment tend to block transmission signals. As position-based routing protocols usually take the advantages of junctions for packet relay, they are also called “junction-based” routing protocols. In previous junction-based routing protocols, a vehicle needs to judge if it is at a junction by the help of other vehicles or by different designs as in some protocols [6] which may cause extra control overhead. Nowadays, aided by GPS and digital maps, each vehicle can automatically learn if it arrives at a junction. That is, current junction-based routing protocols do not need to provide extra algorithms for judging if a vehicle is at a junction [7-8].

A number of protocols have been introduced to attain efficient junction-based routing for the vehicle-to-vehicle (V2V) VANET environment assisted with GPS and digital maps. Some protocols, such as the Junction-based Multipath Source Routing (JMSR) protocol [7], let each packet carry a predetermined static path in its header. The static path contains a sequence of junctions calculated according to GPS and digital maps by the source of the packet, and each vehicle receiving a packet will simply check the packet for the static path information to find relay vehicles. Other protocols, including the Greedy Perimeter Coordinator Routing (GPCR) [9] and Junction-Based Geographic Routing (JBR) [10] protocols, choose to search relay vehicles in a dynamic, instead of static, way. For instance, when vehicles receive incoming packets, they can locate appropriate relay vehicles by such mechanisms as the greedy forwarding or perimeter forwarding method.

To attain more efficient junction-based routing for the V2V VANET environment, we present a new routing protocol in this paper. The new protocol works by two major mechanisms: effective routing table learning and maintenance. Its routing table learning helps each vehicle establish its own static path information, and when receiving a packet, a vehicle can move on to check its own table for suitable relay vehicles. This is a different design from previous “static” protocols which need to record the path predetermined by the source in the packet to conduct packet relaying. Our routing table learning is apparently a better choice as it helps reduce the packet length and corresponding communication cost. On the other side, when a vehicle receives a packet but cannot locate neighbor vehicles for further relaying, our new protocol will not discard the packet but instead dynamically search alternative vehicles by the greedy forwarding or perimeter forwarding methods to fulfill the ongoing packet relaying. It will then record the updated results in the routing table to attain routing table maintenance which may largely cut down the probability of finding no suitable relaying vehicles, increase transmission stability and meanwhile decrease the cost due to unnecessary packet discarding or repeated path searching.

Extensive simulation runs using SUMO (Simulation of Urban MObility) [11], MOVE (MObility model generator for VEhicular networks) [12] and NS-2 (Network Simulator - Version 2) [13] are conducted to compare the performance of our protocol and other protocols, including GPCR, JBR and JMSR. The obtained results show that, without additional control overhead, our new protocol can locate relay vehicles more accurately and hence ensure more efficient packet transmission in the highly mobile V2V VANET environment.

2. Background Study

A VANET is extended from a MANET. While a MANET [14] has such features as node mobility, limited bandwidth and transmission range and being independent of pre-constructed facilities, a VANET [15] has varied features. It generates more frequently changed topologies and wider moving ranges, and moves on the level in a restricted way. In contrast to MANETs, VANETs have more difficulties in maintaining connections. Therefore, the primary challenges for current VANET protocols will be how to overcome the problem of broken transmission routes which usually happens when vehicles move at high speeds and also the problem of transmission interruption due to the tall and massive urban buildings. To deal with the two major problems, VANET protocols tend to divide nodes into street nodes and junction nodes by algorithms. Routing protocols taking the advantages of junctions for packet relay are then called “junction-based” routing protocols.

In a VANET, packet transmission can be carried out by V2R or V2V. A vehicle can take the roadside unit (RSU) as the transmission media to perform the V2R transmission. In V2R transmission, the vehicle’s moving speeds and the transmission ranges of RSU may force vehicles to change RSU frequently. This will result in the “handover” problem and increased chances of packet loss. For V2R protocols, how to handle the RSU handover problem and reduce the risk of packet loss becomes a critical issue. In a VANET, we can also conduct vehicle to vehicle (V2V) transmission directly through wireless apparatus between vehicles. V2V transmission is convenient and direct, but without the assistance of street information (including the positions of street buildings, infrastructure and road facilities), it may have problems maintaining the location information of vehicles. When two moving vehicles run too distantly apart in the streets (i.e., running out of each other’s transmission range), they will likely lose communication. Despite of this, V2V transmission remains a more practical transmission way than V2R because it does not need as much construction cost as V2R.

In previous junction-based routing protocols, a vehicle needs to judge if it is at a junction by the help of other vehicles or by different designs as in some protocols [6] which may cause extra control overhead. Nowadays, aided by GPS and digital maps, each vehicle can automatically learn if it arrives at a junction. That is, current junction-based routing protocols do not need to provide extra algorithms for judging if a vehicle is at a junction [7-8]. To facilitate our future discussion, we will briefly introduce, in the following, a number of routing protocols which have been introduced to attain efficient junction-based routing for the V2V VANET environment. (Note that the two protocols in Sections 2.3 and 2.4 are assisted with GPS and digital maps.)

