

TO-GO: TOpology-assist Geo-Opportunistic Routing in Urban Vehicular Grids

Kevin C. Lee, Uichin Lee, Mario Gerla
University of California
Department of Computer Science
Los Angeles, CA 90095
{klee,ucllee,gerla}@cs.ucla.edu

Abstract—Road topology information has recently been used to assist geo-routing, thereby improving the overall performance. However, the unreliable wireless channel nature in urban vehicular grids (due to motion, obstructions, etc) still creates problems with the basic greedy forwarding. In this paper, we propose TO-GO (TOpology-assisted Geo-Opportunistic Routing), a geo-routing protocol that exploits topology knowledge acquired via 2-hop beaconing to select the best target forwarder and incorporates opportunistic forwarding with the best chance to reach it. The forwarder selection takes into account of wireless channel quality, thus significantly improving performance in error and interference situations. Extensive simulations confirm TO-GO superior robustness to errors/losses as compared to conventional topology-assisted geographic routing.

I. INTRODUCTION

Geographic routing¹ has become one of the popular routing protocols in a *vehicular ad hoc network* (VANET) for two reasons. First, geo-routing is stateless, thus obviating the need of route discovery and maintenance procedures as in conventional on-demand and table-driven routing protocols, which is very costly in highly mobile vehicular environments such as VANETs. Second, with Navigation System/GPS devices getting cheaper and becoming a common add-on, it is becoming easier to support geo-routing.

In urban grids, it is known that conventional geo-routing protocols such as *Geographic Perimeter Stateless Routing* (GPSR) [1] may not work well due to the unique layout of road structure. Given this, Lochert et al. recognized road segments and proposed *Geographic Perimeter Coordinator Routing* (GPCR) [2], by incorporating the road topology in the geo-routing. The key distinction is that nodes in junctions (or intersections) decide to which junction a packet must be forwarded.

In our previous work, we proposed an enhancement to GPCR, called GpsrJ+ (Gpsr Junction+) [3]. The key idea is to note that not every packet must be stored and forwarded by a junction node; in other words, the junction is not a necessary stop. More precisely, a packet must be stored/forwarded by

a junction node only when it needs to make a left/right turn at that junction. This greatly reduces the dependency on junction nodes. In the previously published GpsrJ+ scheme, a forwarding node uses two-hop neighbor information to better estimate the path and detect advantageous junction turns. Upon learning that there are no advantageous turns, GpsrJ+ simply bypasses the junction. This two-hop prediction reduces hop counts, increases the packet delivery ratio, and obviates the need to distinguish junction nodes from ordinary nodes.

However, existing geo-routing protocols for urban grids do not consider the high urban wireless channel error rates due to multi-path fading and shadowing, common characteristics in vehicular networks. Clearly, the assumption of an unit disc propagation does not hold well. In existing urban geo-routing approaches the packet is greedily forwarded to the furthest node to the destination. However, the further the transmission range, the higher the attenuation and the more likely the packet is to be lost.

This observation brings forth the concept of opportunistic routing [4], [5]. In opportunistic routing, a sender can take advantage of random packet receptions in its neighboring nodes due to the error-prone wireless channel and of opportunistic forwarding by a subset of the neighbors that received the packet correctly. A key question is to decide at each node which neighbors should be potential relay nodes to reach a destination. ExOR and Least Cost Opportunistic Routing (LCOR) used ETX based table-driven methods to find such a set. In geo-routing, Shah et al. proposed to use a lens shape *forwarding region* that is the intersection between a circle centered at a forwarding node and that centered at the destination [6]. Contention-based forwarding (CBF) algorithms (e.g., [7]), which are not originally designed for opportunistic routing, use a similar forwarding shape. Nodes in a forwarding region contend for packet forwarding based on a distance based timer. These geo-routing protocols ([7], [8]), however, always use the radius between the forwarding node and destination in order to find the region and set the timer.

