Contention-based Adaptive Position Update for Intermittently Connected VANETs

Qiang Zhang[1], Hongli Zhang[1], Xiaojiang Du[2], Zhigang Zhou[1], Shen Su[1], Rui Jin[1]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
zhanghongli@hit.edu.cn, {zhangqiang, zhouzhigang, sushen, jinrui}@pact518.hit.edu.cn
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
dux@temple.edu

***Abstract*—Position information of nodes in vehicular ad hoc networks (VANETs) plays a key role in geographic routing. A sender or intermediate node employs position information of its neighbors and destination node to make routing decision. Under a greedy forwarding algorithm, the neighboring node closest to the destination node is selected as the next hop. Hence, it is critical in geographic routing to ensure that the selected next hop has a better position than other neighboring nodes. Position information is usually propagated to local nodes through periodical beaconing. In most geographic routing protocols, each node broadcasts beacons in a fixed interval, but this method can not always achieve both position accuracy and low overhead. In this paper, we propose a contention-based adaptive position update (CAPU) scheme for intermittently connected VANETs. CAPU concentrates on the position accuracy of the next hop when data transmission happens. If the position deviation of the next hop is greater than the permitted deviation range, the next hop updates its position. A special next hop time-out approach is proposed to find and delete the unreachable next hop as soon as possible. CAPU can find key nodes in local topology for greedy forwarding and intermittent connectivity. In addition, contention beacons broadcasted by key nodes maintain the local topology. Experimental results show that the proposed approach provides key position information for routing decision and exhibits better routing performance with acceptable overhead.**

Keywords—VANETs; geographic routing; position update; adaptive beaconing

I. INTRODUCTION

Vehicular ad hoc networks (VANETs) [1] have emerged as a popular wireless network technology that provides helps to the development of intelligent transportation system (ITS). VANETs mainly consist of vehicles equipped with wireless interfaces and some fixed equipments on the roadside. Communication among vehicles and between vehicles and roadside units is the basis of many new applications and potential services in VANETs. In recent years, a number of routing technologies have been proposed to achieve efficient communications. Currently geographic routing [2-4] is considered as a good method for VANETs. Unlike other routing protocols (AODV [5] and OLSR [6]) that need to maintain routing information, a geographic routing protocol is more flexible on selecting the next hop using node position information.

Beaconing is the basis of neighbor discovery, position update, cooperative awareness and data dissemination in VANETs. In most geographic routing protocols (e.g., [7-9]), each node broadcasts beacons in a fixed interval to provide its own position and speed information. However, periodical beaconing has several drawbacks. First, when the speed of nodes changes fast, beaconing in a fixed long interval cannot always provide accurate information to form good local topology. If beacons are broadcast in a fixed short interval, it is very likely that some slow nodes cause lots of communication overheads due to the relative stable local topology. Second, redundant periodical beaconing will contend for channel with data packets. Hence, MAC layer collisions may happen frequently and cause network congestion [10][11]. Then retransmission will be triggered, and the end-to-end delay of data packets increases. Third, when no data packet is transmitted in a region, beaconing in a fixed short interval in this region is not useful for data transmission but causes overhead. Hence, adaptive position update is necessary to achieve both position accuracy and low overhead of beaconing for geographic routing in VANETs.

In geographic routing, nodes use position information to make routing decision. If a sender or intermediate node has one or more neighboring nodes closer to the destination node than itself, the neighboring node closest to the destination node is selected as the next hop by greedy forwarding mode (GFM). Otherwise, the sender or intermediate node selects the next hop by perimeter forwarding mode. However, periodical beaconing can not always guarantee the correct selection of the next hop in the above way. Especially, for two cases it is very possible that the neighbor information is false or outdated for routing decision. In the first case a new neighbor appears, and it becomes the optimal selection of the next hop. But the position information of the new neighbor may have not been updated by periodical beaconing for a period. Therefore, the optimal selection of the next hop is not achieved. In the second case, the next hop just moves out of the communication range and becomes the unreachable next hop. The false selection of the next hop happens due to the outdated neighbor information is still stored in neighbor table. Then the data transmission fails and retransmission is triggered. The routing performance is decreased by the unreachable next hop. Moreover, in other cases the outdated neighbor information can also lead to the wrong selection of the next hop. For example, a fast moving node arrives at the advantageous position in the neighborhood in a little time. However the corresponding position information has not been updated owing to the fixed beacon interval. Then the optimal selection of the next hop still cannot be achieved. Hence, how to guarantee the node that has the most advantageous position can be selected as the next hop is critical.

