Adaptive Position Update in Geographic Routing

Quan Jun Chen[*], Salil S. Kanhere[*], Mahbub Hassan[*†], Kun-Chan Lan[†]

[*]

School of Computer Science and Engineering,
The University of New South Wales, Sydney, Australia,
Email: {quanc, salilk, mahbub}@cse.unsw.edu.au

[†]

National ICT Australia, Sydney, Australia,
E-mail: Kun-Chan.Lan@nicta.com.au

UNSW-CSE-TR-0519

October 2005

THE UNIVERSITY OF
NEW SOUTH WALES

SYDNEY • AUSTRALIA

Abstract

In geographic routing, nodes need to maintain up-to-date positions of their immediate neighbours for making effective forwarding decisions. Periodic broadcasting of beacon packets that contain the geographic location coordinates of the nodes is a popular method used by most geographic routing protocols to maintain neighbour positions. We contend that periodic beaconing regardless of network mobility and traffic pattern does not make optimal ulilisation of the wireless medium and node energy. For example, if the beacon interval is too small compared to the rate at which a node changes its current position, periodic beaconing will create many *redundant* position updates. Similarly, when only a few nodes in a large network are involved in data forwarding, resources spent by all other nodes in maintaining their neighbour positions are greatly wasted. To address these problems, we propose the *A*daptive Position Update (APU) strategy for geographic routing. Based on mobility prediction, APU enables nodes to update their position adaptively to the node mobility and traffic pattern. We embed APU into the well known Greedy Perimeter Stateless Routing Protocol (GPSR), and compare it with original GPSR in the ns-2 simulation platform. We conducted several experiments with randomly generated network topologies and mobility patterns. The results confirm that APU significantly reduces beacon overhead without having any noticeable impact on the data throughput of the network. This result is further validated through a trace driven simulation of a practical vehicular ad-hoc network topology that exhibits realistic movement patterns of public transport buses in a metropolitan city.

I. Introduction

With the growing popularity of positioning devices (e.g. GPS) and other localization schemes [1], geographic routing protocols are becoming an attractive choice for use in mobile ad hoc networks [2], [3], [4], [5]. The underlying principle used in these protocols involves selecting the next routing hop from amongst a node's neighbors, which is geographically closest to the destination. Since the forwarding decision is based entirely on local knowledge, it obviates the need to create and maintain routes for each destination. By virtue of these characteristics, position-based routing protocols are highly scalable and particularly robust to frequent changes in the network topology. Furthermore, since the forwarding decision is made *on the fly*, each node always selects the optimal next hop based on the most current topology. Several studies [2], [3], [6] have shown that these routing protocols offer significant performance improvements over topology-based routing protocols such as DSR [7] and AODV [8].

The forwarding strategy employed in the aforementioned geographic routing protocols requires the following information: (i) the position of the final destination of the packet and (ii) the position of a node's neighbors. The former can be obtained by querying a *location service* such as the Grid Location System (GLS) [9] or Quorum [10]. To obtain the latter, each node exchanges its own location information (obtained by using GPS or the localization schemes discussed in [1]) with its neighboring nodes. This allows each node to build a local map of the nodes within its vicinity, often referred to as the *local topology*.

However, in situations where nodes are mobile or when nodes often switch off and on, the local topology rarely remains static. Hence, it is necessary that each node periodically broadcasts its updated location information to all of its neighbors. These location update packets are usually referred to as *beacons*. In most geographic routing protocols (e.g. GPSR[2], GeoCast [11]), beacons are broadcast periodically for maintaining an accurate neighbor list at each node. Beaconing suffers from several drawbacks:

- The periodic transmission, reception and processing of beacon packets consumes energy which is a scarce resource in mobile devices.
- Beacon packets can collide with data packets. To recover from these MAC layer collisions, the nodes have to retransmit the data packets resulting in increased end-to-end delays and wastage of battery power.

Clearly, given the cost associated with transmitting beacons, it makes sense to adapt the frequency of beacon updates to the node mobility and the traffic conditions within the network, rather than employing a static periodic update policy. For example, if certain nodes are highly mobile, it makes sense to frequently broadcast their updated position. However, for nodes that do not change their positions frequently, periodic broadcasting of beacons is wasteful. Further, if only a small percentage of the nodes are involved in forwarding packets, it is unnecessary for nodes which are located far away from the forwarding path to employ periodic beaconing because these updates are not useful for forwarding the current traffic.

