Macro-level and Micro-level Routing (MMR) for Urban Vehicular Ad Hoc Networks

Youndo Lee, Hojin Lee, Nakjung Choi, Yanghee Choi, and Taekyoung Kwon
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
{ydlee, lumiere, fomula}@mmlab.snu.ac.kr, {yhchoi,tkkwon}@snu.ac.kr

***Abstract*— Finding a reliable and efficient routing path in vehicular ad hoc networks (VANETs) is a challenging issue due to high mobility of vehicles and frequent link breakage. Motivated by this, we propose a robust and efficient routing protocol, called MMR. The contribution of this paper is two-fold: two-level routing and a new routing metric. A routing process of MMR consists of the macro level and the micro level. MMR forwards a packet to an approximate location of the destination at the macro level and then forwards a packet to the exact location of the destination at the micro level. This two-level routing reduces the protocol overhead and improves scalability in terms of the number of nodes. MMR also introduces a new routing metric that reduces the protocol overhead and path breakage by considering velocities of vehicles. Through simulations, we show that MMR improves the routing performance by about 30∼40% in highly mobile environments, compared to the existing ad hoc routing protocols such as AODV and GPSR**

I. Introduction

Over the past few years, vehicular ad hoc networks (VANETs) have been gaining attention from the academia, the industry and the government. FCC has allocated a portion of spectrum to inter-vehicle and vehicle-roadside communications. Car manufacturers such as BMW and Daimler-Chrysler started projects for inter-vehicle communications and IEEE is working on standardization for inter-vehicle communications. The importance of VANETs will be increased continuously because VANET will have an important role in realizing Intelligent Transportation Systems [1].

In VANETs, there are many research issues such as Medium Access Control, routing, addressing and so on. Since a VANET is a kind of ad hoc networks, it seems reasonable that applying the existing protocols for ad hoc networks to VANETs to address the above issues. However, the characteristics of the VANETs such as high mobility of vehicles and road-based movement make researchers readdress the above issues.

One of the issues that is worth readdressing is to design a robust and efficient routing protocol. Most of the existing routing protocols designed for mobile ad hoc networks are not suitable for VANETs. For example, AODV [6], the representative of reactive ad hoc routing protocols, incurs large control packet traffic when the path is frequently broken. The reason is that AODV's path rediscovery process triggers network-wide flooding of the control packet. Since VANETs will experience frequent path breakage, AODV is not a good option for VANETs. GPSR [7], another representative ad hoc routing protocol, is also not suitable for VANETs. In VANETs, nodes are placed only on roads. In this situation, GPSR frequently falls into a routing hole, where no neighbor nodes are closer to the destination than the node itself [11]. The frequent occurrence of routing holes will degrade the routing performance severely.

In this paper, we propose a robust and efficient routing protocol for VANETs, especially for the urban environments. The characteristics of VANETs assumed in this paper are as follows. 1) Each node is aware of its location, 2) nodes move along the courses, 3) the map of courses is available to nodes, and 4) vehicles are densely populated. We insist that the urban VANETs can satisfy the above characteristics because more and more cars are equipped with GPS nowadays and the digital road map is also broadly available to the cars.

The proposed routing algorithm, called Macro-level and Micro-level Routing protocol (MMR), addresses and leverages the above network characteristics. MMR models roads as a graph, which employs a two-level routing that consists of the macro level and the micro level routing. At the macro level, a packet is forwarded to an approximate location of the destination using the modeled graph. At the micro level, MMR forwards a packet to an exact location of the destination. With this two-level routing, MMR reduces the control traffic overhead and improves scalability. MMR also introduces a new routing metric that reduces the protocol overhead and path breakage by considering velocities of vehicles. Through simulations, we show that MMR outperforms the existing ad hoc routing protocols such as AODV and GPSR in terms of data delivery ratio, data transfer delay and protocol overhead.

The remainder of the paper is organized as follows. we describe our protocol, MMR in Section II. Section III evaluates the performance of MMR. Section IV concludes the paper.