This concept can also be exploited in geo-routing by finding a local *forwarding set* that maximizes the Expected Packet Advancement (EPA) to the destination. In general, this approach does not guarantee optimality as in LCOR, but it

This research is supported by the International Technology Alliance sponsored by the U.S. Army Research Laboratory, the U.K. Ministry of Defense under Agreement Number W911NF-06-3-0001, and by ARMY MURI under funding W911NF0510246.

¹In the rest of the paper, “geographic routing” will be shorthanded as “geo-routing.”

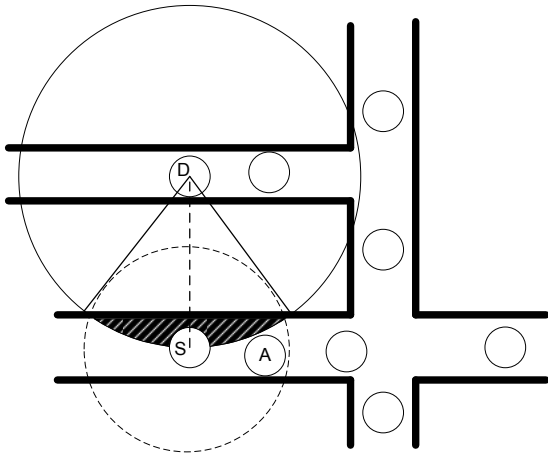


Fig. 1. The shaded area is the forwarding region established between S and D ; however, there is no node in this region. The best next forwarding node should be A instead.

attempts to reduce the number of transmissions (i.e., average hop count to the destination). Zeng et al. formally showed that the larger the number of nodes with positive advancement to the destination, the larger the average packet advancement to the destination [9]. In existing schemes, to realize this, a distance-based timer is used: the further the distance from the sender, the shorter the packet expiration timer [7], [8], [6]. Lower priority nodes will cancel their impending transmissions when they hear a higher priority transmission. Moreover, to guarantee this packet suppression (i.e., to guarantee that nodes in a forwarding set can hear one another) under unit disk propagation, the forwarding set must be reduced to a lens shape *forwarding region* that is the intersection between a circle centered at a forwarding node and that centered at the destination [6], [7].

The drawback of the defined forwarding region is that it often yields a suboptimal set in terms of its size and packet advancement. Since the destination does not usually lie on the same road as the current forwarding node in an urban environment, the set can be a lot smaller. Because the set can be a lot smaller, the current forwarding node can forward to a node that does not make maximum packet advancement; even worse, there may be nothing in the set to choose a node which the current forwarding node can forward to. Consider Figure 1, the forwarding region set by the the destination D and the forwarding node S is the shaded area. However, there is no node in this area. Even if there were, node A that makes the maximum packet advancement is not chosen.

In this paper, we exploit GpsrJ+’s 2-hop neighbor information, to further enhance opportunistic forwarding by making the following contributions:

- Unlike previous approaches where a forwarding region is defined between the current sender and the destination, we can find a more effective forwarding set² between the

²We use a forwarding set instead of a forwarding region because of the irregular radio range which makes the geometric representation of such a region difficult to represent in mathematical form.

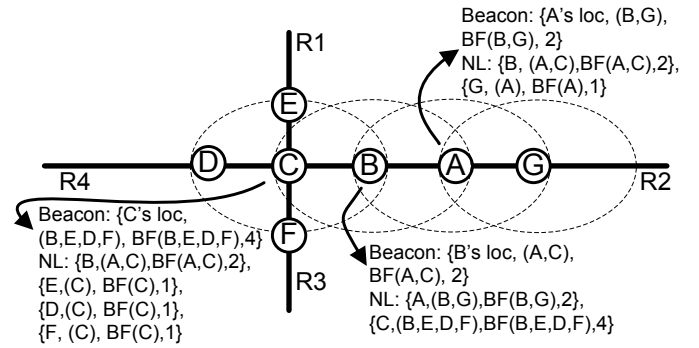


Fig. 2. Enhanced beacon and neighbor list. Node C for example, beacons its location, its furthest neighbors, and all its neighbors in a Bloom filter. Node B ’s neighbor list keeps its neighbors A and C , their furthest neighbors, and their Bloom filters.

sender and the *target node*.