2.1. Greedy Perimeter Stateless Routing (Gpsr) [16]

GPSR is a position-based routing protocol using GPS to obtain the location information. It is less affected by topological changes and can therefore strengthen network scalability. When facing an empty topology, it will start the surrounding mode to bypass the block until finding an appropriate relay and then pass into the greedy mode, to save control packets. The protocol needs to deal with two problems: growing transmission delay (the topological information may produce empty topologies) and increasing discarded packets (no effective route repair). Figure 1 illustrates how GPSR operates. When node X receives a packet, it detects that the neighbor closest to destination D is node Y and therefore chooses Y as the next transmission hop. The example shows GPSR can attain the shortest packet transmission path. There are nevertheless failed examples, as Figure 2 shows. In Figure 2, when node X detects no other neighbors closer to destination D than itself, it then passes along node W or Y by the perimeter (PERI) mode to relay the packet – bypassing the empty part of the topology.

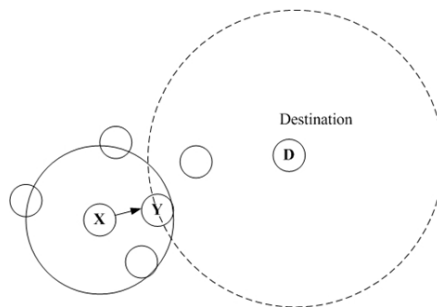


Figure 1. An Example Illustration for the Greedy Forwarding

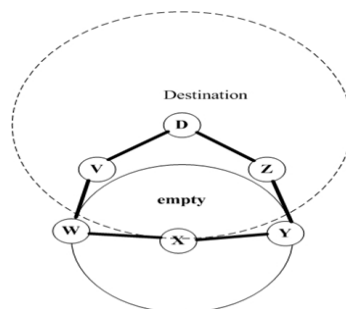


Figure 2. An Example of Greedy Forwarding Failure

2.2. Greedy Perimeter Coordinator Routing (Gpcr) [9]

GPCR is an extended form of GPSR. By considering the relationship between junctions and city streets, it is more suitable for performing routing in city environments. As GPCR performs routing directly on the planar map of junctions and streets, it need not practice the graph planarization in GPSR, but, by adding the relay junction judgment into the mechanism of GPSR only, it still faces the empty topology problem of GPSR.

Figure 3 demonstrates the different performance of GPSR and GPCR. As mentioned, GPSR will send the packet to a neighbor node closer to the destination, but in a city environment, such a routing mode may likely send a packet to a node which is practically unfeasible for transmission. In Figure 3, we can see that when node u sends the to-be-relay packet to node 1a according to the greedy approach of GPSR, node 1a will then transmit the packet to node 1b. Packet transmission thus enters the local optimum and will continue by using the PERI mode to search for a new relay node. GPCR works differently. It gives the junction node the highest relay priority. In the practice of GPCR, node u will send the packet to node 2a which then judges which junction neighbor node is closer to destination D. In this case, node 2a detects that the junction neighbor node closer to destination D is node 2b and then relays the packet to node 2b. Upon receiving the packet, node 2b can instantly relay it to the destination, thus avoiding the problem of local optimum in GPSR.

Unlike GPSR, GPCR has a repair mechanism to fix the problem of local optimum. When local optimum occurs, GPCR will initiate the repair mode by recording the node which enters the situation and then using the right hand rule to find the next hop (note that the junction nodes still maintain the highest relay priority). Figure 4 displays an example of the repair act of GPCR. In this example, node S (the node entering the local optimum) uses the right hand rule to locate the next hop which then forwards the packet to node C1. C1 selects, also based on the right hand rule, node I as the next hop. Node I then chooses the junction node C2 (having the highest relay priority) as the next relay node. C2 moves on to relay the packet to node L which eventually sends the packet to destination D. Since the distance between nodes D and L is shorter than that between D and S, the repair mode will stop here. Transmission returns to the regular greedy mode.