In this paper, we propose a contention-based adaptive position update (CAPU) scheme for intermittently connected VANETs. In CAPU, two mechanisms are proposed to update positions of the key nodes so that geographic routing performance can be improved for intermittently connected VANETs. The first mechanism is adaptive position update of the next hop. As the next hop is the important role for forwarding data packets, the position accuracy of the next hop needs to be guaranteed. In CAPU, position update of the next hop is triggered adaptively based on its mobility and data transmission. When the position deviation of the next hop is greater than the permitted deviation range, the next hop needs to update its position. In addition, due to the characteristic of the intermittently connected network, nodes can move out of the communication range of other nodes frequently. So a special next hop time-out interval based on data transmission is proposed to solve the problem of the unreachable next hop. The second mechanism is contention beacon. In intermittently connected network, the actual connected network can appears the unconnected network due to false or outdated position information. So we propose an algorithm to find the key nodes for greedy forwarding and intermittent connectivity, and contention beacons are broadcast to update their positions.

The rest of the paper is organized as follows. Section II introduces related work. Section III describes the detailed design of our proposed scheme. Section IV presents the performance evaluation, and Section V concludes the paper.

II. Related Work

APU (Adaptive Position Update) [12] is proposed to adapt the beacon update interval to the node mobility and the traffic load. In APU, the MP (Mobility Prediction) rule is used by every node in the network to calculate the location estimate error. If the location estimate error of a node is greater than a given threshold, the node broadcasts a beacon right now. In addition, when a node overhears a data packet, the ODL (On Demand Learning) rule is employed to judge whether the sender of the data packet is a new neighbor. If yes, the monitor broadcasts a beacon. However, this work does not propose a new method to solve the problem of the unreachable next hop, and just hands the problem to MAC layer.

The research in [13] evaluated three methods of beaconing. The three methods are distance-based adaptive beacon interval, speed-based adaptive beacon interval, reactive beaconing. In the distance-based method, a beacon is broadcast whenever a node has moved a “beacon distance” since its last transmission. A node deletes an entry if it has moved more than *k*-times the “beacon distance”. In the speed-based method, the beacon interval is correlated to the speed a node is moving at. And the beacon interval can be determined using a continuous function of the nodes’ speed within a predefined time range. In the reactive beaconing method, only when a node has data packets to transmit, it solicits beacons from its neighbors by transmitting a beacon request packet. Each node overhearing this request replies with a beacon to announce its position. The distance-based and speed-based adaptive beacon intervals trigger beaconing without considering whether there is data transmission in the neighborhood. Although the reactive beaconing triggers beaconing on demand, all neighbors need to replies with their beacons. So the end-to-end delay is increased due to waiting for the reply beacons.

CAR (Connectivity-Aware Routing) [14] changes the beacon interval according to the number of neighboring nodes. The more neighboring nodes are, the lower frequency of beaconing is. In addition, the researchers [15] present a distance-based method to determine the beacon interval. When the difference between the predicted position and the actual position is greater than a threshold value, the beacon is broadcast. But these works do not solve the problem of the unreachable next hop.

III. The Contention-based Adaptive Position Update Scheme

A. Basic Beacon and Data Packet Format

We define basic beacon as a kind of short message including the id, position and speed of the sender. And we consider that hello message is another kind of short message which does not contain position and speed information but the id. So, a node broadcasts a basic beacon for the purpose of informing its position to its neighboring nodes. If a node receives a basic beacon from another node, it will list the sender of the beacon and update the corresponding position and speed in its neighbor table. In our proposed approach, basic beacon is broadcast periodically by each node to maintain the basic local topology in VANETs. And each node uses a basic time-out interval to remove outdated neighbors to maintain neighbor table if it has no data to transmit. So, after receiving a basic beacon, the receiver updates the time stamp of the corresponding entry in neighbor table. If a node has data to transmit, it uses a special time-out interval for the next hop in neighbor table. And we demonstrate the special time-out interval in following part.

We add new fields in the traditional data packet format of geographic routing in order to implement contention-based adaptive position update. The new data packet consists of data, destination position, next hop id, predicted position of the next hop and forwarding mode. We consider that nodes employ position prediction mechanism [4] to determine the next hop by the greedy forwarding mode. This means that, when a node determines the selection of the next hop, it needs to calculate the predicted position of each node in its neighbor table by virtue of the corresponding speed, position and time difference. Then, the neighboring node whose predicted position is the closest to the destination is selected as the next hop. In addition when a node overhears a data packet, it is able to obtain the corresponding forwarding mode from the data packet.