- Given that the unit-disk propagation assumption does not generally hold in urban grids, instead of the lens shape, we use the actual “intersection” of neighbors made available by GpsrJ+’s 2-hop neighbor information.
- Simulation results confirm that TO-GO can effectively avoid poor wireless links and is thus robust to channel impairments. For instance, in some error-prone wireless channel situations, TO-GO achieves up to 98% packet delivery ratio, which is 40% higher than GpsrJ+.

The rest of this paper is organized as follows: Section II illustrates our enhanced beacon as a basis for junction prediction and forwarding set selection. Section III describes TO-GO design in aspects of junction prediction, forwarding set selection, timer setting, retransmission, duplicate management. Section IV provides experiment results. Section V provides a panoramic view of geographic, opportunistic, and geo-opportunistic routing. Section VI summarizes our contribution and concludes the paper.

II. ENHANCED BEACON

In the two-hop beacon strategy of GpsrJ+ [3], the conventional hello beacon is augmented to include the furthest neighbors (and their locations) in each direction on the urban map (typically, only two neighbors except for intersection nodes). This is required to support junction forwarding prediction in both greedy and perimeter modes. Furthermore, the beacon would also contain a Bloom filter, a space efficient membership checking data structure, of all node’s neighbors and its size. This enables the construction of a forwarding set while keeping the broadcast overhead at a minimum. Upon receiving a beacon, a node would have a neighbor list that contains its neighbor, every neighbor’s furthest neighbors, and a Bloom filter of their neighbors and its size.

Figure 2 demonstrates beacons and neighbor lists of Node A , B , and C . Node A broadcasts its location, its furthest neighbors B and G , and a Bloom filter containing B and G with size 2. Node B broadcasts its location, its furthest neighbors A and C , and a Bloom filter including A and C with size 2. Node C broadcasts its location, its furthest

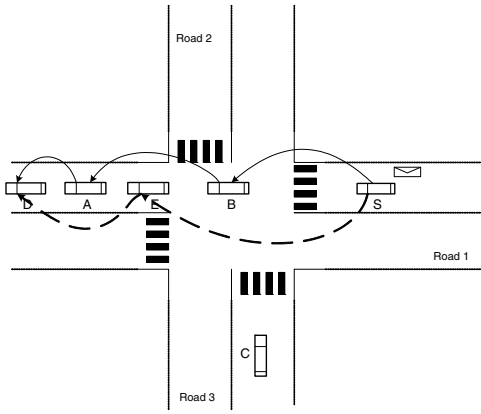


Fig. 3. Dashed arrows are TO-GO and solid arrows are GPCR.

neighbors B , D , E , and F , and a Bloom filter including B , D , E , and F with size 4. After receiving beacons from its neighbors, Node A 's neighbor list will have B and G , their furthest neighbors (A, C) and (A), their Bloom filters, and filters' size, respectively. Node B 's neighbor list will have A and C , their furthest neighbors (B, G) and (B, E, D, F), their Bloom filters, and their size. Similarly, Node C 's neighbor list will have B , D , E , and F , their furthest neighbors (C), their Bloom filters, and size (1), respectively.

The storage overhead is $O(n)$ where n is the number of nodes in the network. The overhead is the same as if each node did not keep its neighbors' furthest neighbors, its neighbors' Bloom filters, and the size of their Bloom filters. Because a node's furthest neighbors are at most four in a typical urban environment with four intersecting roads and the size of the Bloom filters and their count are simply constants, the storage overhead is on the same order of n as before.