In this paper, we propose a novel beaconing strategy for geographic routing protocols called *Adaptive Position Updates strategy (APU)*. Our scheme eliminates the drawbacks of periodic beaconing by adapting to the system variations. APU incorporates two rules for triggering the beacon update process. The first rule uses a simple mobility prediction scheme to estimate when the location information broadcast in the previous beacon becomes inaccurate. The next beacon is broadcast only if the predicted error in the location estimate is greater than a certain threshold, thus tuning the update frequency to the mobility of the nodes. The second rule proposes an on-demand learning strategy, whereby beacons are exchanged in response to data packets from new neighbors in a node's vicinity. This ensures that nodes involved in forwarding data packets maintain a fresh view of the local topology. On the contrary, nodes that are not in the vicinity of the forwarding path are unaffected by this rule and do not broadcast beacons. By reducing the beacon updates, APU reduces the power and bandwidth utilization, resources which are scarce in MANETs. It also decreases the chance of link-layer collisions with the data packets and consequently reduces the end-to-end delay.

Note that, APU simply governs the beacon update strategy and is hence compatible with any geographic routing protocol. In this work, we have incorporated the APU strategy within GPSR (Greedy Perimeter Stateless Routing) [2] as a representative example. We have carried out simulations to evaluate the performance improvement achieved by APU with randomly generated network topologies and mobility patterns. We have also performed some initial experiments with realistic movement patterns of buses in a metropolitan city. Our initial results indicate that APU significantly reduces beacon overhead without having any noticeable impact on the data delivery rate. Thus, the APU strategy is a promising choice for use in Vehicular Ad-hoc Networks (VANETs) [12], which is an emerging and popular instantiation of MANETs.

The rest of paper is organized as follows. In Section II, we briefly discuss related work. A detailed description of the APU scheme is provided in Section III. Section IV presents a simulation-based evaluation highlighting the performance improvements achieved by the APU strategy in comparison with GPSR. Finally, Section V concludes the paper.

II. Related Work

DREAM [3] was one of the first protocols that incorporated position information within a routing protocol. In DREAM, each node maintains a position database that stores position information about all other nodes within the network. Of course, this approach is not scalable and requires a large number of beacon updates. The paper does mention that the position updates could be adapted to the node mobility. However, no details or practical strategies are discussed.

In [14], the location information is used to predict the expiration time of the link between two mobile nodes, known as the Route Expiration Time (RET). The routing protocol always selects routes with the largest RET for data forwarding. However, they only consider topology-based routing protocols in their work. In our work, we adopt a similar prediction scheme but use it for triggering the beacon updates. Further, our focus is on geographic routing protocols.

Greedy Perimeter Stateless Routing (GPSR) [2] uses one-hop neighbor's position and the destination location information to make the forwarding decision. It employs a greedy forwarding strategy wherein the packet is forwarded to the neighbor which is closest to the destination. Nodes broadcast beacon to immediate neighbors periodically for maintaining local topology.

Dongjin Son et al. [15] showed that the inaccuracy of location information has a significant impact on the performance of geographic routing protocols. They applied a similar mobility prediction scheme as [14] to GPSR and studied its impact of on the performance. However, they only use the prediction scheme to compute current position of neighbors and still employed periodic beacon updates.

Several other schemes have proposed strategies for reducing the routing overhead in location services, e.g. GLS, Quorum System, Homezone [16]. However, no one has yet addressed the issue of reducing the beacon updates. To the best of our knowledge, this is the first work to propose an adaptive beaconing strategy for geographic routing protocols.

III. Adaptive Position Update (APU)

We begin by listing the assumptions that our work is built upon: (1) all nodes are aware of their own position and velocity, (2) all links are bi-directional, (3) the beacon updates include the current location and velocity of the nodes, and (4) data packets can piggyback position and velocity updates and all one-hop neighbors operate in the promiscuous mode and hence can overhear the data packets.