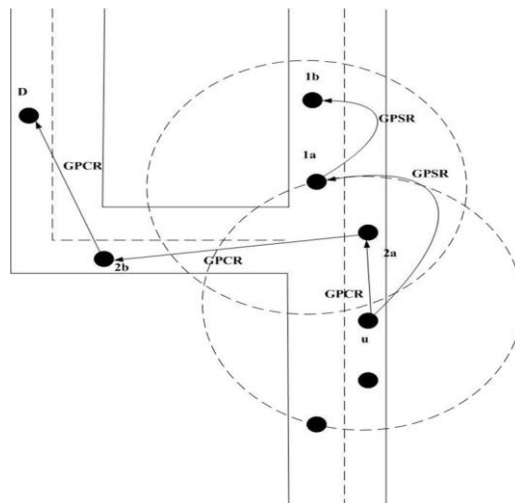


Figure 3. Greedy Forwarding in the GPCR Protocol

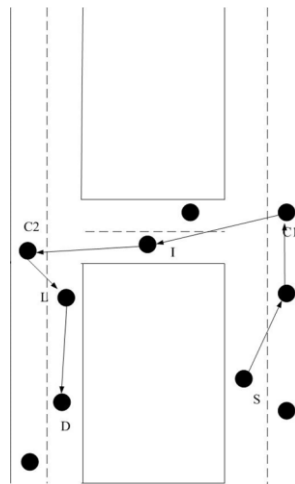


Figure 4. The Repair Mode in the GPCR Protocol

2.3. Junction-Based Geographic Routing (Jbr) [10]

By combining the function of GPS and geographic data systems, JBR allows nodes to learn more accurately about their locations: at the junctions or in the streets. When selecting a junction node for greedy forwarding, JBR will pick up a node at the junction that sits closer to the destination for packet relay. Basically, JBR remains a routing protocol utilizing the information of coordinates to conduct transmission. It may also encounter the problem of empty topologies.

In the practice of greedy forwarding, when a node receives a packet, it tends to relay the packet to a neighbor node which locates possibly the farthest from the node itself and the nearest to the destination. Such a neighbor node usually situates on the border of the node's transmission range. In JBR, when a node is to transfer a packet and the destination is not in its own transmission range, it will start searching the neighbor nodes. JBR divides nodes into simple node and coordinator nodes. Simple nodes refer to nodes located in the streets, while coordinator nodes are nodes staying at the junctions. JBR and GPCR perform distinctly. GPCR needs to perform the junction judgment at each junction, with the purpose of avoiding the occurrence of local optimum. But such an approach may make a node misjudge the routing direction and consequently result in an undesirable or failed transmission. To avoid the situation, JBR will choose a coordinator node which sits at the junction near the destination for packet relay.

Figure 5 gives an example to display the different operations of GPCR and JBR. It shows that, to transmit the packet to the destination, GPCR must transfer the packet by nodes S, A, B, C and D which are respectively located at junctions J1 to J5, whereas JBR can complete the transmission by way of vehicle B only.

In JBR, when a node is searching the next hop for packet relay but finds that it turns out to be the nearest node to the destination in the transmission range, the recovery mode will start. In the recovery mode, junction nodes also maintain the highest relay priority. The basic practice will be connecting each of the nodes in this transmission range (including the one that enters the recovery mode and all others) to the destination node, and take the node with the smallest connection angle to the destination as the next relay node. Figure 6 illustrates how the recovery mode works. Assuming node S is the node that enters the recovery mode, it will first send the packet to node C2 which is the junction node in its neighbor table. According to the repair mechanism, node C2 will then choose the node with the smallest connection angle (in this case, the node with angle 3) as the next hop to relay the packet.