B. Adaptive Position Update of the Next Hop

Two kinds of short messages (i.e., the basic beacon and hello message) are used to update the position of the next hop. The next hop position update is triggered adaptively based on its mobility and data transmission. The predicted position of the next hop node i contained in data packet is denoted as P_i' calculated by the last hop node. The actual position of the next hop node i is denoted as P_i, which is known by node i. We design Algorithm 1 to implement the adaptive position update of the next hop based on mobility and data transmission.

Algorithm 1

INPUT: *data packet,* P_i

Forward the data packet.
Calculate the position deviation $D(P_i, P_i')$ between the actual and predicted positions.
if $D(P_i, P_i') > \alpha$
    Broadcast a basic beacon.
else
    Broadcast a hello message.

When the next hop node i receives a data packet, firstly it forwards the data packet. Then, the next hop calculates its own position deviation denoted as $D(P_i, P_i')$ between the predicted position and the current actual position. We define a threshold α which indicates the permitted deviation range. If the position deviation of the next hop is greater than α, it means that the node has changed its speed and the information in its last beacon is outdated. Because the optimal selection of the next hop is based on the correct local topology, the position inaccuracy of the next hop may cause the wrong routing decision. Therefore, in this situation the next hop broadcasts a basic beacon to update its position and speed information in order to maintain the correct local topology. Otherwise, it shows that the information in the last beacon of the next hop is not outdated. Hence, when the condition $D(P_i, P_i') \leq \alpha$ holds, the next hop does not need to update its position and speed information, and just broadcasts a hello message to save overhead.

In fact, we have to face and solve the problem that the next hop is out of communication range (we also call it the unreachable next hop in this paper). When some data packets are forwarded to the unreachable next hop, the routing performance will be decreased. So, we need to delete the neighbor information of the unreachable next hop as soon as we find that the next hop node is unreachable. And we try to use acceptable cost to solve the problem. We demonstrate our specific method as follows. If a node has data to transmit, the data transmitter will use two time-out intervals to maintain neighbor information. One is the special time-out intervals T_s for the neighbor information of the next hop. The other is the basic time-out intervals T_b for neighbors except the next hop in the transmitter's neighbor table. After a data transmitter has forwarded a data packet to the next hop, the transmitter starts a timer with the special time-out interval to wait a short message (basic beacon or hello message) from the next hop. After the timer with time-out interval T_s for the next hop (this special timer is referred to as ST) begins, if the data transmitter receives a basic beacon or hello message from the next hop within the time period of T_s, the transmitter updates the neighbor information by the received basic beacon or just retains the neighbor information by the received hello message. Otherwise, the data transmitter deletes the neighbor information of the next hop in neighbor table. In addition, after the data transmitter receives a basic beacon or hello message from the next hop, it triggers a new timer with the basic time-out interval T_b (this basic timer is referred to as BT) for the next hop rather than a ST. In other words, transmitting a data packet makes the transmitter to trigger a ST for the next hop of the data packet, and receiving an update short message (i.e., basic beacon or hello message) from the next hop before time-out triggers a new BT. The special time-out interval T_s is calculated based on the transmission delay, propagation delay and processing delay. So, T_s is smaller than T_b.

We let each node keep a next hop list (NHL), and the NHL records the nodes which have been selected as the next hop and attached with the special time-out interval T_s. Specifically, when a node has data to forward, we use Algorithm 2 to solve the problem of the unreachable next hop.

Algorithm 2

INPUT: *data packet, nh(next hop)*

Forward the data packet;
if (*nh* is not in NHL)
    if ($\text{time}_{\text{current}} - \text{time}_{\text{nh}} < T_b$)
        $\text{time}_{\text{nh}} = \text{time}_{\text{current}}$;
        Add *nh* in NHL;
    else
        Delete *nh* in neighbor table;
else
    if ($\text{time}_{\text{current}} - \text{time}_{\text{nh}} > T_s$)
        Delete *nh* in neighbor table;
        Delete *nh* in NHL;

When a node has data to transmit and has selected the next hop, firstly it forwards the data packet. Then, the data transmitter determines whether the next hop is in NHL. If not, the data transmitter calculates the difference between the current time and the time stamp of the next hop in the neighbor table. If the time difference is smaller than T_b, the time stamp of the next hop is updated by the current time and the next hop is added in NHL, which means the data transmitter has triggered a timer with T_s for the next hop in neighbor table. If the time difference is greater than T_b, it means that the neighbor information of the next hop is outdated, and the next hop is deleted in neighbor table. If the next hop is in NHL, the data transmitter uses T_s to judge whether the next hop is outdated. If the next hop is outdated, it is deleted in neighbor table and NHL.