Upon initialization, each node broadcasts a beacon informing its neighbors about its presence and its current location and velocity. Following this in most geographic routing protocols such as GPSR, each node periodically broadcasts its current location information. The position information received from neighboring beacons is stored at each node. Based on its own transmission range, current location and the position updates received from its neighbors, each node continuously updates its neighbor list. Neighbors which are outside the nodes transmission range are not considered as possible candidates for data forwarding. Thus, the beacons also play an important part in building the local topology.

Instead of periodic beaconing, APU adapts the beacon update intervals to the mobility of the nodes and the amount of data being forwarded in the neighborhood of the nodes. APU employs two mutually exclusive beacon triggering rules, which are discussed in the following subsections.

A. Mobility Prediction (MP) Rule

This rule adapts the beacon generation rate to the mobility of the nodes. Nodes that are highly mobile need to frequently update their neighbors since their locations are changing dynamically. On the contrary, nodes which move slowly do not need to send frequent updates. A periodic beacon update policy cannot satisfy both these requirements simultaneously, since a small update interval will be wasteful for slow nodes, whereas a larger update interval will lead to inaccurate position information for the highly mobile nodes.

In our scheme, upon receiving a beacon update from a node i, each of its neighbors, denoted by the set $N(i)$, records its current position and velocity and continues to track node i's location using a simple prediction scheme (discussed below). Based on this position estimate the neighbors $N(i)$, check whether node i is still within their transmission range and update their neighbor list accordingly. The goal of the MP rule is to send the next beacon update from i when the error between the predicted location in $N(i)$ and i's actual location is greater than an acceptable value. To achieve this, node i, must track its own predicted location in its neighbors, $N(i)$.

We use a simple location prediction scheme based on the physics of motion to track a nodes current location. Note that, in our discussion we assume that the nodes are located in a two-dimensional coordinate system with the location indicated by the x and y coordinates. However, this scheme can be easily extended to a three dimensional system. Table I illustrates the notations used in the rest of this discussion.

TABLE I

THE NOTATIONS FOR MOBILITY PREDICTION

Variables	Definition
(X_l^i, Y_l^i)	The coordinate of node i at time T_l (included in the previous beacon)
(V_x^i, V_y^i)	The velocity of node i along the direction of the x and y axes at time T_l (included in the previous beacon)
T_l	The time of the last beacon broadcast.
T_c	The current time
(X_p^i, Y_p^i)	The predicted position of node i at the current time

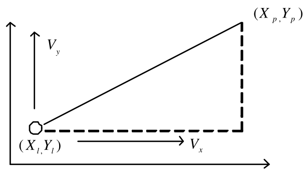


Fig. 1. An example of mobility prediction

As shown in Fig. 1, given the position of node i and its velocity along the x and y axes,at time T_l, its neighbors, $N(i)$ can estimate the current position of i, by using the following equations:

$$\begin{aligned} X_p^i &= X_l^i + (T_c - T_l) * V_x^i \\ Y_p^i &= Y_l^i + (T_c - T_l) * V_y^i \end{aligned} \tag{1}$$

Note that, here (X_l^i, Y_l^i) and (V_x^i, V_y^i) refers to the location and velocity information that was broadcast in the previous beacon from node i. Node i uses the same prediction scheme to keep track of its predicted location among its neighbors. Let (X_a, Y_a), denote the actual location of node i, obtained via GPS or other localization techniques. Node i then computes the deviation D_{devi}^i as follows:

$$D_{devi}^i = \sqrt{(X_a^i - X_p^i)^2 + (Y_a^i - Y_p^i)^2} \tag{2}$$

If the deviation is greater than a certain threshold, know as the *Acceptable Error Range (AER)*, it acts as a trigger for node i to broadcast its current location and velocity as a new beacon.

The AER threshold is an important parameter that can affect the performance of the APU scheme. A large value of AER will minimize the beacon updates but will result in a larger error in the estimated location of the node at its neighbors. On the contrary, a smaller value guarantees accuracy of location information amongst the neighbors but increases the beacon overheads. We have conducted several experiments and concluded that for most situations a value of 10 meters for the AER threshold achieves a good balance. These results have not been included due to space constraints.

