# Adaptive Position Update for Geographic Routing in Mobile Ad Hoc Networks

Quanjun Chen, *Member*, *IEEE*, Salil S. Kanhere, *Senior Member*, *IEEE*, and Mahbub Hassan, *Senior Member*, *IEEE*

**Abstract**—In geographic routing, nodes need to maintain up-to-date positions of their immediate neighbors 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 neighbor positions. We contend and demonstrate that periodic beaconing regardless of the node mobility and traffic patterns in the network is not attractive from both update cost and routing performance points of view. We propose the *A*daptive Position Update (APU) strategy for geographic routing, which dynamically adjusts the frequency of position updates based on the mobility dynamics of the nodes and the forwarding patterns in the network. APU is based on two simple principles: 1) nodes whose movements are harder to predict update their positions more frequently (and vice versa), and (ii) nodes closer to forwarding paths update their positions more frequently (and vice versa). Our theoretical analysis, which is validated by NS2 simulations of a well-known geographic routing protocol, Greedy Perimeter Stateless Routing Protocol (GPSR), shows that APU can significantly reduce the update cost and improve the routing performance in terms of packet delivery ratio and average end-to-end delay in comparison with periodic beaconing and other recently proposed updating schemes. The benefits of APU are further confirmed by undertaking evaluations in realistic network scenarios, which account for localization error, realistic radio propagation, and sparse network.

**Index Terms**—Wireless communication, algorithm/protocol design and analysis, routing protocols

## 1 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]. The underlying principle used in these protocols involves selecting the next routing hop from among 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], [5] have shown that these routing protocols offer significant performance improvements over topology-based routing protocols such as DSR [6] and AODV [7].

The forwarding strategy employed in the aforementioned geographic routing protocols requires the following information: 1) the position of the final destination of the packet and 2) the position of a node's neighbors. The former can be obtained by querying a *location service* such as the Grid Location System (GLS) [8] or Quorum [9]. To obtain the latter, each node exchanges its own location information (obtained 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 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], [10], [11]), beacons are broadcast periodically for maintaining an accurate neighbor list at each node.

Position updates are costly in many ways. Each update consumes node energy, wireless bandwidth, and increases the risk of packet collision at the medium access control (MAC) layer. Packet collisions cause packet loss which in turn affects the routing performance due to decreased accuracy in determining the correct local topology (a lost beacon broadcast is not retransmitted). A lost data packet does get retransmitted, but at the expense of increased end-to-end delay. 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 frequently changing their mobility characteristics (speed and/or heading), it makes sense to frequently broadcast their updated position. However, for nodes that do not exhibit significant dynamism, periodic broadcasting of beacons is wasteful. Further, if only a

- *Q. Chen is with the Australian Centre for Field Robotics (ACFR), University of Sydney, Sydney, NSW 2006, Australia. E-mail: quanc@cse.unsw.edu.au.*
- *S.S. Kanhere and M. Hassan are with the School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia. E-mail: {salilk, mahbub}@cse.unsw.edu.au.*

*Manuscript received 1 Dec. 2008; revised 1 Oct. 2011; accepted 30 Dec. 2011; published online 12 Jan. 2012.*
*For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2008-12-0480. Digital Object Identifier no. 10.1109/TMC.2012.20.*

1536-1233/13/$31.00 © 2013 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

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)* [12]. 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, referred as *Mobility Prediction (MP)*, 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 dynamism inherent in the node's motion.

The second rule, referred as *On-Demand Learning (ODL)*, aims at improving the accuracy of the topology along the routing paths between the communicating nodes. ODL uses an on-demand learning strategy, whereby a node broadcasts beacons when it overhears the transmission of a data packet from a *new* neighbor in its vicinity. This ensures that nodes involved in forwarding data packets maintain a more up-to-date 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 very frequently.

We model APU to quantify the beacon overhead and the local topology accuracy. The local topology accuracy is measured by two metrics, *unknown neighbor ratio* and *false neighbor ratio*. The former measures the percentage of new neighbors a forwarding node is unaware of but that are actually within the radio range of the forwarding node. On the contrary, the latter represents the percentage of obsolete neighbors that are in the neighbor list of a node, but have already moved out of the node's radio range. Our analytical results are validated by extensive simulations.

In the first set of simulations, we evaluate the impact of varying the mobility dynamics and traffic load on the performance of APU and also compare it with periodic beaconing and two recently proposed updating schemes: distance-based and speed-based beaconing (SB) [13]. The simulation results show that APU can adapt to mobility and traffic load well. For each dynamic case, APU generates less or similar amount of beacon overhead as other beaconing schemes but achieve better performance in terms of packet delivery ratio, average end-to-end delay and energy consumption. In the second set of simulations, we evaluate the performance of APU under the consideration of several real-world effects such as a realistic radio propagation model and localization errors. The extensive simulation results confirm the superiority of our proposed scheme over other schemes. The main reason for all these improvements in APU is that beacons generated in APU are more concentrated along the routing paths, while the beacons in all other schemes are more scattered in the whole network. As a result, in APU, the nodes located in the hotspots, which are responsible for forwarding most of the data traffic in the network have an up-to-date view of their local topology, thus resulting in improved performance.

The rest of paper is organized as follows: In Section 2, we briefly discuss related work. A detailed description of the APU scheme is provided in Section 3, followed by a comprehensive theoretical analysis in Section 4. Section 5 presents a simulation-based evaluation highlighting the performance improvements achieved by APU in comparison with other schemes. Finally, Section 6 concludes the paper.

## 2 Related Work

In geographic routing, the forwarding decision at each node is based on the locations of the node's one-hop neighbors and location of the packet destination as well. A forwarding nodes therefore needs to maintain these two types of locations. Many works, e.g., GLS [8], Quorum System [9], have been proposed to discover and maintain the location of destination. However, the maintenance of one-hop neighbors' location has been often neglected. Some geographic routing schemes, e.g., [14], [15], simply assume that a forwarding node knows the location of its neighbors. While others, e.g., [2], [10], [11], use periodical beacon broadcasting to exchange neighbors' locations. In the periodic beaconing scheme, each node broadcasts a beacon with a fixed beacon interval. If a node does not hear any beacon from a neighbor for a certain time interval, called neighbor time-out interval, the node considers this neighbor has moved out of the radio range and removes the outdated neighbor from its neighbor list. The neighbor time-out interval often is multiple times of the beacon interval.

Heissenbuttel et al. [13] have shown that periodic beaconing can cause the inaccurate local topologies in highly mobile ad-hoc networks, which leads to performances degradation, e.g., frequent packet loss and longer delay. The authors discuss that the outdated entries in the neighbor list is the major source that decreases the performance. They proposed several simple optimizations that adapt beacon interval to node mobility or traffic load, including distance-based beaconing (DB), speed-based beaconing and reactive beaconing. We discuss these three schemes in the following.

In the distance-based beaconing, a node transmits a beacon when it has moved a given distance $d$. The node removes an outdated neighbor if the node does not hear any beacons from the neighbor while the node has moved more than $k$-times the distance $d$, or after a maximum time out of 5 s. This approach therefore is adaptive to the node mobility, e.g., a faster moving node sends beacons more frequently and vice versa. However, this approach has two problems. First, a slow node may have many outdated neighbors in its neighbor list since the neighbor time-out interval at the slow node is longer. Second, when a fast moved node passes by a slow node, the fast node may not detect the slow node due the infrequent beaconing of the slow node, which reduces the perceived network connectivity.

In the speed-based beaconing, the beacon interval is dependent on the node speed. A node determines its beacon interval from a predefined range $[a, b]$ with the exact value chosen being inversely proportional to its speed. The neighbor time-out interval of a node is a multiple $k$ of its beacon interval. Nodes piggyback their neighbor time-out interval in the beacons. A receiving node compares the piggybacked time-out interval with its own time-out interval, and selects the smaller one as the time-out interval for this neighbor. In this way, a slow node can have short time-out interval for its fast neighbor and therefore eliminate the first problem presented in the distance-based beaconing. However, the speed-based beaconing still suffer the problem that a fast node may not detect the slow nodes.