II. Protocol Specification

We model the set of courses, i.e., the road map, as a graph. A routing process consists of two levels: the macro level and the micro level. A packet is forwarded to the approximate location of the destination at the macro level and to the exact location of the destination at the micro level. Another characteristic of MMR is that the metric used for routing decision is calculated by the information about nodes' mobility. In the following subsections, we describe the details of the proposed protocol.

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.

A. Model Description

Vertices in the graph represent the cross-points of courses and edges do the segments of courses separated by the adjacent cross-points. Note that vertices do not mean the real network entities such as vehicles. The real network entities are called nodes and not modeled in the graph.

We assume that each node has the digital road map of the area of interest and a GPS device such as the navigator system, which most vehicles are equipped with. Moreover, the location of a destination is assumed to be known by a location service (e.g., [10]). This assumption is typical in geographic routing protocols.

B. Two-level Routing

A proposed routing protocol consists of two levels: the macro level and the micro level. At the macro level, a source node finds out an edge on which a destination node located by using the geographic location of the destination node and the digital road map. Then, a source node selects one vertex between two vertices of the edge. The criterion of the selection is the routing metric, which will be explained in Section II.D. We call the selected vertex as a *destination vertex*. A source node writes the *destination vertex* in a data packet header so that intermediate nodes know the *destination vertex*. The data packet is forwarded to the *destination vertex* at the macro level. This means that the data packet is forwarded to a node *near* the *destination vertex* [1]. We say that a node is *near* a certain vertex v when the distance between the node and vertex v is less than a predetermined threshold T.

For vertex-based routing at the macro level, we construct a routing table that maintains a route for each vertex, not for each node. With this routing mechanism, in MMR, the size of the routing table is dependent on the number of vertices, not on the number of nodes, unlike other proactive routing protocols such as [3] and [4]. In vehicle-dense networks such as the urban VANETs, we can expect that the number of vertices is much smaller than the number of nodes. Therefore, our mechanism reduces the control message overhead in the urban VANETs. It improves scalability because the amount of the routing information remains constant irrespective of the number of nodes. The maintenance of a routing table will be detailed in Section II.C.

Once a packet arrives at a node *near* a *destination vertex*, the micro level routing is triggered. At the micro level, MMR performs the greedy forwarding [7] toward a geographic location of the destination node. MMR forwards a packet to the neighbor whose geographic location is the closest to that of the destination node. For the greedy forwarding at the micro level, every node maintains a neighbor table which is a list of geographic locations of neighbors. The maintenance of a neighbor table is detailed in Section II.C. If a *routing hole*, where there is no neighbor who is closer to the destination node than the current node [7], occurs during the greedy forwarding, the *destination vertex* is changed to the other vertex of the edge on which the destination node located and the macro level routing is restarted.

[1]Recall that a vertex indicates the cross-point of the courses.

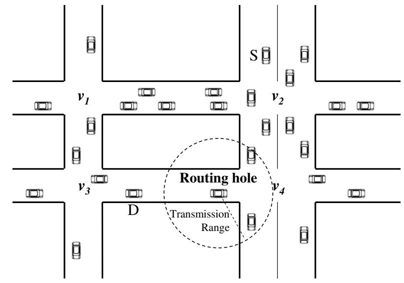


Fig. 1. Forwarding a data packet from S to D. Suppose that S selects v_4 as the *destination vertex*. Then, the packet is forwarded to v_4 at the macro level. Since there's a routing hole between v_4 and D, the *destination vertex* is changed to v_3 and the macro level routing is started again.

Let us illustrate MMR forwarding in Fig. 1. Suppose that node S wishes to send a packet to node D. The first thing to do is to identify the edge on which node D is located. As shown in Fig. 1, D is located on edge $\overline{v_3 v_4}$. S selects one vertex among v_3 and v_4 based on the routing metric. Suppose that S chooses v_4. Then, the packet is forwarded to v_4 thanks to the routing table (the macro level). Once the packet arrives at a node *near* v_4, the greedy forwarding is performed toward the geographic location of D (the micro level). Suppose that a *routing hole* occurs during the greedy forwarding. In that case, the node changes the *destination vertex* from v_4 to v_3 and then restarts the macro level routing.