The MP rule thus, tries to maximize the effective duration of each beacon, by broadcasting a beacon only when the position information in the previous beacon becomes inaccurate. This extends the effective duration of the beacon for nodes with low mobility, thus reducing the number of beacons. Further, highly mobile nodes can broadcast frequent beacons to ensure that their neighbors are aware of the rapidly changing topology.

B. On-Demand Learning (ODL) Rule

The MP rule solely may not be sufficient for maintaining an accurate local topology. Consider the example illustrated in Fig. 2, where node A moves from $P1$ to $P2$ at a constant velocity. Now, assume that node A has just sent a beacon while at $P1$. Since node B did not receive this packet, it is unaware of the existence of node A. Further, assume that the AER is sufficiently large such that when node A moves from $P1$ to $P2$ the MP rule is never triggered. However, as seen in Fig. 2 node A is within the communication range of B for a significant portion of its motion. Even then, neither A nor B will be aware of each other. Now, in situations where neither of these nodes are transmitting data packets, this is perfectly fine since they are not within communicating range once A reaches $P2$. However, if either A or B was transmitting data packets, then their local topology will not be updated and they will exclude each other while selecting the next hop node. In the worst-case, assuming no other nodes were in the vicinity, the data packets would not be transmitted at all.

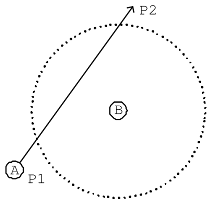


Fig. 2. An example illustrating a drawback of the MP rule

Hence, it is necessary to devise a mechanism which will maintain a more accurate local topology in those regions of the network where significant data forwarding activities are on-going. This is precisely what the *On-Demand Learning (ODL)* rule aims to achieve. As the name suggests, a node broadcasts beacons *on-demand*, i.e. in response to data forwarding activities that occur in the vicinity of that node. According to this rule, whenever a node overhears a data transmission from a *new* neighbor, it broadcasts a beacon as a response. In reality, a node waits for a small random time interval before responding with the beacon to prevent collisions with other beacons. Recall that, we have assumed that the location updates are piggybacked on the data packets and that all nodes operate in the promiscuous mode, which allows them to overhear all data packets transmitted in their vicinity. In addition, since the data packet contains the location of the final destination, any node that overhears a data packet also checks its current location and determines if the destination is within its transmission range. If so, the destination node is added to the list of neighboring nodes, if it is not already present. Note that, this particular check incurs zero cost, i.e. no beacons need to be transmitted.

We refer to the neighbor list developed at a node by virtue of the initialization phase and the MP rule as the *basic* list. This list is mainly updated in response to the mobility of the node and its neighbors. The ODL rule allows active nodes that are involved in data forwarding to enrich their local topology beyond this basic set. In other words, a *rich* neighbor list is maintained at the nodes located in the regions of high traffic load. Thus the rich list is maintained only at the active nodes and is built reactively in response to the network traffic. All inactive nodes simply maintain the basic neighbor list. By maintaining a rich neighbor list along the forwarding path, ODL ensures that in situations where the nodes involved in data forwarding are highly mobile, alternate routes can be easily established without incurring additional delays.

Fig. 3(a) illustrates the network topology before node A starts sending data to node P. The solid lines in the figure denote that both ends of the link are aware of each other. The initial possible routing path from A to P is A-B-P. Now, when source A sends a data packets to B, both C and D receive the data

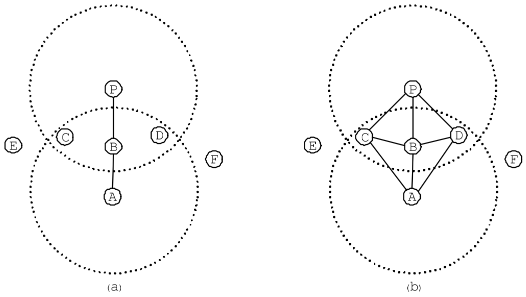


Fig. 3. An example illustrating the ODL rule

packet from A. As A is a new neighbor of C and D, according to the ODL rule, both C and D will send back beacons to A. As a result, the links AC and AD will be discovered. Further, based on the location of the destination and their current locations, C and D discover that the destination P is within their one-hop neighborhood. Similarly when B forwards the data packet to P, the links BC and BD are discovered. Fig. 3(b) reflects the enriched topology along the routing path from A to P.