III. TO-GO DESIGN

In TO-GO routing, a packet is marked for a target node, determined by the *Next-hop Prediction Algorithm* (NPA), and broadcasted. Nodes at or close to the target node will receive the packet. Nodes that are in the set determined by the Forwarding Set Selection (FSS) set their timer based on their relative distance to the target. The closer the distance to the target, the sooner the timer goes off and the sooner the packet gets forwarded. As the next node successfully broadcasts, the other contending nodes will cancel their timer. This in turn prevents redundant packet transmissions.

A. Next-hop Prediction Algorithm

TO-GO uses the aforementioned enhanced beacon to predict the target node: either the furthest node or the *junction node*. Here, a junction node is a node that is located at the junction and can forward packets to any directions. In the greedy mode, the best forwarding node is the furthest node when its neighboring junction node's neighbor closer to the destination lies on the same road segment as the furthest node; i.e., a packet will not make left/right turns at the junction. Otherwise, the best forwarding node is the junction node. The two-hop

information in enhanced beacons enables TO-GO to make an advanced decision on whether to bypass the junction node. In the appendix, we provide a proof that keeping a node's furthest neighbors in the beacon (as opposed *all* the neighbors of a node) is sufficient to determine whether the best forwarding node is either the furthest node or the junction node.

Figure 3 illustrates the advantage of prediction. The figure shows that TO-GO can bypass the junction area and forward the packet to node E directly, yet GPCR forwards it to the junction node B , thus causing more transmissions. In the perimeter mode [3], TO-GO uses the right-hand rule to determine the best direction (as opposed to final destination direction) and thereby the best forwarding node. That is, if the furthest node is in the same direction as the best direction, the best forwarding node is the furthest node; otherwise, the best forwarding node is a junction node.

B. Forwarding Set Selection

After finding the target node, the current forwarding node C must determine which nodes will be in a forwarding set. In principle, the forwarding set should be selected such that nodes in the set can hear each other to prevent hidden terminal collisions. A brute force algorithm to find a forwarding set in which nodes hear one another is analogous to finding a maximal *clique* in which every node has a connection to every other node. Such a problem is NP-complete. We propose a simplified scheme to obtain an approximate forwarding set by first eliminating C 's neighbors that cannot hear the target node. Out of the neighbors that remain, we then pick the neighbor that has the most number of neighbors. Denote this neighbor as M . For each neighbor N of the current forwarding node, test its membership in M 's Bloom filter. If N is in the Bloom filter and N 's Bloom filter contains M , test N 's membership using the Bloom filters of existing elements in the forwarding set. If N is in the Bloom filters of all these elements, add N to the set. Continue adding such N until all the neighbors of C have been checked. The algorithm takes $O(n^2)$ where n is the number of C 's neighbors.

The intuition behind the approximate algorithm is that the neighbor M that has the most neighbors is in the most dense area. Despite irregular and different radio ranges, nodes selected from that region are more likely to have one another as neighbors. The forwarding set produced thus should be close to a maximal set that provides largest number of nodes as potential next hop forwarders.

Since it is expensive to keep the resulting forwarding set in the data packet, the forwarding set is kept as a Bloom filter. Upon receiving the packet, a node drops the packet if its address is not in the Bloom filter, or if its current location does not make a positive advancement to the target node.

From the practical standpoint, the elimination of "hidden terminals" in the forwarding set is very expensive in terms of processing time and line overhead. On the other hand, the presence of hidden terminals may be rare, especially when node density is low, when *irregular* radio range is obstructed by roads [10], and when the forwarding region

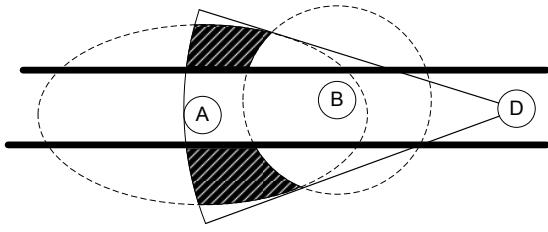


Fig. 4. The shaded area depicts what would have been the region where nodes cannot hear B . But since there is road obstruction, irregular radio range, and the forwarding region formed by A and B , the hidden terminal area does not happen.

is no longer determined by the radius formed by the current forwarding node and the destination (but the target node) as shown in Figure 4. Thus, one may relax the hidden terminal suppression in the forwarding set. This may lead to some loss of performance in the results. In future experiments we will evaluate the trade offs between cost of hidden terminal detection and performance improvements.