In reactive beaconing, the beacon generation is triggered by data packet transmissions. When a node has a packet to transmit, the node first broadcasts a beacon request packet. The neighbors overhearing the request packet respond with beacons. Thus, the node can build an accurate local topology before the data transmission. However, this process is initiated prior to each data transmission, which can lead to excessive beacon broadcasts, particularly when the traffic load in the network is high.

The APU strategy proposed in this work dynamically adjusts the beacon update intervals based on the mobility dynamics of the nodes and the forwarding patterns in the network. The beacons transmitted by the nodes contain their current position and speed. Nodes estimate their positions periodically by employing linear kinematic equations based on the parameters announced in the last announced beacon. If the predicted location is different from the actual location, a new beacon is broadcast to inform the neighbors about changes in the node's mobility characteristics. Note that, an accurate representation of the local topology is particularly desired at those nodes that are responsible for forwarding packets. Hence, APU seeks to increase the frequency of beacon updates at those nodes that overhear data packet transmissions. As a result, nodes involved in forwarding packets can build an enriched view of the local topology.

There also exist some geographic routing protocols that do not need to maintain the neighbor list and therefore can avoid position updates, e.g., IGF [16], GeRaf [17], BLR [18], ALBA-R [19]. These protocols are commonly referred to as beaconless routing protocols. The main ideal is that, the forwarding node broadcasts the data packet to all its neighbors who then distributedly decide which node relays the packet. Normally, in these protocols, after receiving a packet, each neighbor sets a timer for relaying the packet based on some metrics, e.g., the distance to the destination. The neighbor that has the smallest timer will expire first and relay the packet. By overhearing the relayed packet, other neighbors can cancel their own timers and ensure that no duplicate packet is transmitted. Hence, the beaconless routing protocols can avoid excessive position updates and are particular suitable for networks where the topology is highly dynamic, e.g., in wireless sensor network where nodes periodically switch on and off (to save energy consumption) [20].

## 3 Adaptive Position Update

We begin by listing the assumptions made in our work:

1. all nodes are aware of their own position and velocity,
2. all links are bidirectional,
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 the position updates received from its neighbors, each node continuously updates its local topology, which is represented as a neighbor list. Only those nodes from the neighbor list are considered as possible candidates for data forwarding. Thus, the beacons play an important part in maintaining an accurate representation of the local topology.

Instead of periodic beaconing, APU adapts the beacon update intervals to the mobility dynamics 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.

### 3.1 Mobility Prediction Rule

This rule adapts the beacon generation rate to the frequency with which the nodes change the characteristics that govern their motion (velocity and heading). The motion characteristics are included in the beacons broadcast to a node's neighbors. The neighbors can then track the node's motion using simple linear motion equations. Nodes that frequently change their motion 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 records node $i$'s current position and velocity and periodically track node $i$'s location using a simple prediction scheme based on linear kinematics (discussed below). Based on this position estimate, the neighbors can 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 node $i$ when the error between the predicted location in the neighbors of $i$ and node $i$'s actual location is greater than an acceptable threshold.

We use a simple location prediction scheme based on the physics of motion to estimate a node's current location. Note that, in our discussion, we assume that the nodes are located in a 2D coordinate system with the location indicated by the $x$ and $y$ coordinates. However, this scheme can be easily extended to a 3D coordinate system. Table 1 illustrates the notations used in the rest of this discussion.

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 can estimate the current position of $i$, by using the following equations:

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

TABLE 1
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 |

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 $D^i_{devi}$ as follows:

$$D^i_{devi} = \sqrt{(X_a^i - X_p^i)^2 + (Y_a^i - Y_p^i)^2}. \qquad (2)$$

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 MP rule, thus, tries to maximize the effective duration of each beacon, by broadcasting a beacon only when the predicted position information based on 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.

### 3.2 On-Demand Learning 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.

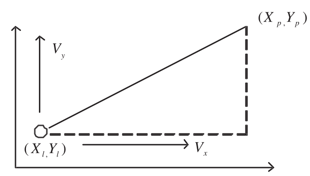



Fig. 1. An example of mobility prediction.

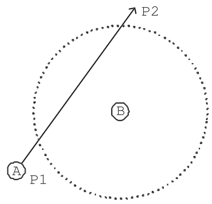



Fig. 2. An example illustrating a drawback of the MP rule.

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* 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. By a *new* neighbor, we imply a neighbor who is not contained in the neighbor list of this node. 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. 3a 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 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

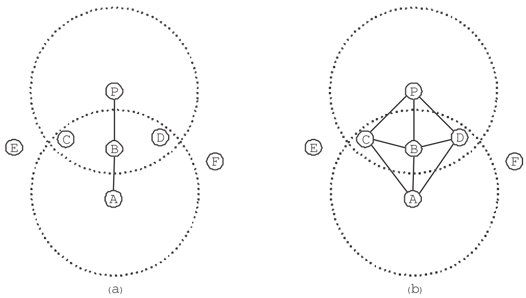



Fig. 3. An example illustrating the ODL rule.

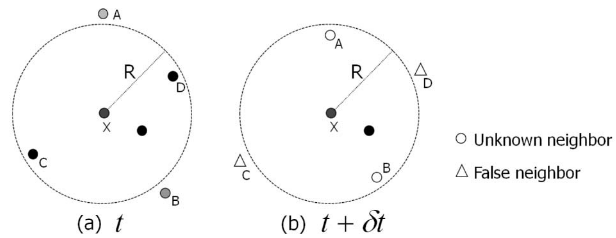



Fig. 4. Example illustrating unknown and false neighbors.

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. 3b 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.

## 4 Analysis of Adaptive Position Update

In this section, we analyze the performance of the proposed beaconing strategy, APU. We focus on two key performance measures: 1) update cost and 2) local topology accuracy. The former is measured as the total number of beacon broadcast packets transmitted in the network. The latter is collectively measured by the following two metrics:

- *Unknown neighbor ratio*. This is defined as the ratio of the new neighbors a node is not aware of, but that are within the radio range of the node to the total number of neighbors.
- *False neighbor ratio*. This is defined as the ratio of obsolete neighbors that are in the neighbor list of a node, but have already moved out of the node's radio range to the total number of neighbors.

The unknown neighbors of a node are the new neighbors that have moved in to the radio range of this node but have not yet been discovered and are hence absent from the node's neighbor table. Consider the example in Fig. 4, which illustrates the local topology of a node X at two consecutive time instants. Observe that nodes A and B are not within the radio range R of node X at time $t$. However, in the next time instant (i.e., after a certain period $\delta t$), both these nodes have moved into the radio range of X. If these nodes do not transmit any beacons, then node X will be unaware of their existence. Hence, nodes A and B are examples of unknown neighbors.

On the other hand, false neighbors of a node are the neighbors that exist in the node's neighbor table but have actually moved out from the node's radio range (i.e., these nodes are no longer reachable). Consider the same example in Fig. 4. Nodes C and D are legitimate neighbors of node X at time $t$. However, both these nodes have moved out of the radio range of node X in the next time instant. But, node X would still list both nodes in its neighbor table. Consequently, nodes C and D are examples of false neighbors.

Note that, the existence of both unknown and false neighbors adversely impacts the performance of the geographic routing protocol. Unknown neighbors are ignored by a node when it makes the forwarding decision. This may lead to suboptimal routing decisions, for example, when one of the unknown neighbors is located closer to the destination than the chosen next-hop node. If a false neighbor is chosen as the next hop node, the transmitting node will repeatedly retransmit the packet without success, before realizing that the chosen node is unreachable (in 802.11 MAC, the transmitter retransmits several times before signaling a failure). Eventually, an alternate node would be chosen, but the retransmission attempts waste bandwidth and increase the delay.