Note that, though E and F receive the beacons from C and D, respectively, neither of them respond back with a beacon. Since E and F do not lie on the forwarding path, it is futile for them to send beacon updates in response to the broadcasts from C and D. In essence, ODL aims at improving the accuracy of topology along the routing path from the source to the destination, for each traffic flow within the network.

IV. Simulation Results

Our APU scheme is compatible with any geographic routing protocol. In this study, we have incorporated the APU strategy in the popular GPSR protocol, which we refer to as GPSR-APU. In this section, we present a simulation-based comparison of GPSR-APU with the original GPSR scheme. We initially use a random topology which allows us to study the effect of varying the node mobility on the performance of GPSR-APU. In addition, we have also studied the effect of the traffic load on APU using a realistic vehicular network.

For our evaluations we use the following metrics:

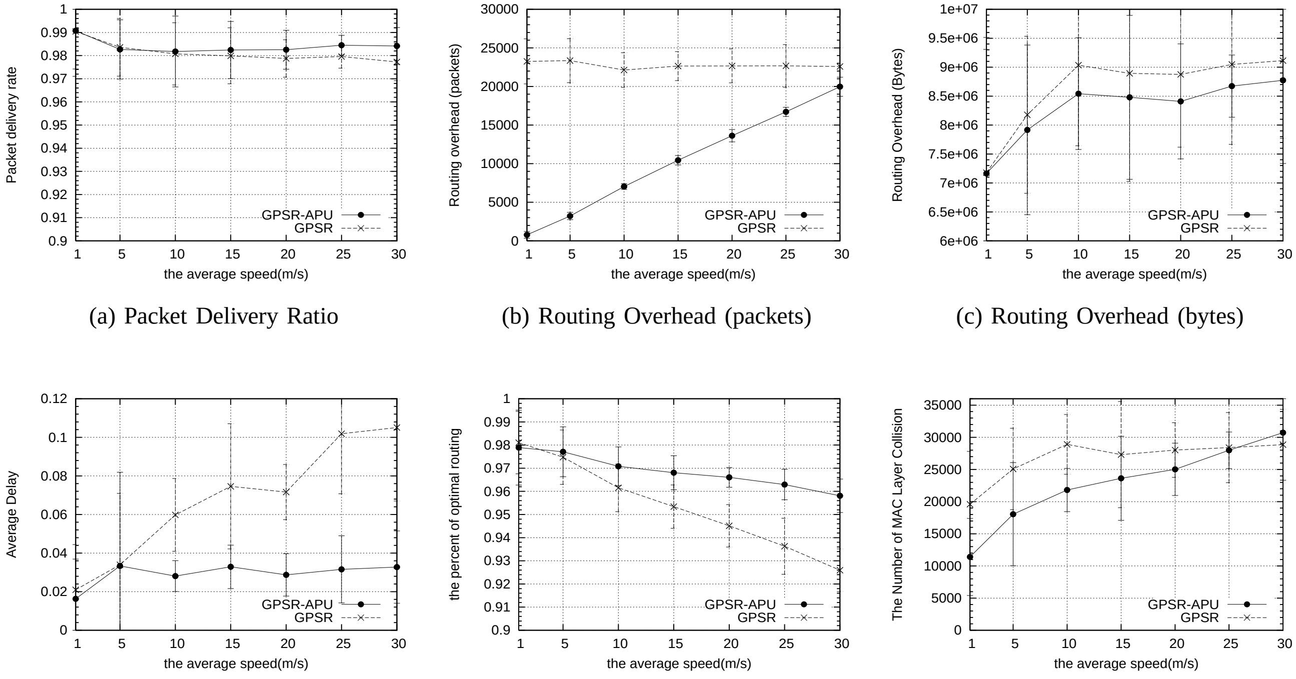
1) Packet Delivery Ratio: This measures the ratio of data packets delivered to the destinations to those generated by the sender. It reflects the accuracy of the protocol.
2) Routing Overhead (in packets): The beacon packets in geographic routing protocols constitute the routing overhead. This metric records the total number of beacons packets transmitted.
3) Routing Overhead (in bytes): Note that with APU, the location update is also piggybacked onto the data packets. Hence, it is unfair to just compare the total excess packets transmitted. This

metric records both the excess bytes transmitted in the data packets and the bytes transmitted in the beacon packets to reflect the overall overhead incurred due to beaconing.