C. Priority Scheduling

Having found the forwarding set, we want a node closer to the target node to become the next forwarder. Ideally, we want the target node to receive the packet. To this end, we set the timer T as follows:

$$T = C \times \frac{\text{dist}(\text{receiving node}, \text{target node})}{\text{dist}(\text{sending node}, \text{target node})}$$

where C is the maximum forwarding delay that varies with the transmission rate and the processing time. Setting T too high results in poor system throughput. Conversely, setting it too low results in packet duplication. It is possible that two nodes A and B that either are close to each other or have the same distance to the target node have similar T such that if A goes off first, B will go off as well since the time it takes to suppress B is longer than the time B waits before its timer goes off. A practical solution to eliminate packet duplication as a result of similar T is to impose an additional constraint in selecting the forwarding set FS by excluding a node N_k such that $|T_{N_k} - T_{N_i}| < \delta$, where δ is the *minimum* time interval needed for suppression, for all nodes $N_i \in FS$.

The shorter the distance between the receiving node and the target node, the greater the progress, therefore, the shorter the timer. Unlike the timer formula in [7], this distance between the receiving node and the *target node* is normalized by the radio range, not by the distance between the sending node and the *target node* under the realistic assumption that radio range differs from vehicle to vehicle.

D. Retransmission

For reliability, a node will rebroadcast the packet if it does not overhear a packet forwarded by one of the nodes in its forwarding set. TO-GO limits a node to only rebroadcast three times before dropping the packet, but this number can be configured based on the channel conditions. The retransmission timer RT is set as follows:

$$RT = 2 \times Prop + C$$

The two propagation delays account for the time it takes for the broadcast to reach the desired node, either close to the target or at the target, and to come back. The maximum timer delay is C which occurs when the receiving node is at the same location as the retransmitting node.

If the node experiences repeated timeouts, it then selects a specific forwarder using the conventional greedy algorithm (e.g., best advancement promise) and retransmits the packet to that forwarder. This simple scheme recovers from packet loss (due to collision) and prevents possible deadlocks in case of hidden terminals.

E. Removing Duplicates at the Destination

When the packet finally arrives at the destination, the destination is responsible for notifying its neighbors that it has received the packet with a certain sequence number. In that way, the destination will not have to receive multiple copies of the same packet from those neighbors who hold the packet and whose timer has not gone off. The destination notifies its neighbors by rebroadcasting the message it has received. For nodes that hold the packet, they will check if the previous node of the received packet is the destination. If it is, they cancel the timer and remove the packet. If it is not, their timer continues to count off to trigger broadcast. For nodes that do not hold the packet, the broadcast is simply dropped if the previous node of the received packet is the destination; otherwise, the timer goes off as specified.

IV. PERFORMANCE EVALUATION

A. Experimental Setup

The experiments were conducted on Qualnet simulator 3.95 with IEEE 802.11b DCF as the MAC with a transmission rate of 2Mbps and transmission range of 250m. Nodes on different roads cannot talk to each other because of obstacles (trees, buildings, etc.) unless two roads share the same extension in either the horizontal or vertical direction.

The mobility traces were generated on an urban grid of 1800m by 300m using VanetMobiSim [11], an open source and freely available realistic vehicular traffic generator for network simulators. VanetMobiSim's functionalities are decomposed into macro- and micro- mobility features of a vehicular environment to produce realistic urban mobility traces. The macromobility part is composed of motion constraints and a traffic generator, while the micro-mobility part controls cars acceleration and deceleration in order to keep a safe inter-distance and avoid accidents and overlapping.