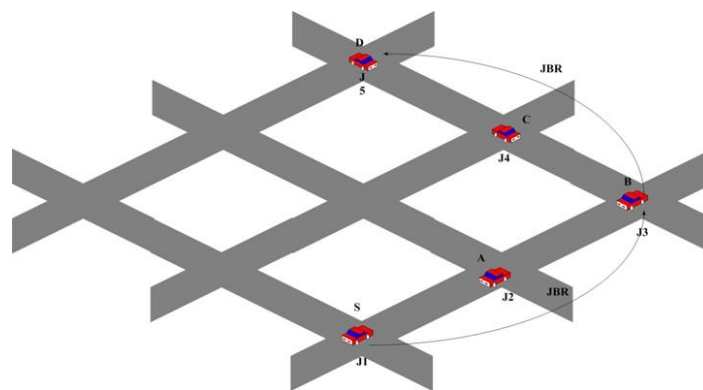


Figure 5. Greedy Forwarding in the JBR Protocol

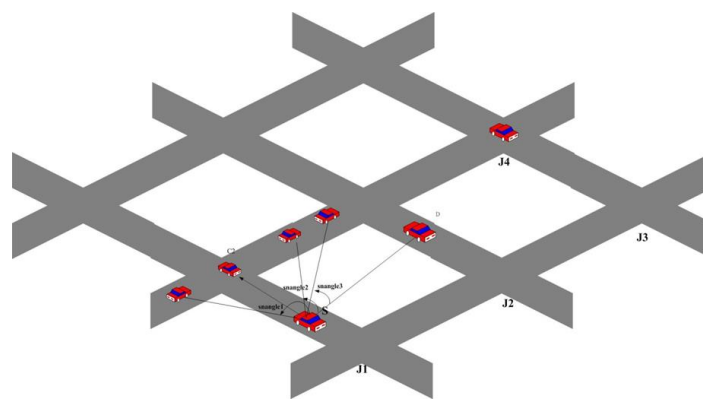


Figure 6. The Recovery Mode in the JBR Protocol

2.4. Junction-based Multipath Source Routing (JMSR) [7]:

JMSR is a routing protocol based on static junction information. It lets the packet to be relayed carry the junction data which can lead to the destination. By doing so, it can reduce the overall influence of node mobility and eventually strengthen packet transmission. As JMSR assumes a high junction density and that, under such a high junction density, a relay node can always be located, it has no route repair mechanism.

As mentioned, JMSR relays packets mainly based on junctions. Thus, the locations of junctions turn out more important than the locations of nodes. The good point is, the junction information is fixed and can be handily added to the packet header. When sending out a packet, the source will add a header – which carries the information of all junctions that will lead to the destination – to the packet. When a node receives a packet, it will follow the junction information in the header to choose the next relay node from its neighbor table. If there are multiple candidate next hops, the node will randomly select one to continue the transmission. In contrast to GPCR and JBR which use geographic data to judge and determine if a node is located in the junction and based on the result to perform transmission, JMSR is more straightforward: packet transmission simply follows a path which moves along the junctions pre-set in the packet header. As JMSR has no recovery mechanism, when it fails to locate any node at a pre-set junction (which JMSR ignores because it assumes a relay node can always be found under the high junction density of a city environment), it will simply discard the packet.

3. The Proposed Protocol

Our new protocol works mainly based on Routing Table Learning and Maintenance, and is hence called the RTLM protocol. We adopt the idea of switch table learning [17-18] to facilitate our routing table learning. A switch is a network device responsible for relaying packets. Each switch needs to know the port from which it can relay an incoming packet with a given destination. For this purpose, a switch must learn in advance which destination each of its ports can reach. This is the basic job of switch table learning. Switch table learning works as follows. Each switch has a switch table, also known as a MAC address table, in which each entry will record a (MAC address, port) pair. When a switch receives an incoming packet, it will record the source MAC address and incoming port of the packet in a table entry. In this way, we can consider each entry input, i.e., the (MAC address, port) pair, a piece of switch learning which exhibits that a packet can be relayed from the indicated "port" to a network node with the indicated "MAC address". Thus, when a switch is about to relay an incoming packet, it can simply look up the table for the destination MAC address of the packet to find the corresponding port for packet relay. If the destination MAC address of the incoming packet is not in the table, the switch will relay the packet from all of the ports except the incoming port.

Our routing table learning has a different design. Instead of recording a (MAC address, port) pair in each switch table entry, our routing table records a (junction, vehicle) pair in each entry where "junction" and "vehicle" respectively correspond to the "MAC address" and "port" in a switch table entry. In our design, when a vehicle receives an incoming packet, it will record the "junction" near the source and the sender "vehicle" in a table entry. In this way, we can take each entry input, i.e., the (junction, vehicle) pair, as a piece of vehicle learning to facilitate packet transmission, that is, to help relay a packet from the "vehicle" in an entry to another network vehicle near the "junction". That is, to relay an incoming packet, a vehicle will simply look up the routing table for the junction near the destination (of the packet) to obtain a corresponding vehicle for packet relay.

Our new protocol uses the static information recorded by routing table learning to find a relay vehicle, an approach which appears similar to the JMSR protocol [7]. The JMSR protocol lets a packet carry a predetermined path in its header. The predetermined path contains a sequence of junctions which the source obtains by GPS and digital maps. To find the relay vehicles, a vehicle receiving a packet may just check the packet for the predetermined path. We can hence easily detect the difference between the two protocols: JMSR needs to record the path predetermined by the source in a to-be-relayed packet, whereas our protocol implicitly record the path information in the routing table of each vehicle by routing table learning. In contrast to JMSR, our protocol can handily reduce the length of a packet and also the corresponding communication cost.

For the JMSR protocol, when a vehicle receives an incoming packet, checks the predetermined path in the packet for the next junction, but finds no neighbor vehicles residing in the junction (i.e., finding no relay vehicles for the packet), it will simply discard the packet. When facing the same situation, our protocol handles in a different way. It will execute a similar operation as the GPCR [9] and JBR [10] protocols. That is, when a vehicle receiving a packet detects that the relay vehicle attained from the routing table is no longer a neighbor vehicle, our protocol will adopt the greedy forwarding or perimeter forwarding method [6,9] to search dynamically for alternative relay vehicles and then record the newly found results in the routing table to update routing information and as a result achieve routing table maintenance. Aided by such routing table maintenance, our protocol can significantly reduce the probability of finding no suitable relay vehicles at a

designated junction during packet transmission. It ultimately lifts up the stability of transmission paths and decreases the cost due to unnecessary packet discarding as well as repeated path searching.