4) MAC Layer Collisions: This measures the number of link layer collisions and reflects the interference caused due to the beacon packets.
5) End-to-End Delay: We also record the end-to-end delay incurred from the sender to the destination.
6) Optimal Route Percentage: This represents the percentage of data packets that were routed over the shortest-hop path to the destination. Since in geographic routing protocols, each node is unaware of the entire network topology, the forwarding path chosen may be longer than the optimal shortest-hop path.

A. Results Studying the Effects of Network Mobility

The simulations were conducted in NS-2 [13] with each experiment being run for 900 seconds. The results represented here are averaged over six runs, each using a different random seed. In each simulation run, 50 nodes were randomly placed in a region of size 1500m*600m. The radio range for each node was assumed to be 250 meters. The nodes move according to the random waypoint model [17]. In our experiments we varied the average node speeds from 1 m/s to 30 m/s. We used Constant Bit Rate (CBR) traffic sources with each source generating four packets per second. The size of data packets was 64 bytes, as used in [2], [17]. We selected 15 random source-destination pairs as the traffic flows.



(a) Packet Delivery Ratio

(b) Routing Overhead (packets)

(c) Routing Overhead (bytes)

(d) Average End-to-End Delay

(e) Percentage of Optimal Routes

(f) MAC Layer Collisions

Fig. 4. Simulation Results Studying the Effects of Network Mobility

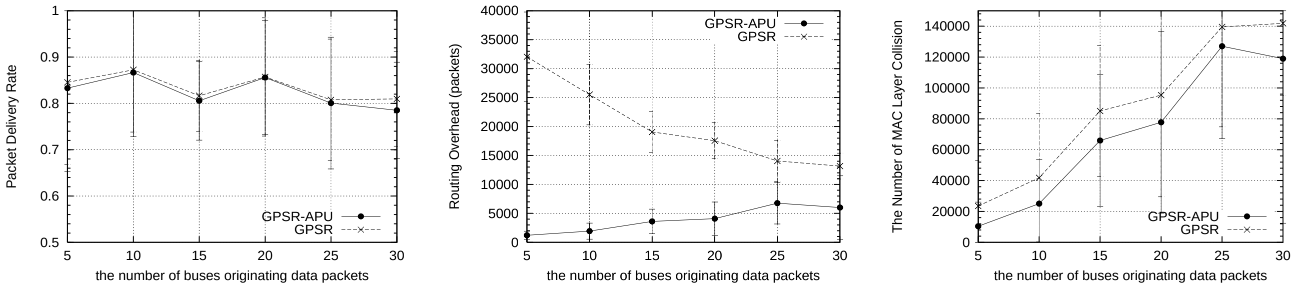
Fig. 4(a) illustrates that for low mobility, both GPSR and GPSR-APU achieve similar high values of the packet delivery ratio. However, at higher speeds GPSR-APU outperforms GPSR. This performance improvement is attributed to the fact that the APU scheme maintains an accurate topology along the data forwarding path. Hence, even though nodes move away quite frequently alternate routes are available for packet delivery due to the enriched topology maintained by APU.

Fig. 4(b) clearly shows APU can decrease the number of beacons exchanged without compromising on the packet delivery rate. At low mobility (1 m/s) the reduction in overhead with GPSR-APU is 95%. As expected, the overhead increases linearly with the speed, primarily due to the ODL rule, which generates more beacons to maintain an accurate topology along the forwarding path. On the contrary, since GPSR employs a periodic beaconing scheme, the overhead is independent of the mobility. Notice that even at very high speeds GPSR-APU reduces the overhead by 15% as compared to GPSR. Even comparing the overall overhead in terms of bytes (i.e. including the additional bytes sent in each data packet with APU), Fig. 4(c) indicates that GPSR-APU achieves a significant reduction. APU introduces an overhead of 4 bytes for each data packet. In our simulations this amounts to a 6.25% overhead (data packets are 64 bytes). For longer packets the corresponding overhead would be significantly smaller.