For mathematical tractability, we make the following simplifying assumptions:

- Nodes move according to the Random Direction Mobility (RDM) model, a popular model used in the analysis and simulations of wireless ad hoc networks. This mobility model maintains a uniform distribution of nodes in the target region over the entire time interval under consideration [21].
- Each node has the same radio range $R$, and the radio coverage of each node is a circular area of radius $R$.
- The network is sufficiently dense such that the greedy routing always succeeds in finding a next hop node. In other words, we assume that a forwarding node can always find a one-hop neighbor that is closer to the destination than itself.
- The data packet arrival rate at the source nodes and the intermediate forwarding nodes is constant.

### 4.1 Analysis of the Beacon Overhead

Recall that the two rules employed in APU are mutually exclusive. Thus, the beacons generated due to each rule can be summed up to obtain the total beacon overhead. Let the beacons triggered by the MP rule and the ODL rule over the network operating period be represented by $O_{MP}$ and $O_{ODL}$, respectively. The total beacon overhead of APU, $O_{APU}$, is given by

$$O_{APU} = O_{MP} + O_{ODL}. \tag{3}$$

Next, we proceed to separately analyze $O_{MP}$ and $O_{ODL}$.

#### 4.1.1 Beacon Overhead Due to the MP Rule ($O_{MP}$)

Recall that we have assumed that the nodes follow the RDM mobility model. According to this model, a node's trajectory consists of multiple consecutive linear segments. In each segment, the node randomly selects a direction (or heading), a speed, and a travel duration from certain predefined ranges. The node moves at the selected speed in the chosen direction until the selected travel duration expires. At the end of the segment, the node pauses for a random time interval and then randomly selects another set of values for the next segment and changes its motion accordingly. For mathematical tractability, we neglect the pause time between successive segments (i.e., we assume that nodes instantly transition to the next segment).

Recall that, according to the MP rule, a node periodically predicts its own location using the motion parameters advertised in the last transmitted beacon, and compares the predicted location with its actual location. If this difference is greater than the threshold $AER$, a new beacon is broadcast (see Section 3.1). Consequently, the threshold $AER$ directly influences the frequency and hence the number of beacon broadcasts. We seek to derive the upper bound of the beacon overhead and hence assume that the AER is zero (the lowest possible value). In this case, a beacon will be broadcast immediately in response to any change in the node's motion characteristics (direction and speed). Since, in the RDM model, a node changes these characteristics at the end of every linear segment, the number of beacons transmitted by the node are equal to the total number of linear segments traversed by the node. Since, the travel duration of each segment is randomly selected from $(0, \tau)$, on average, a node completes traversing a linear segment after an interval of $t/2$. In other words, the average duration between two successive beacon broadcasts is $t/2$. The number of beacons broadcast by a node during a finite time period of $\Gamma$ is $2\Gamma/\tau$. Therefore, for a total of $N$ nodes in the network, the total beacon overhead triggered by the MP rule, $O_{MP}$ is given by

$$O_{MP} = \frac{2N\Gamma}{\tau}. \tag{4}$$

#### 4.1.2 Beacon Overhead Due to the ODL Rule ($O_{ODL}$)

According to the ODL rule, whenever a node overhears a data transmission from a *new* neighbor, it broadcasts a beacon as a response (see Section 3.2). In other words, beacons are transmitted in response to data forwarding activities. Let $\chi$ denote the total number of data packet forwarding operations that occur over the network operating period and let $\gamma$ be the average number of beacons that are triggered by each forwarding operation. Now, the total beacons triggered by the ODL rule, $O_{ODL}$, can be represented by

$$O_{ODL} = \chi \cdot \gamma. \tag{5}$$

Next, we proceed to derive $\chi$ and $\gamma$.

***1. Analysis of*** $\chi$**.** The total number of data packet forwarding operations can be represented as the product of the number of packets generated in the network and the number of times each packet is forwarded. The number of packets generated in the network during a finite time period of $\Gamma$ can be expressed as $\lambda M\Gamma$, where $\lambda$ is the packet generation rate (packets per second) at each source, $M$ is the number of communication pairs (i.e., source-destination pairs). Let $\overline{H}$ be the average number of hops along the forwarding paths between the source and destination nodes. In other words, each packet is forwarded on average, $\overline{H}$ times, as it progresses from the source to the destination. Hence, $\chi$ can be represented as

$$\chi = \lambda M\Gamma \cdot \overline{H}. \tag{6}$$

Since $\lambda$, $M$, and $\Gamma$ are known network parameters, we only need to derive $\overline{H}$.

In [17], the authors have analyzed the forwarding behavior of greedy geographic routing and derived the average number of hops along a forwarding path, given the euclidean distance separating the source and destination node in a static multihop wireless network. However, in this paper, we consider a mobile ad hoc network, wherein, due to the mobility of the nodes, the distance between the source and destination nodes of a communicating pair is bound to change with time. This distance can be represented as a random variable. In the following, we first estimate the mean value of the source-destination distance. Then, we use the results in [17] to estimate the average hop count, $\overline{H}$.

Since, the nodes are uniformly distributed in the network (a property of the RDM model [21]), the distance between a source-destination pair is equivalent to the distance between two randomly selected points. In [22], Bettstetter et al. have analyzed the distance between two randomly select points, and formulated the average distance ($\overline{D}$) as

$$\begin{aligned}\overline{D} = &\frac{1}{15}\left[\frac{A^3}{B^2} + \frac{b^3}{B^2} + \sqrt{A^2+B^2}\left(3 - \frac{A^2}{B^2} - \frac{B^2}{A^2}\right)\right] \\ &+ \frac{1}{6}\left[\frac{B^2}{A}\operatorname{arccosh}\left(\frac{\sqrt{A^2+B^2}}{B}\right) + \frac{A^2}{B}\operatorname{arccosh}\left(\frac{\sqrt{A^2+B^2}}{A}\right)\right],\end{aligned} \tag{7}$$

where $A \times B$ denotes the network dimensions. Based on work [17], given the euclidean distance $\overline{D}$ between the source and destination node, the average number of hops between these nodes can be represented as follows:

$$\overline{H} = \frac{\overline{D}}{R \cdot \left[1 - \int_0^1 1 - \exp(\rho R(\arccos{(t)} - t\sqrt{1-t^2}))dt\right]}, \tag{8}$$

where $\rho$ is the average node density, which is given by $A \cdot B/N$.

Combining (6), (7), and (8), we obtain the total number of data packet forwarding operations $\chi$.

***2. Analysis of*** $\gamma$**.** According to the ODL rule, when a node forwards a data packet, the new neighbors that have moved in to the radio range of this forwarding node (and are hence unaware of the existence of the node forwarding the packet), broadcast beacons upon overhearing the packet transmission. This allows the forwarding node to maintain an up-to-date view of the local topology. Thus, the average number of beacons triggered by each packet forwarding operation, i.e., $\gamma$ is equal to the number of new neighbors that have entered the radio range of the forwarding node in the time interval between two successive data forwarding operations.

Recall that one of the assumptions in our analysis is that the packet arrival rate at the source nodes and the intermediate forwarding nodes is constant, and is represented by $\lambda$. Thus, the time interval between two consecutive data forwarding operations at a node is $1/\lambda$. Since the nodes are uniformly distributed in the network, on average each node has the same number of one-hop neighbors, which is given by $\rho\pi R^2$ (where $\rho$ is the nodes density). In steady state, the average number of new neighbors that enter the radio range of a node during the interval $1/\lambda$ is equal to the average number of neighbors that leave this region (this has been validated by simulations but have been omitted for brevity). Therefore, $\gamma$ is equal to the average number of neighbors that move out of the radio range of the forwarding node during the interval $1/\lambda$.