In practice, existing GPCR and JBR need to search a whole new routing path for each packet transmission. This may result in two situations: repeated selection of similar routing paths or/and repeatedly entering the repair mode. For both protocols, when different packets are to be sent to the same destination node or junction, they will start the path discovery mechanism at each of such transmission attempts and repeatedly turn over the same routing path. A path discovery mechanism like this will also lead to repeated routing failure and path repair, and eventually degrade transmission efficiency. This is because, when any packet transmission which travels along the same route enters the same area where local optimum used to happen, it is very likely to fall into local optimum again and then be forced into the repair mode. JMSR is free of the two situations. It transmits packets by the pre-fixed routes which are stored in the header of each to-be-sent packet. Having no path recovery mechanism nevertheless leads JMSR to a different problem. In its operation, when a packet moves to a pre-set junction but finds no nodes for further relay, it cannot travel forward and has to be dropped.

Our new protocol deals with the situation by practicing the proposed routing table maintenance. When detecting that a designated node is not in the transmission range, we will search for a new route and store the information of the updated route in the routing table. Thus, by achieving real-time routing table maintenance, we do not need to arrange or search for a whole new route each time when resending a packet. That is, to resend a packet, the mechanism of routing table maintenance enables us to carry on the transmission by a partially rearranged path, significantly improving packet transmission efficiency. In our protocol, as the node at a junction will take some time to move far away from the junction (and leave the transmission range of the node in search of a relay vehicle), we will have a better chance to find relay vehicles along the route maintained by the table.

When conducting transmission by greedy forwarding, we allow each node to search its own routing table for available relay nodes and junctions. If there is such an available node in the table, the node will transfer the packet accordingly. If a node discovers no information of a corresponding junction in its table or that a possible relay node has moved out of the transmission range, it will search for a new next hop by the greedy forwarding mode and record the new finding in the table. After the above attempts, if a node still fails to locate a suitable relay node or encounters the situation of local optimum, it will switch to the PERI routing mode (like existing GPSR) to find a new relay node for the transmission, store the updated information in the table, and then switch back to the regular greedy forwarding mode. At this point, if the node that receives the packet is able to locate a suitable relay node from its table, it will act to forward the packet; if unable to locate such a relay node in the table, it will start the above searching process until successfully finding one.

For better illustration, Figure 7 and Table 1 give, respectively, the operation flowchart and algorithm for each vehicle upon receiving an incoming packet.

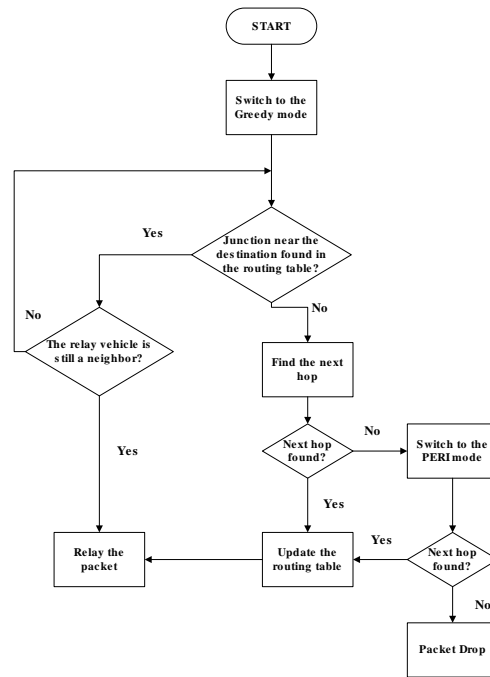


Figure 7. The Operation Flowchart for Each Vehicle upon Receiving an Incoming Packet

Table 1. The Algorithm for Each Vehicle upon Receiving an Incoming Packet

| | |
|----|---|
| 1. | //an incoming packet is received |
| 2. | switch to the Greedy mode; |
| 3. | while (a junction near the destination is found in the routing table) { |
| 4. | if (the relay vehicle for the junction is still a neighbor) |
| 5. | relay the packet through the relay vehicle; |
| 6. | } |
| 7. | use the Greedy mode to find the next hop; |
| 8. | if (next hop found) { |
| 9. | update the routing table; |
| 1 | relay the packet through the found next hop; |
| 0. | } |
| 1 | else { |
| 1. | switch to the PERI mode; |
| 1 | use the PERI mode to find the next hop; |
| 2. | if (next hop found) { |
| 1 | update the routing table; |
| 3. | relay the packet through the found next hop; |
| 1 | } |
| 4. | else |
| 1 | drop the packet; |
| 5. | } |
| 1 | } |
| 6. | } |
| 7. | } |
| 1 | } |
| 8 | } |