Fig. 4(d) shows that APU can reduce the average end-to-end delay for all speeds. At the low speed of 1m/s, the delay with APU falls 24%, whereas, in the case of high mobility (25m/s), the delay reduces by 70%. The reduced delay at low mobility is mainly due to the lower MAC layer collisions as is shown in Fig. 4(f). On the other hand, with high mobility, the shorter delay is largely because the topology with APU is more accurate than that with GPSR. This is justified by Fig. 4(e), which shows that at high speeds, APU can forward more data packets along the optimal shortest hop path as compared to GPSR. Thus, the average end-to-end delay of data packets are reduced. On the contrary, in GPSR, the average delay increases considerably with the node speed. This is due to the fact that the periodic beaconing employed by GPSR is not sufficient in maintaining an accurate topology map. As a result, a node may frequently send a packet to its neighbor, which is no longer within its transmission range. After several retransmissions the MAC layer would report that the next hop is unreachable, causing the node to pick a different neighbor. This increases the queuing delay at the intermediate nodes resulting in a significantly longer end-to-end delay. The accurate topology maintained by APU, however, minimizes the chances of similar prolonged queuing.

Finally, as expected, the number of MAC layer collisions for low to moderate speeds are much lower with GPSR-APU, mainly due to the reduction of beacon broadcasts, as depicted in Fig 4(f). For very high mobility, collisions are unavoidable, since beacons need to be sent frequently to maintain accurate local topology for achieving a high packet delivery ratio.

B. Results for a Realistic VANET Scenario



(a) Packet Delivery Ratio

(b) Routing Overhead (packets)

(c) MAC Layer Collisions

Fig. 5. Simulation Results for a Realistic Vehicular Networks with Varying Traffic Load

We now present some initial evaluations of the APU strategy in a real-word vehicular ad hoc network, a popular application domain for MANETs. Our aim here is to confirm whether some of the findings that we observed with random topologies do hold true in a realistic scenario. The mobility model used in the simulation is based on the actual movement of buses in the King County Metro bus system in Seattle, Washington. The same bus traces were also previously used by Jetcheva et al. [18]. The format of the bus traces consists of time, bus id, route id and bus location. We look at the bus movement at three different times in a rectangular region of 5 km x 8 km, each consisting of 50 buses. The three scenarios start at 10am, 11am and 12am respectively, with each run being active for 900 seconds. We assumed a radio range of 1km, which is consistent with that for the DSRC (Dedicated Short Range Communications) [19] standard proposed for vehicular communication. We used CBR traffic sources with the sender transmitting at 4 packets per second and a packet size of 64 bytes. The traffic load was varied from 5 to 30 flows. The results presented here are averaged over 9 runs, with each scenario being executed thrice with different random seeds.

Fig. 5(a) demonstrates that GPSR-APU achieves a similar packet delivery ratio as that of vanilla GPSR. This is despite the significantly lower beacon packets broadcast by APU as evidenced in Fig. 5(b). However, with an increase in the traffic load, we notice a slight increase in the beacons exchanged in GPSR-APU. This is primarily due to the ODL rule, which tries to maintain an accurate topology along the forwarding paths. On the contrary, with GPSR, since the beacons are piggybacked on the data packets, the number of explicit beacon packets that need to be broadcast decreases with increasing load. However, even at high traffic load, they are still significantly greater as compared to APU.

Finally, Fig. 5(c) shows that, the reduced number of beacon packets with GPSR-APU, results in a lower number of MAC layer collisions as compared with GPSR.

V. Conclusions

In this paper, we have identified the need to adapt the beacon update policy employed in geographic routing protocols to the node mobility and the traffic load. We proposed the *A*daptive Position Update (APU) strategy to address these problems. The MP rule uses mobility prediction to estimate the accuracy of the location estimate and adapts the beacon update interval accordingly, instead of using periodic beaconing. The ODL rule allows nodes along the data forwarding path to maintain an accurate view of the local topology by exchanging beacons in response to data packets overhead from new neighbors.