C. Routing and Neighbor Tables Maintenance

In this section, we present how to advertise routing information and manage a routing and a neighbor tables.

1) Routing Information Advertisement: In MMR, a routing table is constructed by using the Distance Vector mechanism [3], [8]. Every node in the network periodically broadcasts a routing information advertisement (RIA) packet to its neighbor. The RIA packet contains not only its routing table but also its geographic location and velocity. With the exchange of a RIA packet, each node constructs a routing and a neighbor tables, and knows velocities of its neighbors.

A RIA packet consists of the following fields: *node_id, location, velocity, number_of_Information_Element (IE), IE (vertex_id, routing_metric) list*.

The *node_id*, *location* and *velocity* fields represent the identifier, geographic location and velocity of the sender, respectively. The *IE* contains the path information that the sender knows. Note that the path here refers to the path to a vertex, i.e., the path to any node *near* a particular vertex v. The first field of each *IE*, *vertex_id* represents an identifier of the reachable vertex [2] from the sender. The next field is the *routing_metric*. The routing metric is detailed in section II.D.

Basically, *IE* fields are filled with the routing table entries of the sender. Additionally, if the sender is *near* vertex v, it

[2]We say that vertex v is reachable from node n if node n has a path to any node *near* v.

adds one more *IE* whose *vertex_id* is v. This enables other nodes in the network to find the new path to vertex v. The *number_of_the_IEs* corresponds to the number of reachable vertices via the sender.

2) Neighbor and Routing Tables Management: Every node in the network maintains the neighbor table of the node and the routing table. The neighbor table is the list of its own neighbors. A neighbor table entry consists of a *neighbor_id*, its *geographic_location* and *entry_expiration_time*. When a node receives a RIA packet, it checks whether there is an entry for the sender or not. If so, the receiver updates the *geographic_location* field of the relevant entry, otherwise just creates a new entry for the sender in the neighbor table. As stated in Section II.B, the neighbor table is used for greedy forwarding.

The routing table of a node is the list of path information to reachable vertices from the node. Each entry consists of *vertex_id, next_hop*, *routing_metric* and *entry_expiration_time*. The *vertex_id* is a vertex identifier of the vertices which are reachable. The *next_hop* field is an identifier of next hop node of that path. The routing table is updated with a RIA packet. When a node receives a RIA packet, it updates its routing table by comparing the IE fields of the received RIA packet with the corresponding entries in the routing table. Algorithm 1 shows the detailed routing table update procedure.

Algorithm 1 Update the routing table

```
for all IE i do
  v ⇐ the vertex id of IE i
  if there is the entry e whose vertex id is v then
    if i.routing_metric is better than e.routing_metric then
      e.next_hop ⇐ sender of a RIA
      e.routing_metric ⇐ i.routing_metric
    else if e.next_hop = sender of a RIA then
      e.routing_metric ⇐ i.routing_metric
    else
      do nothing
    end if
  else
    create new entry for v and add it to the routing table
  end if
end for
```

Whenever a new routing table entry (i.e., a routing entry for a vertex that has not been maintained) is created, a RIA packet broadcasting is triggered in order to propagate the path information for the new vertex as fast as possible. However, when a routing table entry is removed by expiration timeout, a RIA packet broadcasting is not triggered because every node already knows when the path will be expired with the proposed routing metric. The detailed explanation is presented in the next section.

D. Routing metric

In this section, we propose a new routing metric which reduces the control message overhead and helps to select a path with longer lifetime. We exploit the velocities of nodes for the new routing metric. Based on the velocity information of other nodes, each node calculates three time values. The first one, Link Expiration Time (LET), is the expected remaining time until the link to the current neighbor is expired. Every node calculates the LET for each neighbor whenever receiving a RIA packet from each neighbor. On receipt of the RIA packet, it is straightforward to calculate a LET because the velocity and the geographic location of a neighbor are contained in the RIA packet.