| |
|---|
| 1 |
| 9 |
| 2 |
| 0 |
| 2 |
| 1 |

4. Experimental Evaluation

In this section, we use the traffic simulation software SUMO [11] and MOVE [12] to generate a vehicular mobile model (which can mimic the city scenario and urban roadway traffic and form the simulated network topology) and then combine the model with NS2 [13] to carry out the simulation. Table 2 lists the environmental parameters of our settings. We have collected simulation results on packet delivery ratios (PDR), packet drop ratios, and average delay time (ADT) for performance comparisons between our RTLM protocol and other protocols, including GPCR [9], JBR [10] and JMSR [7].

Table 2. Simulation Parameters

| | |
|--------------------------|-----------------|
| MAC layer Protocol | 802.11p [19-20] |
| Transmission Radio Range | 250 m |
| Simulation Time | 1000 s |
| Region | 1000 m × 1000 m |
| Number of vehicles | 100 |
| Speed | 0~60 km/h |
| Beaconing Rate | 1s |
| CBR Flows | 5 ~ 30 |
| Packet Size | 512 bytes |

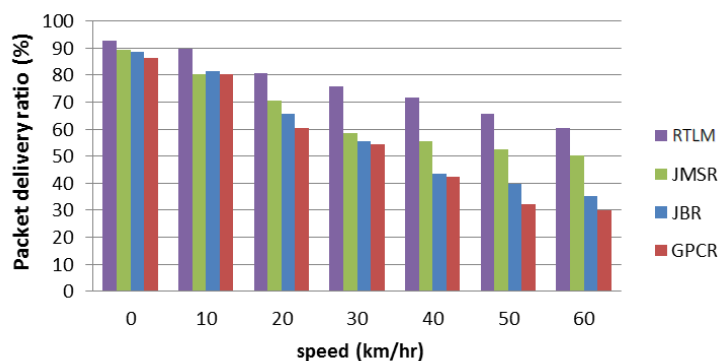


Figure 8. Packet Delivery Ratios (PDR) vs Speeds for the Protocols

Figure 8 depicts the result of packet delivery ratios (PDR) versus speeds for the above protocols. PDR is defined as (total packets received) / (total packets sent). Note that in the figure, each number in the x-axis indicates a speed range, not a single speed. For example, the number 30 indicates the simulated speed range is 0-30 m/s, and 0 indicates the simulated speed range is 0-0 m/s, i.e., all vehicles stop in the simulation scenario. Figure 8 shows that PDR drops down when the speed (the mobility of vehicles) increases. This is understandable because high vehicle mobility incurs more frequent network topology changes. Among the protocols, GPCR yields the least PDR because it locates a relay vehicle simply based on the coordinates of vehicles and, after locating one relay vehicle, it becomes less likely to find the next relay vehicle. Besides, at higher speeds, the search for relay vehicles may fall into local optima more frequently and so further worsen the PDR performance. In search of the next relay vehicle, the JBR protocol may first pick up the vehicle at a junction but will also consider the distance between the destination and the next relay vehicle to choose a vehicle with the shortest distance. By this practice, JBR can lessen the PDR degradation which GPCR tends to suffer (due to possible selection of inferior relay vehicles). In contrast to JBR and GPCR, the JMSR protocol can select better relay vehicles with its routing paths predetermined by the sources and aided by GPS and digital maps. JMSR nevertheless needs extra header space in the packet to store the predetermined path and will also face decreased PDR when unable to locate suitable relay vehicles at designated junctions. Of all protocols, ours constantly yields the highest PDR at different speeds and generates the least PDR change upon speed growth. It outperforms the others mainly due to its ability to solve or lessen their problems by the proposed routing table learning and maintenance.