Let $\delta(t)$ be the probability that a neighboring node moves out the radio range of a node during a small interval $t$. In other words, $\delta(t)$ denotes the link breakage probability. Given that a node has an average of $\rho\pi R^2$ neighbors, the number of neighbors that move out of the radio range of a node during the time $1/\lambda$ follows:[1]

$$\gamma = \rho\pi R^2 \cdot \delta\left(\frac{1}{\lambda}\right). \tag{9}$$

Next, we derive $\delta(t)$. Intuitively, $\delta(t)$ is a function of the mobility pattern of the nodes. The faster the nodes move, the higher is the link breakage probability. We prove the following theorem:

**Theorem 1.** *The probability that the link between two neighboring nodes ceases to exists after a small time interval t is given by*

$$\delta(t) = \frac{1}{\pi a R^2}\int_0^R l \cdot \left[\int_0^{2\pi}\int_0^a g(r,\theta,l)drd\theta\right]dl, \tag{10}$$

*where* $a = v_{max} \cdot t$, *and* $g(r,\theta,l)$ *is defined as*

$$g(r,\theta,l) = \begin{cases} 1-\alpha a + u\sin\alpha - \int_{\pi-\alpha}^{\pi}\sqrt{R^2-u^2\sin^2 v}dv & \mathrm{u} \geq \sqrt{(\mathrm{R}+\mathrm{a})^2}, \\ 1-\alpha a + u\sin\alpha - \int_{\pi-\alpha}^{\pi}\sqrt{R^2-u^2\sin^2 v}dv - 2\int_{\pi-a\sin\frac{R}{u}}^{\pi-\alpha}\sqrt{R^2-u^2\sin^2 v}dv & \sqrt{(\mathrm{R}+\mathrm{a})^2} > \mathrm{u} \geq \mathrm{R}, \\ 1-\alpha a + u\sin\alpha - \int_0^{\pi-\alpha}\sqrt{R^2-u^2\sin^2 v}dv & \mathrm{R} > \mathrm{u} \geq \mathrm{R}-\mathrm{a}, \\ 0 & \mathrm{R}-\mathrm{a} > \mathrm{u}, \end{cases} \tag{11}$$

*where* $u = \sqrt{(l - r\cos\theta)^2 - r^2\sin^2\theta}, \alpha = \arccos\frac{u+a^2-R^2}{2ua}$.

The proof is omitted here due to the page constraint. Interested readers can refer to the technical report [23]. Given the link breakage probability $\delta(t)$, we can use (9) to estimate $\gamma$, i.e., the average number of beacons that are triggered by each data packet forwarding operation. Since, we have derived the total number of data packet forwarded $\xi$ earlier, we can calculate the beacon overhead triggered by ODL rule using (5).

Finally, according to (3), the total beacon overhead generated by APU ($O_{APU}$) follows:

$$O_{APU} = O_{MP} + O_{ODL} = \frac{2N \cdot \Gamma}{\tau} + \chi \cdot \gamma. \tag{12}$$

### 4.2 Analysis of the Local Topology Accuracy

Recall that we have defined two metrics that collectively represent the neighbor table accuracy: 1) unknown neighbor ratio and 2) false neighbor ratio. The neighbor table maintained by a node is only referenced when the node has to forward a packet. Consequently, it only makes sense to calculate the neighbor table accuracy at the time instants when the node is forwarding a data packet.

We first analyze the unknown neighbor ratio. In our earlier analysis (see analysis of $\gamma$), we have shown that, according to the ODL rule, the average number of new neighbors that enter the radio range of a node between two successive forwarding operations (i.e., the interval $1/\lambda$) is given by $\gamma$. The node will only become aware of these new neighbors when it forwards the next packet, since these neighbors will broadcast beacons announcing their presence in response to the packet transmission. According to (9), on average $\rho\pi R^2 \cdot \delta(1/\lambda)$ new neighbors enter the radio range of a forwarding node during the interval $1/\lambda$. The number of actual neighbors is the total number of nodes within the radio range of the forwarding node, which is $\rho\pi R^2$ on average. Therefore, the unknown neighbor ratio, represented by $\Lambda^m_{APU}$, can be computed as follows:

$$\Lambda^m_{APU} = \frac{\rho\pi R^2 \cdot \delta\left(\frac{1}{\lambda}\right)}{\rho\pi R^2} = \delta\left(\frac{1}{\lambda}\right). \tag{13}$$

We now proceed to evaluate the false neighbor ratio. As per the MP rule, a node periodically estimates the current locations of its neighbors using (1). Let $\omega$ denote the periodicity of this operation. At the beginning of each period, the node updates its neighbor list by removing all the false neighbors (i.e., those nodes that are estimated to have moved out of its radio range). Since, data packets arrive at the forwarding node at random during the interval $\omega$, the average time of arrival of a packet is given by $\omega/2$. The number of false neighbors at time $\omega/2$ is the number of neighbors that have moved out of the radio range during $\omega/2$. Therefore, according to (9), the false neighbor ratio, denoted by $\Lambda^f_{APU}$ is given by

$$\Lambda^f_{APU} = \frac{\rho\pi R^2 \cdot \delta\left(\frac{\omega}{2}\right)}{\rho\pi R^2} = \delta\left(\frac{\omega}{2}\right). \tag{14}$$

## 5 Simulation Results

In this section, we present a comprehensive simulation-based evaluation of APU using the popular NS-2 simulator. We compare the performance of APU with other beaconing schemes. These include PB and two other recently proposed adaptive beaconing schemes in [13]: (i) Distance-based Beaconing and (ii) Speed-based Beaconing (see Section 2).

1. Note that, (9) only holds as an approximation. The correct way to calculate $\gamma$ is $\int_0^{+\infty}\rho\pi R^2\delta(t)\lambda e^{-\lambda t}dt$. However, numerical comparisons have shown that the approximation is quite accurate. These results are omitted for reasons of brevity.

TABLE 2
Energy Consumption in Each Operation

| Operation | $\mu$W· sec/byte[8] | $\mu$W· sec |
|---|---|---|
| point-to-point send | 0.48 ×size | +431 |
| broadcast send | 2.1 ×size | +272 |
| point-to-point recv | 0.12 ×size | +316 |
| broadcast recv | 0.26 ×size | +50 |
| promiscuous recv | 0.12 ×size | +83 |
| promiscuous discard | 0.11 ×size | +54 |

*(The point-to-point communication uses date rate of 11 Mbps. The broadcasting uses data rate of 2 Mbps. Therefore, broadcasting costs more energy than point-to-point sending)*

We conduct three sets of experiments. In the first set of simulations, we demonstrate that APU can effectively adapt the beacon transmissions to the node mobility dynamics and traffic load. In addition, we also evaluate the validity of the analytical results derived in Section 4, by comparing the same with the results from the simulations. In the second set of experiments, we consider the impact of real-world factors such as localization errors, realistic radio propagation, and sparse density of the network on the performance of APU. In the third set of experiments, we evaluate the impact of parameter AER (which is from MP component) on the overall performance of APU. This enables us to investigate which component (MP or ODL) contributes to the performance more significantly.