We discuss the effect of Algorithm 2 as follows. When the next hop is unreachable and is selected for the first time, if the next hop is not outdated for T_b, the data transmitter forwards the first data packet to the next hop and updates the time stamp T_0 of the next hop in neighbor table by the current time T_1. Then, the next hop is added in NHL. Obviously, the first data packet cannot be received by the unreachable next hop, and the false neighbor information of the next hop is still in neighbor table this moment. We assume that $T_1 + T_s < T_0 + T_b$. When the second data packet has the same next hop with the first data packet, the data transmitter will find that the next hop is in NHL. If the current time is in the range of $(T_1, T_1 + T_s]$, the second data packet will be forwarded directly. And the second data packet also can not arrive at the unreachable next hop (i.e., the following data packets also select the unreachable next hop if the corresponding times is in the range of $(T_1, T_1 + T_s])$. If the sending time of the second data packet is in the range of $(T_1 + T_s, T_0 + T_b)$, the unreachable next hop will be deleted in neighbor table. And the following packets can be forwarded to another neighbor rather than the unreachable next hop of the first data packet when the corresponding times is in the range of $(T_1 + T_s, T_0 + T_b)$, although the second data packet is not forwarded successfully. If the sending time of the second data packet is in the range of $[T_0 + T_b, \infty)$, the data transmitter will delete the next hop of the first data packet in the process of the selection of the next hop due to the corresponding neighbor information is outdated for T_b. In addition, when a node finds the outdated neighbor in neighbor table using T_b, the same neighbor in NHL is also deleted.

When the next hop is reachable, after the next hop forwards the received data packet, it uses Algorithm 1 to broadcast a short message (i.e., the basic beacon or hello message). The short message will be received by the data transmitter (the last hop). The data transmitter judges whether the sender of the short message is in NHL. If yes, the data transmitter deletes the corresponding entry in NHL. And the data transmitter updates the neighbor information by the received basic beacon or just retains the neighbor information by the received hello message.

C. The Contention Beacon

When the density of nodes is rather low or the node mobility is complex, VANETs will face the situation of intermittent connectivity. In this situation, some neighbors of the nodes will be the important role to transform an unconnected network to the connected one. Hence, updating the positions of the key nodes is very necessary. Actually, for each node if the corresponding local topology is accurate, the whole network topology is accurate. However, if some local topologies are false, they are able to affect the whole network topology. Evenly, the actual connected network will appears the unconnected network due to false or outdated position information. Beaconing periodically in a short interval can guarantee the accurate local topology. But when there is no data transmission in the network, it is a little wasteful. Moreover, the redundant beaconing may collide with data packets. To solve the problem, we proposes contention beacon to update the positions of the key nodes rather than all neighbors. So, when a node overhears a data packet forwarded in GFM (data packet contains the forwarding mode), it should judge whether it is a key node. If yes, the key node broadcasts a contention beacon. Algorithm 3 gives the method for determining whether a node is a key node.

Algorithm 3

INPUT: *data packet,* P_d *(destination position),*
P'_{nh} *(predicted position of the next hop)*

if (*forwarding mode* is GFM)
  if (i is not *next hop*) // *i is the current node*
    $\text{dis}_{\text{current}} = D(P_i, P_d)$;
    $\text{dis}_{\text{nexthop}} = D(P'_{nh}, P_d)$;
    if ($\text{dis}_{\text{current}} < \text{dis}_{\text{nexthop}}$)
      Broadcast a contention beacon;

Contention beacon has the same message format with basic beacon. We call basic beacon as contention beacon here because the sender of basic beacon is a key node for greedy forwarding and intermittent connectivity. As shown in figure 1, the red node represents the position of node A_1 stored in neighbor table of node S, A_2, A_3. And each blue node represents its actual position. In the traditional way, S selects A_2 as the next hop routing. And A_2 will find that no node can be selected as the next hop by GFM. Then A_2 will select A_3 as the next hop by perimeter forwarding mode (PFM). At last, A_2 will select A_1 as the next hop by PFM. And the following data packets also perform the above procedure till A_1 broadcasts a basic beacon at the next period. However, Algorithm 3 can judge that A_1 is a key node as soon as A_1 overhears a data packet from S. Then A_1 broadcasts a basic beacon to make S and other neighboring nodes to update its position. Subsequently the following data packets from S will be forwarded to A_1 directly. Consequently, the contention beacon updates the positions of key nodes in time.