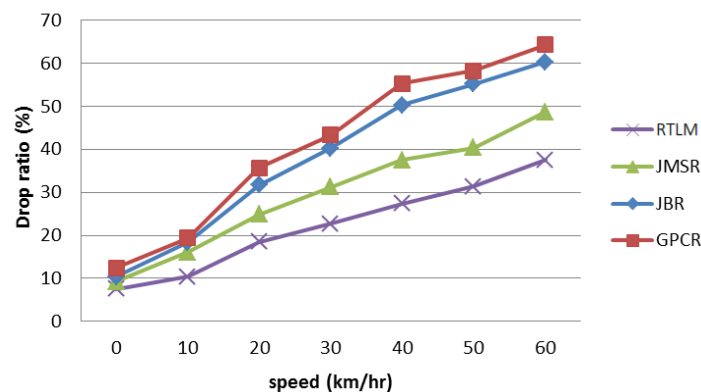


Figure 9. Packet Drop Ratios for the Protocols

Figure 9 depicts the packet drop ratios for the protocols. For GPCR and JBR, packets drop mainly because the two protocols select the worse directions when deciding the junctions for relay and thus result in failed packet transmission. The situation is especially obvious for GPCR because it needs to judge the junctions at every junction, which may lead the packets to the wrong directions and, as a result, to be dropped in higher probability. As for JMSR, it does not have any repair mechanisms when finding no vehicles to relay around junctions. Recall that JMSR assumes the density of vehicles around the junctions in city environments will be so high that the probability of finding no relay vehicles around the junctions is quite low. By contrast, we see that our protocol yields constantly lower packet drop ratios, under different speed assumptions, than the other protocols. This is because, when finding no relay vehicles at the junctions, our protocol has a better chance to locate other relay vehicles by the aid of routing tables and effective local arrangements. Note that packet drop ratios rise for all protocols when

speeds grow. This is reasonable as rising speeds will hasten vehicles away from junctions and thus raise the probability of finding no relay vehicles.

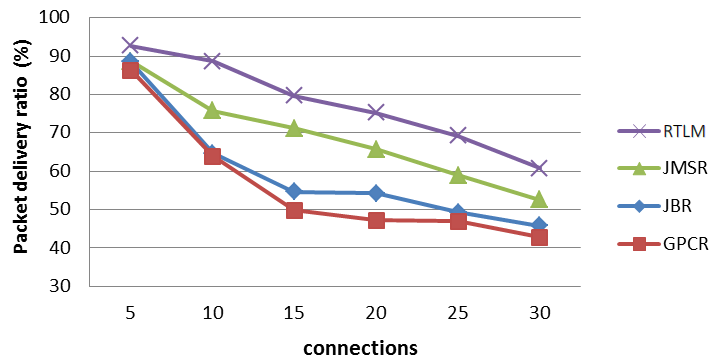


Figure 10. PDR vs Connections for the Protocols

Figure 10 illustrates packet delivery ratios (PDR) vs. connections for the four protocols. As the figure shows, when the number of connections grows, packet collision will cause PDR to drop. For JBR and GPCR which dynamically search for the path (i.e., the next relay vehicle) upon receiving each incoming packet, packet collision may frequently occur due to the same selection logic and network topology. They both will thus produce lower PDR. By contrast, JMSR, which uses paths predetermined by sources for packet transmission, can better distribute the traffic to avoid packet collision. Despite of its ability to reduce packet collision, JMSR fails to generate as good PDR as our new protocol. This is because JMSR will likely degrade its PDR performance when connections grow and thus bring up collisions or when packets are dropped due to failure to locate a suitable relay vehicle at a designated junction. Our new protocol is shown to produce constantly the highest PDR under different connections, thanks again to its effective routing table learning and maintenance which help locate relay vehicles more accurately and as a result ensure better transmission stability.

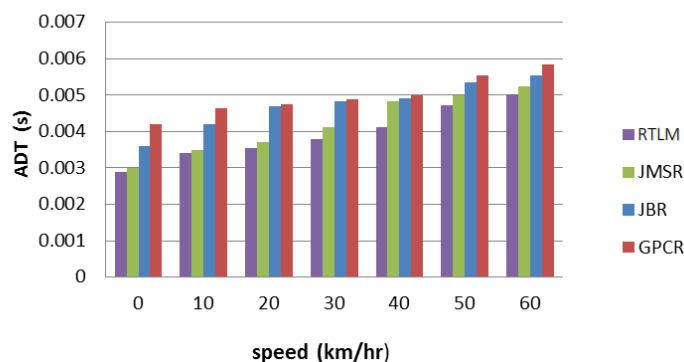


Figure 11. ADT Vs. Speeds for the Protocols

Figure 11 depicts the result of average delay time (ADT) versus speeds for the protocols. ADT is defined as (total packet End-to-End delay) / (total packets received). When the special topology of city streets tends to confine the mobility of vehicles to

restricted directions, speed growth may further broaden the affected area due to the increase of control packets from each vehicle. The consequence will be higher packet collision probability and longer packet transmission time. As Figure 11 demonstrates for all protocols, ADT increases when speeds grow. For both GPCR and JBR, selecting relay junctions and performing repair mechanisms both prolong their packet delay. The situation gets even worse at higher speeds. The delay for GPCR is especially distinctive as it needs to judge the next relay direction at each junction. When receiving a packet, JMSR uses the neighbor table to judge if a neighbor exists at the relay junction indicated in the pre-set route. It has less ADT than GPCR and JBR. Among all protocols, ours yields constantly the lowest ADT at various speeds, mainly because the proposed routing table learning and maintenance can help a packet reach its destination more efficiently.