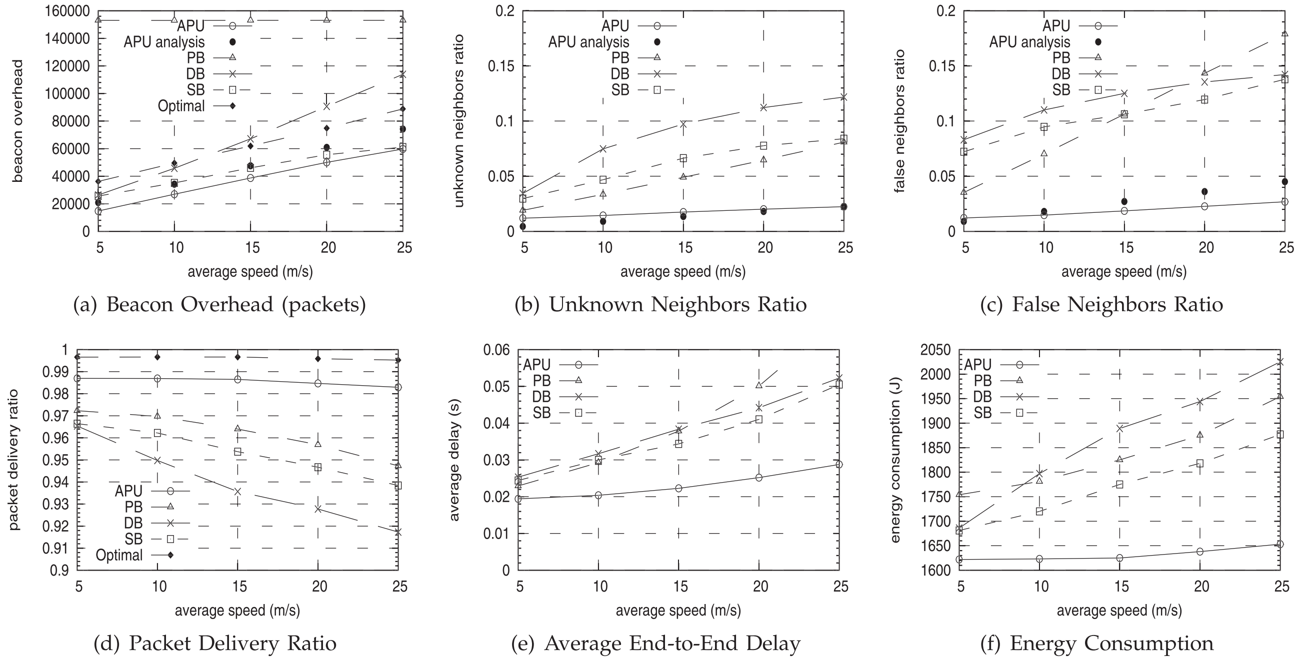
We use two sets of metrics for the evaluations. The first set includes the metrics used in our analysis, viz., beacon overhead and local topology accuracy (false and unknown neighbor ratio), which directly reflect the performance achieved by the beaconing scheme. Note that the beaconing strategies are an integral part of geographic routing protocols. The second set of metrics seek to evaluate the impact of the beaconing strategy on the routing performance. These include: 1) packet delivery ratio, which is measured as the ratio of the packets delivered to the destinations to those generated by all senders, 2) average end-to-end delay incurred by the data packets, and 3) energy consumption, which measures the total energy consumed in the network. We adopt the widely used energy consumption model, which estimates the energy consumption for each basic operation (e.g., transmitting, receiving, and overhearing in promiscuous mode) based on empirical data collected from commercial wireless cards [24]. The energy consumption for each radio operation is listed in Table 2. We also measured the average hop count traversed by the packets. However, we found that this metric is not an effective tool for comparing beaconing schemes (please refer to our technical report [23] for the details). In the simulations, we have implemented GPSR [2] as an illustrative example of a geographic routing protocol. We simulate IEEE 802.11b as the MAC protocol with wireless bandwidth of 11 Mbps and assume a two-ray ground propagation model unless otherwise stated.

### 5.1 Impact of Node Mobility on Beaconing Schemes

We first evaluate the impact of varying the mobility dynamics of the nodes on the performance on APU. In addition, we compare the performance of APU with other beaconing schemes. The simulations are conducted in NS-2 with each experiment being run for 1,000 seconds. The results represented here are averaged over 30 runs (the standard deviation achieved is on average less than 5 percent of the mean value). In each simulation, 150 nodes are randomly placed in a region of size $1{,}500\ \mathrm{m} * 1{,}500\ \mathrm{m}$. The radio range for each node is assumed to be 250 meters (thus the average number of one-hop neighbors for each node is 12). We use Constant Bit Rate (CBR) traffic sources with each source generating four packets per second. We simulate 15 traffic flows and randomly select nodes as source-destination pairs as the traffic flows. We have assumed that the nodes move according to the RDM model, to be consistent with our analytical results. First, we study the impact of changing the mobility dynamics of the nodes on the performance of APU and PB. Note that, the faster the node moves, the more frequently it changes its mobility parameters (i.e., speed and direction). We vary the average speed of the node from 5 m/s (18 km/hr, representing low dynamism) to 25 m/s (90 km/hr, representing high dynamism). This range is consistent with typical vehicular mobility scenarios. The travel duration for each segment in RDM (see Section 4.1.1) is randomly selected from (0, 40 s).

We assume that the prediction period in APU ($\omega$) is 1 s. The parameter of AER is 40 m. We have studied the impact of AER values on the performance of APU. However, we omit the results here due to the page limitation. Please see our technical report [23] for the details. The beacon period ($\epsilon$) in PB is also assumed to be 1 s, which is the default value in NS2 and also is recommended in [13]. The neighbor timeout interval in PB is set to 3 s. In DB [13], assuming that the distance parameter is $d$, and a node is moving at speed $v$, the beacon interval is given by, $d/v$. We have set the distance parameter, $d = 20\ \mathrm{m}$ and the neighbor time-out interval as twice the beacon interval, as suggested in [13]. In SB, if the speed of the node is $v$, then its beacon interval is given by $B = a + (b - a) \cdot (\frac{v_{max} - v}{v_{max} - v_{min}})^n$, where $[a, b]$ is predefined beacon interval range; $v_{min}$ and $v_{max}$ are the minimal and maximal node speeds. We assume that the beacon interval range is [1 s, 5 s] and $n = 4$, as suggested in [13]. Since, the average speed is varied from 5 to 25 m/s in the simulations, $v_{min} = 0$ and $v_{max} = 50$. Note that, in the simulations, PB scheme does not use promiscuous mode while all other schemes piggyback beacon information in data packets and employ promiscuous mode.

We also include the optimal performance that be achieved in terms of delivery ratio as a performance benchmark. The best possible delivery ratio can be achieved if each node can select the optimal next hop node according to geographic routing. This would require each node to be always aware of the exact location of its current neighbors. We simulated such a hypothetical scheme and refer to it as *optimal*. Note that, in simulating the above, we did not actually generate any beacons, since the simulator has a global view of the entire network topology. However, we can readily estimate the minimum beacon overhead incurred by the optimal scheme. The minimum possible beacon overhead can be achieved if a forwarding node (i.e., a node that currently holds a packet that it needs to forward) is immediately informed about a change in the position of its next hop node. At least one neighbor of the




(a) Beacon Overhead (packets) (b) Unknown Neighbors Ratio (c) False Neighbors Ratio

(d) Packet Delivery Ratio (e) Average End-to-End Delay (f) Energy Consumption

Fig. 5. Impact of node speed on the performance of beaconing schemes.

forwarding node should broadcast a beacon to reflect the change. Therefore, the minimum beacon overhead incurred by the optimal scheme is equal to the number of times that forwarding nodes change their next hops, which can be readily computed in simulations by observing the dynamics of the network topology.

We initially focus on the first set of metrics, i.e., the beacon overhead and the unknown and false neighbor ratios. Fig. 5a shows that the beacon overhead of APU increases linearly as a function of the average speed. This behavior is primarily attributed to the ODL rule. Recall that in the OLD rule, when a node forwards a data packet, all of its new neighbors that overhear the data packet respond with beacons.[2] When the network topology is highly dynamic, the local topology of a node frequently changes with several new neighbors entering the radio range. As a result, APU generates more beacons in order to keep up with the frequent changes of topologies. With DB, we observe a similar linear increase. This is expected, because, the beacon periodicity in DB is inversely proportional to the node speed. Finally, with SB, the beacon overhead also increases with increase in average speed, though not linearly. The beacon overhead tends to saturate as the average speed increases. This is because of the polynomial relationship that exists between the beacon update period and the node speed. In contrast, observe that PB results in very high beacon overhead, which does not vary significantly with the node speed. This is because in PB, the beacon broadcasts are independent of the node mobility.

Fig. 5b shows that APU can achieve a similar unknown neighbor ratio as that of PB, despite the fact that APU generates significantly less beacon overhead. Recall that, the beacon broadcasts in APU are more concentrating around the routing paths due to the ODL rule. Therefore, these beacons are highly effective in maintaining an up-to-date view of the local topology at the nodes involved in forwarding most of the traffic. On the contrary, both DB and SB exhibit higher unknown neighbor ratio as compared to APU. In particular, when the average node speed is 25 m/s, the unknown neighbor ratio for DB and SB is more than twice as that of APU. We attribute this increase in the unknown neighbors to the fact that in both DB and SB, when a fast moving node passes a slow node, the fast node may not detect the slow node due to the infrequent beacon transmissions by the slow node. Note that, in APU, due to the ODL rule, if either of these nodes are involved in forwarding packets, beacons would be exchanged, thus reducing the likelihood of unknown neighbors.