All intersections are controlled by stop signs and all road segments contain speed limitations. Unless specified differently, all roads have a single lane and a speed limit of 15m/s (54 km/h). Finally, the micro-mobility is controlled by the IDM-IM4, an extension to the Intelligent Driver Model (IDM) considering intersections.

Parameter	Value
Network simulator	Qualnet 3.95
Mobility simulator	VanetMobisim
Dimension	1800m × 300m
CBR rate	1460 bytes/second
802.11b rate	2 Mbps
802.11b transmission power	15.0 dBm
TX range	250m
Propagation	Free Space with inter-road radio blocking model
Avg vehicle speed	50 km/hr
Simulation runs	20
Confidence interval	95%
Number of nodes	75 to 150

TABLE I
SIMULATION PARAMETERS

With the aforementioned setup, we ran three experiments. The first experiment obtains the optimal value of C , the constant in the timer equation, to maximize throughput. Using the optimal C in the timer equation, the second experiment then compares GPSR, GPCR, GpsrJ+³, and TO-GO in packet delivery ratio (PDR), hop count, and latency in an error-free wireless channel. The data were gathered for node mobility traces of 75 to 150, with 25-node increment. For each node trace, there were 20 simulation runs, with each running for 180 seconds. Each run consists of random source-destination pairs using 1460-byte constant bit rate (CBR), an UDP-based packet generation application. The random source-destination pairs are generated every 10 seconds; one pair is allowed to transmit each 10-second interval. Table I summarizes the simulation parameters.

The setup for the third experiment is the same as the second experiment. However, channel fading and errors were introduced to show the robustness of TO-GO. Since channel fading is as a result of distance, we model channel errors based on the equation of path loss (PL) in [12]:

$$PL(d)[dB] = PL(d) + X_\sigma = PL(d_0) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (1)$$

where n is the path loss exponent which indicates the rate at which the path loss increases with distance, d_0 is the close-in reference distance determined from measurements close to the transmitter, d is the transmitter-receiver distance, X_σ is a zero-mean Gaussian distributed random variable with standard deviation σ to account for random and distributed log-normally path loss.

According to qualnet, the $PL(250m) = -80$ dB. We used $n = 2$ for free space exponent and solved for d_0 . After solving for d_0 , eqn. (1) becomes:

$$PL(d)[dB] = PL(d) + X_\sigma = 20 \log\left(\frac{d}{0.025}\right) + X_\sigma \quad (2)$$

Based on eqn. (2), we can calculate the path loss of any receiving node given its distance to the transmitter. If the path

³We enhanced GpsrJ+ further by enabling it with junction-prediction in both the greedy and perimeter mode.

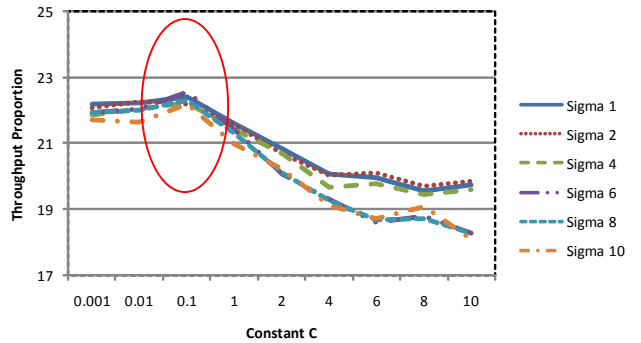


Fig. 5. Throughput Proportion vs. C

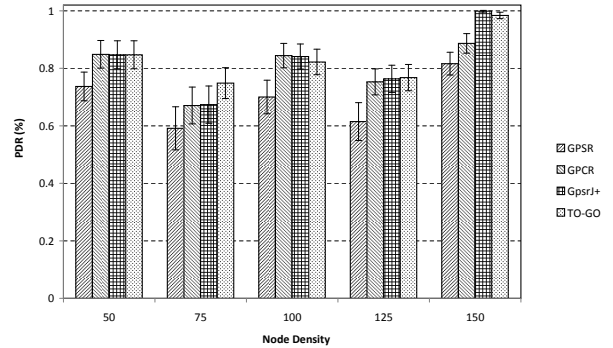


Fig. 6. PDR vs. Node Density

loss is greater than -80 dB, the routing protocol will accept the packet; otherwise, it drops the packet. We vary the standard deviation σ of the zero-mean Gaussian distributed random variable X to simulate different magnitudes of shadowing effect and thereby different probabilities of packet dropping.