5. Conclusions

We propose a new routing protocol in this paper to perform more efficient junction-based packet routing for V2V VANET environments. Our new protocol operates mainly on two approaches, effective routing table learning and maintenance. The proposed routing table learning helps each vehicle in a V2V VANET environment build its own static path information. When receiving an incoming packet, a vehicle can directly check its routing table to find suitable relay vehicles, to reduce the required packet length and save the communication cost of previous protocols. When a vehicle receives a packet but fails to locate suitable relay vehicles, our protocol will dynamically look for an alternative relay vehicle to continue packet transmission (instead of discarding the packet) by the greedy or perimeter forwarding approach. It then moves to record the updated results in the routing table to attain routing table maintenance. The design significantly reduces the probability of finding no suitable relay vehicles. It helps enhance the transmission stability and save the cost due to unnecessary packet discarding or repeated path searching. The obtained simulation results exhibit that, in contrast to related protocols, our new protocol can locate relay vehicles more accurately in a highly mobile V2V VANET environment and consequently enhance packet transmission with no extra control overhead.

Acknowledgements

This paper is a revised and expanded version of a paper entitled “Advanced Junction-based Routing in Vehicular Ad-hoc Networks” presented at the 9th International Conference on Future Generation Communication and Networking, Jeju Island, Korea, Nov. 25-28, 2015.

References

- [1] . Hightower and G. Borriello, “A Survey and Taxonomy of Location Systems for Ubiquitous Computing”, Technical Report UW-CSE 01-08-03, Department of Computer Science and Engineering, University of Washington, (2001).
- [2] F. Li and Y. Wang, “Routing in Vehicular Ad Hoc Networks: A Survey”, IEEE Vehicular Technology Magazine, vol. 2, no. 2, (2007), pp. 12-22.
- [3] S. Yousefi, M. S. Mousavi and M. Fathy, “Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives”, Proceedings of the 2006 International Conference on ITS Communication, (2006).
- [4] M. Mauve, A. Widmer and H. Hartenstein, “a Survey on Position-Based Routing in Mobile Ad-Hoc Networks”, IEEE Network Magazine, vol. 15, no. 6, pp. 30-39, (2001).
- [5] I. Stojmenovic, “Position-based Routing in Ad Hoc Networks”, IEEE. Communications Magazine, vol. 40, no. 7, (2002), pp. 128-134.
- [6] P.J. Chuang and T.L. Huang, “Efficient Vehicular Ad-hoc Networks Routing Based on Junctions”, IET Communications, vol. 9, no. 4, (2015), pp. 487-493.
- [7] P. Sermpezis, G Koltsidas and F.N. Pavlidou, “Investigating a Junction-based Multipath Source Routing Algorithm for VANETs”, IEEE Wireless Communication, vol. 17, (2013), pp. 600-603.

- [8] P.J. Chuang and M.C. Liu, "Advanced Junction-based Routing in Vehicular Ad-hoc Networks", Proceedings of the 9th International Conference on Future Generation Communication and Networking, (2015).
- [9] C. Lochert, M. Mauve, H. Fussler and H. Hartenstein, "Geographic Routing in City Scenarios", Mobile Computing and Communications Review, vol. 9, no. 1, pp. 69-72, (2005).
- [10] S. Tsiachris, G. Koltsidas and F. N. Pavlidou, "Junction-based Geographic Routing Algorithm for Vehicular Ad Hoc Networks", Wireless Personal Communication, vol. 71, (2012), pp. 955-973.
- [11] SUMO, <http://sumo.sourceforge.net>.
- [12] http://lens1.csie.ncku.edu.tw/MOVE/Example%20for%20step-by-step_Chinese.pdf.
- [13] NS2, <http://www.isi.edu/nsnam/ns/>.
- [14] A. Kumar and R. Hans, "Performance Analysis of DSDV, I-DSDV, OLSR, ZRP Proactive Routing Protocol in Mobile Ad Hoc Networks in IPv6", International Journal of Advanced Science and Technology, vol. 77, (2015), pp.25-36.
- [15] R. Sharma and J. Malhotra, "Performance Evaluation of ADV DSR and God in VANET for City and Highway Scenario", International Journal of Advanced Science and Technology, vol. 82, (2015), pp.1-10.
- [16] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", Proceedings of the 2000 IEEE International Conference on Mobile Computing and Networking, (2000).
- [17] http://www.netadmin.com.tw/article_content.aspx?sn=1203290001.
- [18] <http://ops9.blogspot.tw/2008/03/cisco-switch-port-security.html>.
- [19] http://jucs.org/jucs_20_4/classification_of_vanet_mac/jucs_20_04_0462_0487_ajmal.pdf.
- [20] <http://www.ttia-tw.org/knowledge.php?id=36>.