Fig. 5c illustrates that APU can achieve a very low false neighbor ratio as compared with the other three schemes. This can be explained as follows: Since each node in APU uses mobility prediction to track the locations of its neighbors (MP rule), the node can always quickly remove the obsolete neighbors, which have moved out of its radio range, from the neighbor list. On the contrary, a node in PB, DB, or SB only passively removes an obsolete neighbor when the node has not heard any beacons from the neighbor during a certain time window. Therefore, the removal of obsolete neighbors is delayed resulting in a higher false neighbor ratio. In summary, APU succeeds in maintaining an accurate view of the local topology in the network, while keeping the beacon overheads to a minimum.

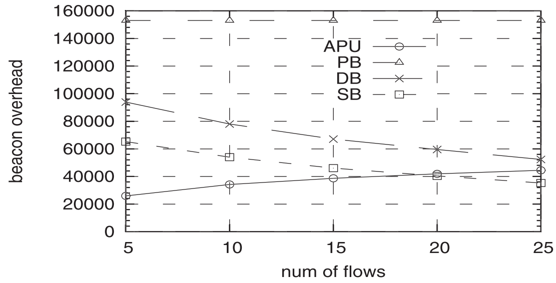
We also seek to validate the results from our analysis in Section 4. We obtain the analytical results for the beacon overhead, false neighbor ratio and unknown neighbor ratio for APU by substituting the simulation parameters in the corresponding equations. These results are compared with the corresponding simulation results in Figs. 5a, 5b, and 5c.

2. In the simulations, each neighbor delays the beacon update for a random time between 0 and 2 ms to avoid synchronizations and packet collisions. We have observed that varying the range of the random delay does not have noticeable impact on the performances.

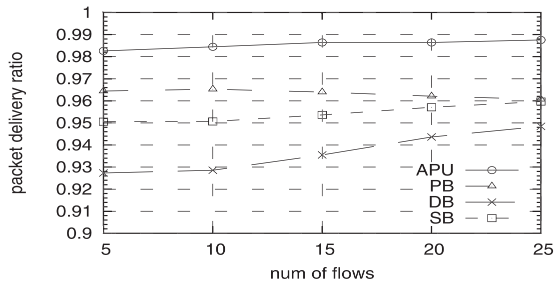
One can readily observe that the analytical model can provide an upper bound for the beacon overhead and false neighbor ratio, and provide an accurate approximation for the unknown neighbor ratio. There are several reasons for the inconsistency between the analysis results and simulation results. First, as explained for (4), our analysis seeks to derive an upper bound for the beacon overhead generated by MP rule. Second, in the analysis, we have assumed that the packet arrival rate at all intermediate nodes is constant ($\lambda$). However, this assumption may not hold if multiple flows share some common forwarding nodes. For example, if an intermediate node forwards data packet from multiple flows, the packet interarrival duration at such nodes would be less than $1/\lambda$. Consequently, in this shorter interval, fewer new neighbors would enter the radio range of these nodes. As a result, the number of beacons transmitted according to the OLD rule would be lower as compared to when the routing paths for multiple flows are completely disjoint (as assumed in the analysis). Hence, our analytical results overestimate the beacon overheads for APU. Third, when we estimate the link breakage probability for two neighboring nodes in Theorem 1, we implicitly assume that, for any two pairs of neighboring nodes, their link breakage probabilities are independent. However, this is not true in practice. For example, assume that node $A$ has two neighbors: $B$ and $C$. The link breakage probability of nodes $A$ and $B$ cannot be independent of the node pair $A$ and $C$, since they share a common node, $A$. This dependency is increased in higher mobility scenarios, which leads to the inconsistency between the analysis results of false neighbor ratio and the corresponding simulation results, as shown in Fig. 5c.

Next, we focus on the second set of metrics, which evaluate the impact of the beaconing strategies on the performance of the geographic routing protocols (GPRS in this case). These metrics include the packet delivery ratio, end-to-end packet delay, and energy consumption. Since, APU is successful in maintaining an up-to-date view of the local network topology, it also achieves a consistently high packet delivery ratio as illustrated in Fig. 5d, independent of the speed, since each node involved in forwarding a packet is almost always able to find an appropriate next hop neighbor. Fig. 5d also shows that APU can achieve comparable packet delivery ratio as the optimal scheme. However, the beacon overhead generated by APU is considerably lower than that of the optimal scheme, as shown in Fig. 5a. Since in APU most packets are forwarded along the optimal paths than other schemes, APU achieves lowest end-to-end delay, as can be seen from Fig. 5e. In comparison, all the other three schemes (i.e., PB, DB, and SB) exhibit a decrease in their packet delivery ratio as the average speed of the nodes increases (Fig. 5d). Further, as seen from Fig. 5e, the average end-to-end delay also increases as a function of speed for these three schemes. This can be attributed to the fact that the false and unknown neighbor ratios are considerably higher in all these schemes as compared to APU.

Fig. 5f compares the total energy consumption for the different schemes. The energy consumption depends on the beacon overhead and the total number of data packets transmitting. Fig. 5f shows that, despite the use of promiscuous mode, APU can achieve the lowest energy consumption. The reason is twofolds. First, comparing promiscuous mode to nonpromiscuous mode, the extra energy consumption used for data packet overhearing is not significant, as shown in [24, Table III]. Second, APU generates less beacon overhead and, since packets are more likely to follow optimal routing paths than other schemes (evidenced by Fig. 5d), the total number of data packets transmitted is also smaller than other schemes. As a result, APU achieves the lowest energy consumption.




(a) Beacon Overhead (packets)

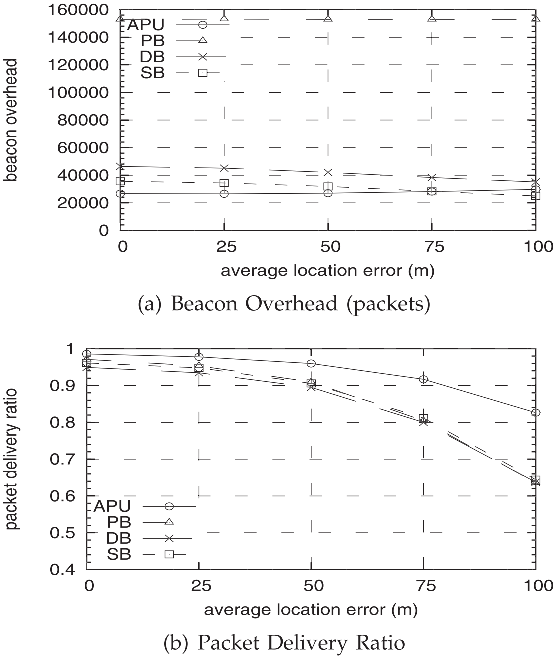



(b) Packet Delivery Ratio

Fig. 6. Impact of traffic load on the performance of beaconing schemes.

### 5.2 Impact of Traffic Load on Beaconing Schemes

In the second set of simulations, we evaluate the impact of varying the traffic load on the performance of APU and also compare APU with the three beaconing schemes under consideration. We use the same scenario as in the first set of experiments. We fix the average node speed to 15 m/s. We vary the number of flows from 5 (low load) to 25 (high load). As the number of traffic flows increase, more nodes in the network are involved in forwarding packets. Since, the ODL rule in APU aims at maintaining an accurate view of the local topology for nodes involved in forwarding packets, we expect the beacon overhead to increase with the traffic load. Fig. 6a confirms our hypothesis. On the contrary, the beacon overhead for DB and SB decrease with an increase in the traffic load. This is because, in these schemes, the beacon information is piggybacked with data packets whenever possible. When the traffic load is high, the opportunities for piggybacking increase, thus reducing the explicit transmission of beacons. However, the beacon overhead of APU is still lower than that of PB, which is constant over the traffic load variation (since we do not use promiscuous mode in PB). For low traffic load, the beacon overhead of APU is also lower than that of DB and SB. However, when the traffic load is high, DB and SB outperform APU. As seen from Fig. 6b, APU achieves better packet delivery ratio than all




(a) Beacon Overhead (packets)

(b) Packet Delivery Ratio

Fig. 7. Impact of localization error on the performance of beaconing schemes.