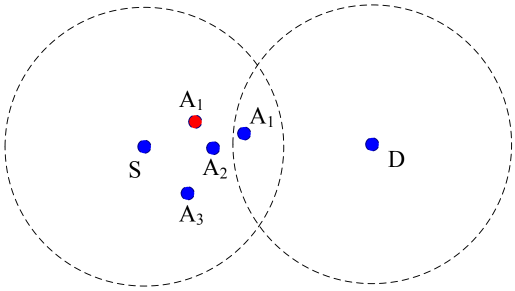


Fig. 1: Position update of a key node

IV. PERFORMANCE EVALUATION

A. Experimental Setting

We use NS-2 for simulation. The communication range of a node is 250 *m*. A total of 30 nodes are deployed in a region of size 1500 *m* x 1500 *m*. Two-ray ground model is used as the radio model. IEEE 802.11 DCF is selected as the medium access control (MAC) layer protocol. The MAC layer data rate is set to 2Mbps. We use VanetMobiSim [16] to generate movements of nodes. The created network is intermittently connected. The node speed is in the range of [5, 20] *m/s*. And we consider the range of [5, 10] *m/s* as the scenario of low speed. The range of [10, 15] *m/s* and the range of [15, 20] *m/s* are respectively corresponding to medium speed and high speed. The simulation time is 220*s*. We randomly select three source-destination pairs in each scenario. The generated constant bit rate (CBR) is in the range [5, 25] *packets/s*. The data flow lasts from the 120th second to the 220th second. The data packet size is 512 bytes. The queue length of each node is set to 50 packets.

We do not implement the overhearing data packet program in the data link layer of NS-2. Instead, we design an equivalent program to perform overhearing data packet in CAPU. Specifically, after a node selects the next hop by GFM and forwards the data packet, it broadcasts a special message including the destination position, next hop id, the current predicted position of next hop. When a node receives the special message, it knows that a data packet has been forwarded. Similarly, an equivalent program is designed to perform overhearing data packet in APU. However, the special message in APU only contains the id and position of the sender. We do not calculate the special messages overhead when discussing the total beacon overhead. We compare CAPU with APU in terms of packet delivery ratio, average end-to-end delay and beacon overhead for intermittently connected VANETs.

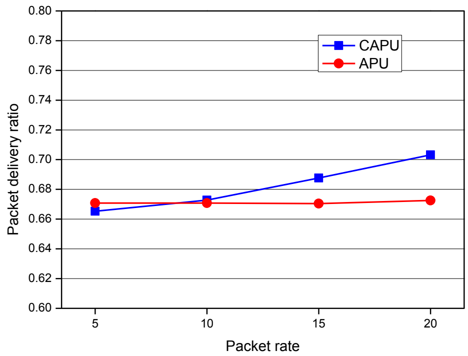
As discussed in [17], the threshold α is chosen as $10m$. T_s is set to $0.1s$, and T_b is $4s$. The basic beacon interval in CAPU is $2s$. It is known that APU adapts the beacon update intervals to the mobility of the nodes and traffic load in the neighborhood of the nodes. And APU needs an initialization phase as demonstrated in [12]. The initialization phase of APU employs periodical beaconing to develop a neighbor list. When data transmission starts, the initialization phase is stopped in APU. The beacon interval in initialization phase of APU is also set to $2s$. And the initialization phase runs from the beginning of simulation to the 120th second.