We have embedded APU within GPSR and have compared it with vanilla GPSR using extensive ns-2 simulations for varying node speeds. Our results indicate that the APU strategy significantly lowers the number of beacon updates while also achieving a better packet delivery rate. Further, with APU the packets are more likely to be routed along the shortest-hop path to the destinations, hence improving the end-to-end delay. We have also presented some initial results using realistic movement patterns of public transport buses within a city, which validate that the performance improvements of APU can be replicated in a real-world VANET scenario.

References

[1] J. Hightower and G. Borriello, ”Location Systems for Ubiquitous Computing”, IEEE Computer, vol. 34, no. 8, August 2001, pp. 57-66.

[2] B. Karp and H. T. Kung, ”GPSR: Greedy Perimeter Stateless Routing forWireless Networks,” in Proceedings of MOBICOM 2000, Boston, MA, USA, 2000, pp. 243-254.

[3] S. Basagni, et al., ”A Distance Routing Effect Algorithm for Mobility (DREAM),” in Proceedings of ACM MOBICOM 1998.

[4] L. Blazevic, S. Giordano, J-Y. LeBoudec, ”Self Organizing Wide-Area Routing”, in Proceedings of SCI 2000/ISAS 2000, Orlando, 2000.

[5] Y. Ko and N. H. Vaidya, ”Location-Aided Routing (LAR) in Mobile Ad Hoc Networks”, in ACM/Baltzer Wireless Networks, vol. 6, no. 4, pp. 307-321, September 2002.

[6] T. Camp et al., ”Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks”, in Proceedings of IEEE INFOCOM, 2002.

[7] D. B. Johnson, D. A. Maltz, Y. Hu, and J. G. Jetcheva, ”The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)”, draft-ietf-manet-dsr-07.txt, February 2002.

[8] C. E. Perkins et al., ”Ad hoc On-Demand DistanceVector (AODV) Routing”, draft-ietf-manet-aodv-11.txt, June 2002.

[9] J. Li, et al., ”A Scalable Location Service for Geographic Ad Hoc Routing”, in Proc. 6th Annual ACM/IEEE Int’l. Conf. Mobile Comp. Net, Boston, MA, 2000, pp. 120-30.

[10] Z. J. Haas and B. Liang, ”Ad Hoc Mobility Management with Uniform Quorum Systems”, IEEE/ACM Transactions on Networking, vol. 7, no. 2, pp. 228-40, Apr. 1999.

[11] J.C.Navas and T.Imielinski, ”Geographic Addressing and Routing”, in Proc. 3rd ACD/IEEE Intn’l. Conf. Mobile Comp. Net., Budaptest, Hungary, Sept. 26-30, 1997.

[12] FleetNet: Inter-vehicle Communication, http://www.et2.tu-harburg.de/fleetnet/

[13] The VINT Project. The UCB/LBNL/VINT Network Simulator-ns 2. http://www.isi.edu/nsnam/ns/

[14] W.Su, S.J. Lee, and M. Gerla, ”Mobility prediction and routing in ad hoc wireless networks”, International Journal of Network Management, 2001.

[15] D.Son, A.Helmy, B.Krishnamachari. ”The Effect of Mobility-Induced Location Errors on Geographic Routing in Mobile Ad Hoc and Sensor Networks: Analysis and Improvement Using Mobility Prediction”, in IEEE Transactions on Mobile Computing, vol. 3, pp. 233-245, July 2004.

[16] I. Stojmenovic, ”Home Agent Based Location Update and Destination Search Schemes in Ad Hoc Wireless Networks”, Tech. Rep. TR-99-10, Comp. Science, SITE, Univ. Ottawa, Sept. 1999.

[17] J. Broch, D.A. Maltz, et al., ”A Per-formance Comparison of Multi-Hop Wireless Ad Hoc Network RoutingProtocols,” in Proceedings of MOBICOM 98, Dallas, TX,Oct. 1998, pp. 85-97.

[18] J. G. Jetcheva, et al.. ”Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture”, in Proc. 5th IEEE Workshop on Mobile Computing Syst.& Applications (WMCSA 2003), Monterey, CA, October 2003.

[19] Dedicated Short Range Communications (DSRC), http://wireless.fcc.gov/services/its/dsrc/