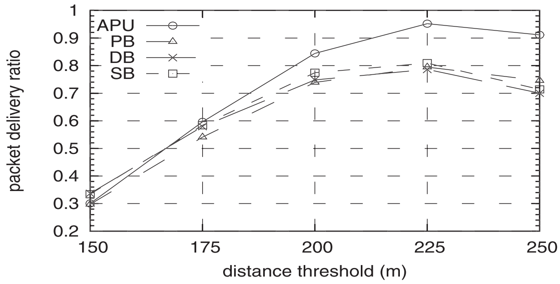



Fig. 8. Impact of fading channel on the performance of beaconing schemes.

other schemes, due to that APU can maintain a more accurate local topologies for the nodes around routing paths. Note that, the packet delivery ratio in APU, DB, and SB increases slightly with the traffic load. This is because that the larger number of data packet forwarding can piggyback more beacon information, which leads to a more accurate local topology and therefore better packet delivery ratio. However, it is expected that the packet delivery ratio will fall if the traffic load is high enough to saturate the network. Note that, we only present the key results (omitting other performance metrics) here and also in the rest of evaluations due to the page limitations.

Overall, the simulation results show that APU is significantly better at adapting to network mobility and traffic load as compared to PB, DB, and SB. The fundamental reason for this is that the beacons generated in APU are more concentrated in the network hotspots, where they are most useful in maintaining an accurate representation of the local neighborhood.

### 5.3 Impact of Localization Errors, Fading Channel, and Node Density on Beaconing Schemes

In this set of simulations, we study the performance of APU and the other three beaconing schemes in a more realistic simulation environment that takes into account several real-world effects such as localization error, fading wireless channel, and sparse node densities. Nodes move according to RDM model and the speed is randomly selected from (0, 20 m/s). The number of flows is fixed at 15. Other parameters are same as those used in previous simulations, unless explicitly noted.

First, we study the impact of localization errors on the performance of beaconing schemes. We introduce a random location error for each node. The error is random selected and fixed for each node during the lifetime of the simulation. The node only knows its erroneous location and broadcast it in the position updates. Consequently, each node has inaccurate information about the location of itself and its neighbors. We define the average location error as the mean distance from the erroneous location to the actual location. We vary the error from 0 to 100 m (in steps of 25 m) and observe their impact on the beacon overhead and packet delivery ratio. Fig. 7a shows that the beacon overhead of APU is lower than all other three schemes in most of cases. At the same time, APU can still achieve the best packet delivery ratio, as shown in Fig. 7b. The results confirm that APU can maintain a fresher topology with a less number of position updates in the presence of localization errors.

Next, we study the impact of fading channel on the performance of beaconing schemes. Note that, in all previous simulations, we have assumed the two-ray ground radio model. In this radio model, the radio coverage of each node is a perfect circle, which is often not true in real-world scenarios [26]. Therefore, in this simulation, we consider a more realistic radio model, i.e., log-normal shadowing [25], which captures the random multipath (or reflections) fading between two nodes. Due to random fading, there exists a transition region near the border of the radio coverage of a transmitter. For the nodes that lie inside this transition region, the existence of a link with the transmitter is a random variable. Further, there is also a high probability that this link exhibits asymmetry (i.e., the link may exist in one direction but not in the other) [26]. In order to cope with this issue, the authors in [26] propose *bounded distance Forwarding*, which excludes nodes in this transition region from being considered as possible forwarders. In other words, a node only includes those neighbors in its neighbor list that are located less than a certain *distance threshold* away from the node. In this set of simulation, we simulate bounded distance forwarding. We vary the distance threshold from 150 to 250 m (in steps of 25 m) and evaluate the corresponding performance of the beacon schemes. For the shadowing radio model, we assume that both the path loss rate and the standard deviation of the random signal are equal to 3. Fig. 8 shows that APU can still achieve better packet delivery ratio than other schemes, since it allows nodes to maintain a more accurate view of their local topology along the routing path. Note that comparing the different distance thresholds, the optimal performance occurs at 225 m. This is because, when the distance threshold is too small, only a few neighbors can be included in the neighbor list, which often leads to routing failure. When the distance threshold is too large, the neighbors in

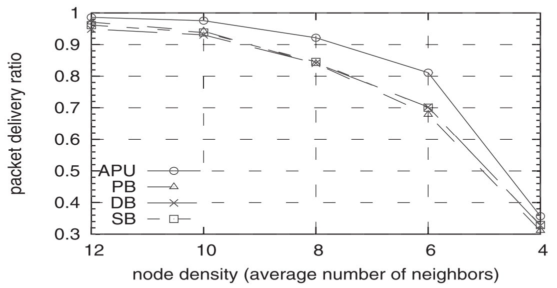



Fig. 9. Impact of node density on the performance of beaconing schemes.

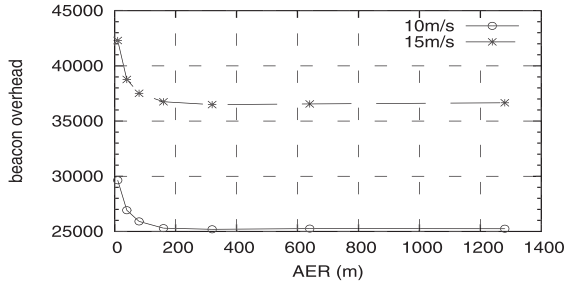
the transition region are considered as potential forwarders and the associated randomness and link asymmetry affects the performance.

Finally, we study the impact of node density on the performance of beaconing schemes. In our previous simulations, we have assumed a sufficiently dense network, such that a node can always find a neighbor that is closer to the destination than itself. In the following simulation, we evaluate the impact of sparser topology on the performance. We vary the total number of nodes in the network such that the average number of neighbors for each node varies from 12 to 4. As expected, Fig. 9 shows that the performance of all schemes degrades as the node density reduces. This is because geographic routing experiences more frequent route failures in sparser networks as forwarding nodes are more likely to not find a suitable next hop node toward the destination. However, Fig. 9 illustrates that APU can still achieve relatively higher performance than other schemes.

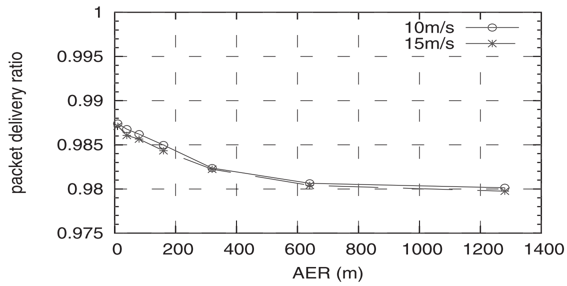
### 5.4 The Impact of AER on the Performance of APU

Recall that, in MP rule, a node $i$ sends the next beacon when the error between its predicted location and its actual location is greater than the *Acceptable Error Range*. In this section, we simulate the impact of AER on the performance of APU.

The simulation setups are similar as the ones presented in Section 5.1. We randomly select 15 communicating pairs and consider two mobility scenarios, one with average speed of 10 m/s and another with average speed of 15 m/s. Fig. 10 shows the performance of APU with varying AER from 10 to 1,280 m. As expected, when AER is 10 m (the smallest value), APU generates the highest amount of beacons (see Fig. 10a) since a smaller value of error threshold can be more frequently reached and triggers more beacon broadcast. With the increase of AER, beacon overhead is decreasing dramatically and then slowly converges to a certain value. This is because, when the AER is large enough (e.g., 720 m), MP rule is more tolerant to the location prediction errors and it rarely triggers beacon broadcast. In such case, MP component is practically disabled and the final performance depends on the position updates from ODL rule. The similar pattern is also found for packet delivery ratio, as shown in Fig. 10b. With the increase of AER, APU maintains a less accurate local topology, which leads to the decreased packet delivery ratio. However, note that the packet delivery ratio is only dropped slightly when a very large AER is used. This interesting results illustrate that, without MP component,




(a) Beacon Overhead (packets)




(b) Packet Delivery Ratio

Fig. 10. Impact of AER on the performance of APU.