The second time value, Vertex Expiration Time (VET), is the remaining time until a node *near* a particular vertex v moves away from v by longer than T[3]. Only nodes *near* a vertex calculate their VET values. Algorithm 2 shows how to calculate VET.

Algorithm 2 Calculate VET

```
if a node gets closer to a vertex then
  VET ⇐ (T + distance between a node and a vertex) / speed of node
else
  VET ⇐ (T − distance between a node and a vertex) / speed of node
end if
```

The last time value is the Path Expiration Time (PET) which means the expected remaining time until the path to a particular vertex is expired. The PET is calculated based on the above two time values, LET and VET.

We explain how to calculate PETs by the following example in Fig. 2, there is a path from node C to vertex v, i.e., (C, B, A). This path will be broken either if one of links ($\overline{CB}$ or $\overline{BA}$) has broken or the distance between node A and vertex v exceeds threshold T. The former case corresponds to an expiration of the LET between A and B or the LET between B and C. The latter case corresponds to an expiration of the VET of A for v. Therefore, the minimum of the LET between B and C, the LET between A and B, and the VET will be the PET of C for vertex v. In reality, the PET of C is calculated by the minimum of the LET between B and C and the PET of B for vertex v because the minimum of the LET between A and B and the VET of A for vertex v corresponds to the PET of B for vertex v. Algorithm 3 shows a formal description of calculating PET. Note that each node calculates a PET for a particular path in a distributed fashion on receipt of a RIA packet.

A path with a larger PET has longer lifetime which will provide path stability. However, the routing metric based only on the PET can cause the path to be very long (many hop counts). A long path can suffer from large delay and have more chance of packet drop due to collisions or channel error. To resolve the situation, the PET normalized by the hop counts (NPET) is used as the routing metric in MMR. The path with

[3]Recall that T is the predetermined threshold. If a distance between a node and a particular vertex v is less than T, we say that the node is *near* v

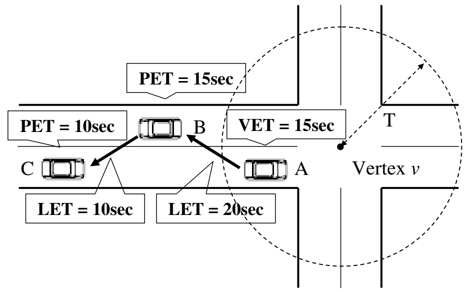


Fig. 2. An example of calculating PET. The circle represents threshold T.

Algorithm 3 Calculate PET

- LET_{n_1,n_2} = LET between node n_1 and node n_2
- $VET_{n,v}$ = VET between node n and vertex v
- $PET_{n,v}$ = PET for vertex v at node n
- $(n_1, n_2, n_3, n_4, ..., n_k)$ = routing path for vertex v where n_k is *near* v

$$PET_{n_i,v} = \begin{cases} min(LET_{n_i,n_{i+1}}, PET_{n_{i+1},v}) & \textbf{if } i < k \\ VET_{n_i,v} & \textbf{if } i = k \end{cases}$$

the larger NPET is preferred over the path with the smaller NPET. With the new routing metric, MMR can balance the path stability and the routing efficiency. Assuming the hop counts of the candidate paths are similar, MMR can reduce the protocol overhead due to path breakage.

E. Mobility Information Error

There may be some deviation between the actual estimated-velocity and the velocity known to the other nodes due to imprecise velocity measurements or speed changes of vehicles. This incurs an error on time values such as LET and PET, resulting in the performance degradation of MMR. In order to alleviate this problem, we introduce a simple mechanism. The idea is that every node scales LET and PET down by a factor S. Upon the assumption that the maximum velocity error ratio α can be obtained, the scaling factor S is set to $\{1-\alpha\}$ where $0 \leq \alpha < 1$. The intuition behind our idea is that the harm of low-estimated velocity error is much more severe than that of over-estimated velocity error.

III. Performance Evaluation

In this section, we evaluate the performance of MMR using NS-2 [12]. We evaluated the performance of MMR with GPSR [7], AODV [6] and DSDV [3]. GPSR, AODV and DSDV are the representative geographic, reactive and proactive routing protocols, respectively. We excluded the results of the DSDV since the routing performance of DSDV is much poorer than others.