B. Experimental Results

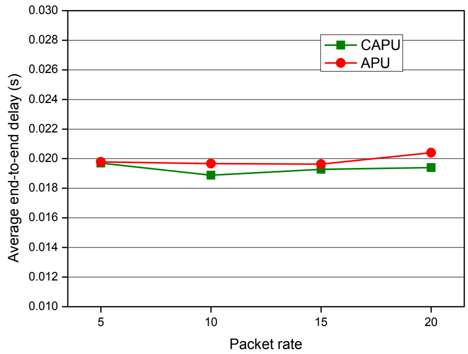
In our experiments, we evaluate the performance of CAPU and APU for intermittently connected VANETs. As shown in Fig. 2, the packet delivery ratio of CAPU outperforms that of APU when the node speed is high. This is because CAPU deletes the neighbor information of the unreachable next hop in time. So the data packets can be forwarded successfully to the reachable next hop. As analyzed in section III, CAPU tries to find and delete the unreachable next hop as soon as possible. In addition, when the next hop is unreachable, MAC layer cannot receive the CTS from the next hop. Then MAC layer retransmits the RTS. When several RTS retransmissions fail, the data packet is dropped. For other scenarios, the packet delivery ratio of CAPU is close to that of APU. In addition, Fig. 2 shows that the average end-to-end delay of CAPU is a little bit smaller than that of APU in most scenarios. This means that the method of contention beacon helps make the optimal selection of the next hop. Table I shows the beacon overhead for different speeds during data transmission. The amount of beacons in CAPU is greater than that of APU. Beacons of CAPU mainly are produced by periodical basic beaconing. As APU does not use periodical beaconing in the process of data transmission, the amount of beacon is smaller.

TABLE I. TOTAL BEACON OVERHEAD

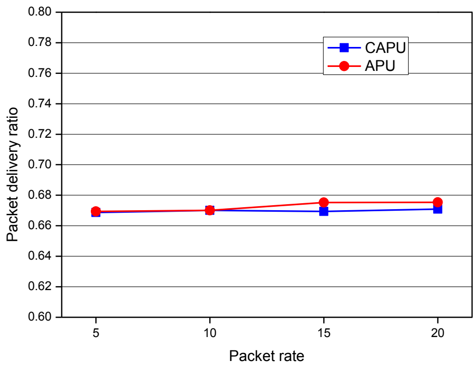
Speed	CAPU	APU
low	1651	376
medium	1565	471
high	1552	521



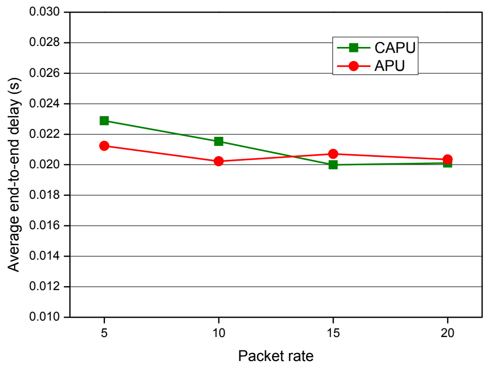
(a) Packet delivery ratio versus packet rate for low speed



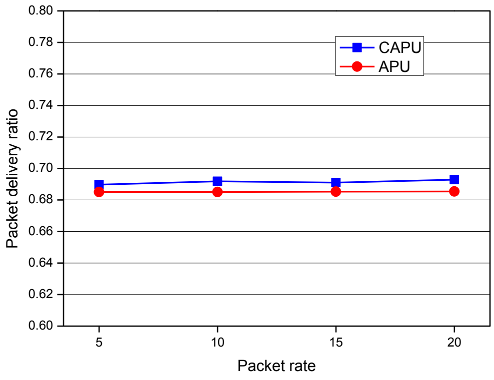
(b) Average end-to-end delay versus packet rate for low speed



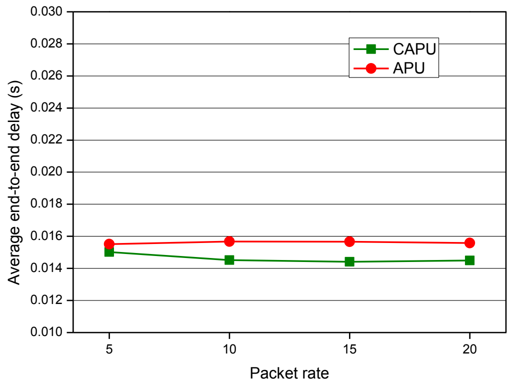
(c) Packet delivery ratio versus packet rate for medium speed



(d) Average end-to-end delay versus packet rate for medium speed



(e) Packet delivery ratio versus packet rate for high speed



(f) Average end-to-end delay versus packet rate for high speed

Fig. 2: Performance evaluation in terms of packet delivery and average end-to-end delay

V. CONCLUSION

In this paper, we presented a contention-based adaptive position update (CAPU) scheme for intermittently connected VANETS to improve geographic routing performance. CAPU concentrates on the position accuracy of the next hop when data transmission happens. If the position deviation of the next hop is greater than the permitted range, the next hop updates its position. A special next hop time-out interval based on data transmission is proposed to find and delete the unreachable next hop as soon as possible. CAPU can find the key node in local topology for greedy forwarding and intermittent connectivity. Key nodes broadcast contention beacons to make the last hop maintain the real local topology, and hence the optimal selection of the next hop is guaranteed. Our experimental results demonstrated that CAPU improves the performance of geographic routing for intermittently connected VANETs.