ODL component can still achieve relative high performance. The reason is that ODL component is far more aggressive in triggering the position updates. Even in the absence of MP component, the first few data packets transmitted will serve as the pilots to discover the topology in ODL rule (though they may fail to reach to the destination). However, we expect that packet lost due to the absence of the MP rule will have a greater impact on TCP connections, since TCP performance is more sensitive to the packet loss, e.g., the data rate can be halved due to packet timeout. The selection of appropriate AER depends on the requirement of application. If the application aims to achieve the highest packet delivery ratio, a small value of AER (e.g., 10 m) should be selected.

## 6 Conclusions

In this paper, we have identified the need to adapt the beacon update policy employed in geographic routing protocols to the node mobility dynamics and the traffic load. We proposed the *Adaptive Position Update* strategy to address these problems. The APU scheme employs two mutually exclusive rules. 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 that are overheard from new neighbors. We mathematically analyzed the beacon overhead and local topology accuracy of APU and validated the analytical model with the simulation results. We have embedded APU within GPSR and have compared it with other related beaconing strategies using extensive NS-2 simulations for varying node speeds and traffic load. Our results indicate that the APU strategy generates less or similar amount of beacon overhead as other beaconing schemes but achieve better packet delivery ratio, average end-to-end delay and energy consumption. In addition, we have simulated the performance of the proposed scheme under more realistic network scenarios, including the considerations of localization errors and a realistic physical layer radio propagation model. Future work includes utilizing the analytical model to find the optimal protocol parameters (e.g., the optimal radio range), studying how the proposed scheme can be used to achieve load balance and evaluating the performance of the proposed scheme on TCP connections in Mobile Ad hoc Networks.

## REFERENCES


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

[2] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM MobiCom,* pp. 243-254, Aug. 2000.

[3] L. Blazevic, S. Giordano, and J.-Y. LeBoudec, "A Location Based Routing Method for Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing,* vol. 4, no. 2, pp. 97-110, Mar. 2005.

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

[5] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks," *Proc. IEEE INFOCOM,* pp. 1678-1687, June 2002.

[6] D. Johnson, Y. Hu, and D. Maltz, *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,* IETF RFC 4728, vol. 15, pp. 153-181, Feb. 2007.

[7] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing,* IETF RFC 3561, July 2003.

[8] J. Li, J. Jannotti, D.S.J.D. Couto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. ACM MobiCom,* pp. 120-130, Aug. 2000.

[9] Z.J. Haas and B. Liang, "Ad Hoc Mobility Management with Uniform Quorum Systems," *IEEE/ACM Trans. Networking,* vol. 7, no. 2, pp. 228-240, Apr. 1999.

[10] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic Routing without Location Information," *Proc. ACM MobiCom,* pp. 96-108, Sept. 2003.

[11] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient Geographic Routing in Multihop Wireless Networks," *Proc. ACM MobiHoc,* pp. 230-241, May 2005.

[12] Q. Chen, S.S. Kanhere, M. Hassan, and K.C. Lan, "Adaptive Position Update in Geographic Routing," *Proc. Int'l Conf. Comm. (ICC '06),* pp. 4046-4051, June 2006.

[13] M. Heissenbuttel, T. Braun, M. Walchli, and T. Bernoulli, "Evaluating of the Limitations and Alternatives in Beaconing," *Ad Hoc Networks,* vol. 5, no. 5, pp. 558-578, 2007.

[14] Y. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic Routing Made Practical," *Proc. Second Conf. Symp. Networked Systems Design and Implementation,* pp. 217-230, May 2005.

[15] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing," *Proc. ACM MobiHoc,* pp. 267-278, June 2003.

[16] B. Blum, T. He, S. Son, and J. Stankovic, "IGF: A State-Free Robust Communication Protocol for Wireless Sensor Networks," technical report, Dept. of Computer Science, Univ. of Virginia, 2003.

[17] M. Zorzi and R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance," *IEEE Trans. Mobile Computing,* vol. 2, no. 4, pp. 349-365, Oct.-Dec. 2003.

[18] M. Heissenbuttel et al., "BLR: Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks," *Computer Comm.,* vol. 27, pp. 1076-1086, July 2004.

[19] P. Casari, M. Nati, C. Petrioli, and M. Zorzi, "Efficient Non Planar Routing around Dead Ends in Sparse Topologies Using Random Forwarding," *Proc. IEEE Int'l Conf. Comm. (ICC),* pp. 3122-3129, June 2007.

[20] S. Basagni, M. Nati, C. Petrioli, and R. Petroccia, "ROME: Routing over Mobile Elements in WSNs," *Proc. 28th IEEE GlobeCom,* pp. 5221-5227, Dec. 2009.

[21] P. Nain, D. Towsley, B. Liu, and Z. Liu, "Properties of Random Direction Models," *Proc. IEEE INFOCOM,* pp. 1897-1907, Mar. 2005.

[22] C. Bettstetter, H. Hartenstein, and X. Prez-Cos, "Stochastic Properties of the Random Waypoint Mobility Model," *Wireless Networks,* vol. 10, no. 5, pp. 555-567, Sept. 2004.

[23] Q. Chen, S.S. Kanhere, and M. Hassan, "Mobility and Traffic Adaptive Position Update for Geographic Routing," Technical Report UNSW-CSE-TR-1002, School of Computer Science and Eng., Univ. of New South Wales, ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/1002.pdf, 2010.

[24] L.M. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," *Proc. IEEE INFOCOM,* pp. 1548-1557, 2001.

[25] C. Bettstetter, "Connectivity of Wireless Multihop Networks in a Shadow Fading Environment," *Wireless Network,* vol. 11, no. 5, pp. 571-579, 2005.

[26] G. Zhou, T. He, S. Krishnamurthy, and J.A. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," *Proc. ACM MobiSys,* pp. 125-138, 2004.


**Quanjun Chen** received the BSc degree in computer science from the Civil Aviation University of China (CAUC), Tianjin, in 1999 and the PhD degree in computer science from the University of New South Wales, Sydney, Australia, in 2010, respectively. He is currently with the Australian Centre for Field Robotics, University of Sydney, Australia. His research focuses on performance analysis and networking protocol design in wireless ad hoc networks. He is a member of the IEEE.

**Salil S. Kanhere** received the BE degree in electrical engineering from the University of Bombay, India, in 1998 and the MS and PhD degrees in electrical engineering from Drexel University, Philadelphia, in 2001 and 2003, respectively. He is currently a senior lecturer in the School of Computer Science and Engineering at the University of New South Wales, Sydney, Australia. His current research interests include wireless sensor networks, vehicular communication, mobile computing, and network security. He is a senior member of the IEEE and the ACM.

**Mahbub Hassan** received the BSc degree in computer engineering (with high honor) from Middle East Technical University, Turkey, in 1989, the MSc degree in computer science from the University of Victoria, Canada, in 1991, and the PhD degree in computer science from Monash University, Melbourne, Australia, in 1997. He is a full professor in the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia, where he leads a research program on mobile and wireless systems. He has coauthored several books which are referenced widely in advanced computer networking courses offered by universities throughout Europe, America, and Asia. He serves on the editorial advisory board of *Computer Communications* (Elsevier Science). In 1999-2001, he served as an associate technical editor for *IEEE Communications Magazine* and was a guest editor of the magazine's feature topic on TCP performance in future networking environments (April 2001 issue) and wireless mesh networks (November 2007 issue). He has written several successful Australian Research Council (ARC) grants, worked on collaborative R&D projects with large industrial research laboratories, and developed industry short courses on leading edge networking topics. His other recent appointments include invited professor at the University of Nantes, France, in 2005 and project leader and principal researcher at the National ICT Australia (NICTA) in 2005-2006. He is a senior member of the IEEE.