A. Simulation Setup

IEEE 802.11 is used for the network interface. The bandwidth is 1 Mbps and the transmission range is 250 m. In a 2000 m x 2000 m area, 10 straight courses are placed at random. 200 nodes are distributed on the courses and move along the courses. The speed is chosen randomly between 1 m/s and the maximum speed (5∼20 m/s). For performance evaluation, 10 pairs of sources and destinations are selected randomly and the sources send 64 bytes packets over UDP with the interval of one second. The interval of broadcasting routing messages is set to one second in both MMR and GPSR. Threshold T is set to 125 m. The performance metrics are averaged over ten times, each for 150 seconds.

B. Data Delivery Ratio

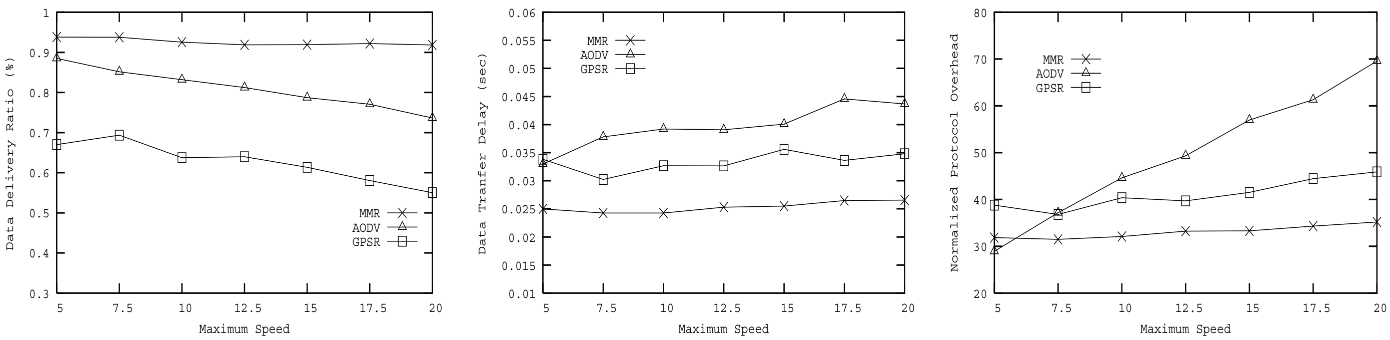
Fig. 3(a) shows the packet delivery ratio with respect to the varying maximum speed. The delivery ratio of GPSR is worst among three routing protocols. This is because GPSR suffers from frequent routing holes at the node placement only on courses [11]. The delivery ratio of AODV decreases as the maximum speed increases. Since AODV is a reactive routing protocol, it performs route rediscovery whenever the packet drops due to route breakage. Compared to AODV and GPSR, the delivery ratio of MMR is higher than 90% at all mobility levels. As mentioned in Section II.D, MMR considers stability in the route selection using PET values. Moreover, MMR can update a new route before the current route is broken because it is a proactive routing protocol.

C. Data Transfer Delay

As shown in Fig. 3(b), the average delay of MMR is the smallest among three routing protocols. AODV needs the route acquire process (RREQ flooding, RREP unicasting) whenever the current path is broken. This causes AODV to be worst in terms of the delay. GPSR suffers from routing holes due to node placement only on courses and the fallback solution (i.e., face routing in [7]) makes the route much longer, which causes the delay to be larger. On the other hand, MMR forwards a data packet using proactively maintained routes with stability.

D. Protocol Overhead

In this section, we compare how many control packets (i.e., RIA in MMR, HELLO in GPSR, RREQ, RREP and RERR in AODV) are transmitted. For fair comparisons, we have normalized the protocol overhead by the number of the data packets which are successfully delivered to destinations. See Fig. 3(c) for the normalized overhead versus maximum speed. The normalized overhead of MMR is less than that of GPSR at all maximum speed simulated. As expected, the normalized overhead of AODV is dependent on maximum speed. Since the route discovery of AODV normally causes network-wide flooding, AODV has the large routing overhead. Especially, as node mobility increases, the routing overhead increases accordingly because link breakages cause route rediscoveries. That is why the normalized overhead of AODV increases as the maximum speed increases.