ACKNOWLEDGMENT

This research was partially supported by the National Basic Research Program of China (973 Program) under grant No. 2011CB302605, the National High Technology Research and Development Program of China (863 Program) under grant No. 2011AA010705, the National Science Foundation of China under grants No. 61100188 and No. 61173144, the CNNIC under grant No. K201211043, as well as the US National Science Foundation under grant CNS-1065444.

REFERENCES

[1] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164-171, Jun. 2008.

[2] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing forWireless Networks," in *Proc. 6th Annu. Int .Conf. Mobile Computing and Networking(MOBICOM)*, Boston, MA, USA, 2000, pp. 243-254.

[3] J. Zhao and G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," *IEEE Trans. Veh. Technol.*, vol. 57, pp. 1910-1922, May 2008.

[4] K. C. Lee, U. Lee and M. Gerla, "Geo-opportunistic routing for vehicular networks," *IEEE Commun. Mag.*, vol. 48, no. 5, pp.164-170, May 2010.

[5] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," in *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps.*, New Orleans, LA, USA, 1999, pp.90 -100.

[6] P. Jacquet, et al. "Optimized link state routing protocol for ad hoc networks," in *Proc.16th International Multi Topic Conference(INMIC)*, 2001, pp. 62-68.

[7] Shafiee, Kaveh and V. Leung. "Connectivity-aware minimum-delay geographic routing with vehicle tracking in VANETs," *Ad Hoc Networks*, vol. 9, no. 2, pp. 131-141, Mar. 2011.

[8] Jerbi, Moez, et al. "Towards efficient geographic routing in urban vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 58, pp. 5048-5059, Nov. 2009.

[9] Saleet, Hanan, et al. "Intersection-based geographical routing protocol for VANETs: a proposal and analysis," *IEEE Trans. Veh. Technol.*, vol. 60, pp. 4560-4574, Nov. 2011.

[10] van Eenennaam, Martijn, et al. "Exploring the solution space of beaconing in VANETs," in *IEEE Vehicular Networking Conference(VNC)*, Tokyo, Japan, 2009, pp. 1-8.

[11] Lloret, Jaime, et al. "Group-based protocol and mobility model for vanets to offer internet access," *Journal of Network and Computer Applications*, vol. 36, no. 3, pp. 1027-1038, May 2013.

[12] Q. Chen, S. S. Kanhere and M. Hassan. "Adaptive Position Update for Geographic Routing in Mobile Ad-hoc Networks," *IEEE Transaction on Mobile Computing*, vol. 12, pp. 489-501, Mar. 2013.

[13] Heissenbüttel, Marc, et al. "Evaluating the limitations of and alternatives in beaconing," *Ad Hoc Networks*, vol. 5, no. 5, pp. 558-578, Jul. 2007.

[14] V. Naumov, and T. R. Gross, "Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks," in *Proc. 26th Int. Con. Computer Communications (INFOCOM'07)*, Anchorage, Alaska, USA, 2007, pp. 1919 -1927.

[15] A. Boukerche, C. Rezende and R. W. Pazzi, "Improving Neighbor Localization in Vehicular Ad Hoc Networks to Avoid Overhead from Periodic Messages," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM'09)*, Honolulu, HI, USA, 2009, pp. 1-6.

[16] Härri J, Filali F, Bonnet C and Fiore M, "VanetMobiSim: generating realistic mobility patterns for vanets," in *Proc. 3th Int. Wksp. Vehicular Ad Hoc Networks(VANET'06)*, New York, NY, USA, 2006, pp. 96–97.

[17] Schmidt, R., et al. "Exploration of adaptive beaconing for efficient intervehicle safety communication," *IEEE NETWORK*, vol. 24, no. 1, pp. 14-19, Jan.-Feb. 2010.