(a) Delivery ratio

(b) Delay

(c) Normalized routing overhead

Fig. 3. Performance comparisons between MMR, GPSR, and AODV with respect to mobility of nodes.

E. Impact of Mobility Information Error

In this section, we show the performance of MMR when there are some errors on measured velocity. All simulation setups are the same as error-free scenarios except that there exist some errors on the estimated velocities of vehicles. Given the maximum error ratio E, we generate an artificial error which is uniformly selected on range [– E, + E]. Fig. 4 shows the data delivery ratio as the maximum speed increases when there is an error on velocity estimation. In both case of E = 10% and E = 20%, MMR still achieves the higher data delivery ratio than AODV and GPSR although there is a little performance degradation compared with the error-free case of MMR.

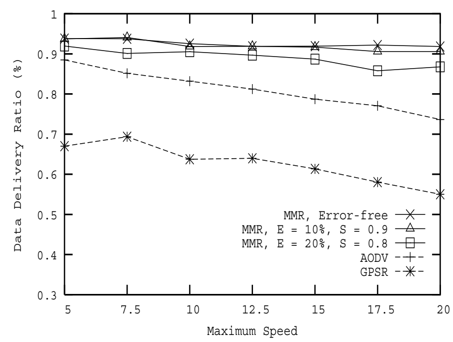


Fig. 4. Data delivery ratio. (Mobility prediction error)

IV. CONCLUSION

In this paper, we proposed a robust and efficient routing protocol for urban VANETs. The proposed protocol, MMR, uses two-level routing in order to reduce the control message overhead. A data packet is forwarded to the approximate location (vertex) of the destination at the macro level and then is forwarded to the exact location of the destination at the micro level. In dense networks such as urban VANETs, the two-level routing of MMR will improve scalability by reducing protocol overhead. MMR also proposes the new routing metric that balances the lifetime of a path and the hop count by exploiting the velocities of other nodes. It improves the routing performance as the mobility increases by reducing the path breakage. Through extensive simulations, we show that MMR outperforms AODV and GPSR.

ACKNOWLEDGEMENT

This work was supported in part by the Brain Korea 21 project of the Ministry of Education Korea, and in part by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. F01-2005-000-10040-0). The ICT at Seoul National University provides research facilities for this study.

REFERENCES

[1] http://www.its.dot.gov/

[2] I. Chlamtac, M. Coti, J J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks 1(2003)*, pp. 13 64, Elsevier, 2003

[3] C. Perkins and P. Bhagwat. "Highly dynamic destination-sequenced distancevector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM 94*, 1994.

[4] T. Clausen, P. Jaqcuet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol (OLSR)," RFC 3626, October 2003.

[5] D. B Johnson and D. A Maltz. "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[6] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-demand Distance Vector (AODV) Routing," *IETF Experimental RFC*, MANET working group, RFC 3561, July 2003

[7] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. ACM MOBICOM 2000*, August 2000.

[8] S. Keshav, *An Engineering Approach to Computer Networking - ATM Networks, the Internet, and the Telephone Network,* Addison-Wesley, Reading, MA, 1997.

[9] F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger, "Geometric Routing: Of Theory and Practice," in *Proc. of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, 2003.

[10] J. Li, J. Jannotti, D. DeCouto, D. Karger and R. Morris, "A Scalable Location Service for Geographic Ad-hoc Routing," in *Proc. ACM MOBICOM 2000*, August 2000.

[11] Hojin Lee, Youndo Lee, Taekyoung Kwon and Yanghee Choi, "Virtual Vertex Routing (VVR) for Course-Based Vehicular Ad Hoc Networks," in *Proc. IEEE WCNC 2007*, March 2007.

[12] "The Network Simulator - NS-2," http://www.isi.edu/nsnam/ns/, Online link.

1930-529X/07/$25.00 © 2007 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2007 proceedings.