B. Experimental Results

1) *Optimal C* : Figure 5 shows the *throughput proportion* with respect to different constant C in the Timer T equation. The throughput proportion is obtained by dividing the PDR by the end-to-end latency. The value is proportional to the throughput. We used 150-node trace to obtain the optimal C , which is 0.1 in different channel qualities set by σ . The higher the σ , the lossy the channel, therefore, the higher the error. Throughput proportion does not vary much with σ , showing the robustness of TO-GO. We use 0.1 for C throughout the rest of experiments.

2) *Error-Free Wireless Channel*: Figure 6, 7, and 8 show the PDR, the hop count, and latency of GPSR, GPCR, GpsrJ+, and TO-GO with respect to node density in an error-free wireless channel. σ in eqn. (2) is set to 0 to model 0% dropping probability. 95% confidence interval is indicated for the mean value. A superficial observation indicates that while GPCR, GpsrJ+, and TO-GO are almost similar to one another in PDR, GPSR always lags behind. The performance hit is due to making “baby steps” in perimeter mode. As node density increases, the frequency of perimeter mode decreases. GPSR’s PDR gradually increases to about 82%.

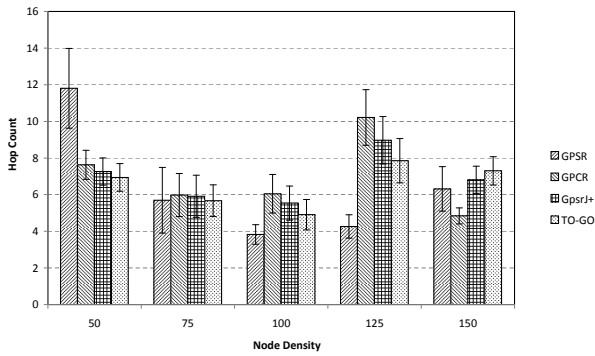


Fig. 7. Hop vs. Node Density

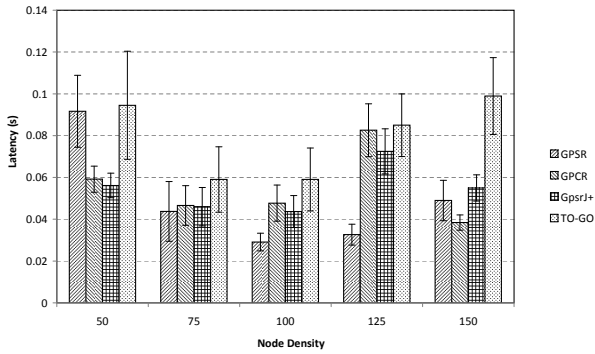


Fig. 8. Latency vs. Node Density

One can see the reduction in hop count due to prediction comparing GPCR with either GpsrJ+ or TO-GO for node density between 50 and 125. At node density 150, however, GPCR's hop count becomes the lowest among all four protocols. GPCR's hop count is less than GpsrJ+ and TO-GO because GPCR's PDR is relatively lower. The low PDR is due to GPCR's always forwarding to junction nodes, hampering GPCR from utilizing nodes beyond junctions. As a result, the longer hop stretch makes it more likely for packets to be dropped before reaching the destination. GPCR's hop count is less than GPSR's because unlike GPSR, GPCR would always make large strides towards the forwarding direction (except when there is a junction node) in both greedy and perimeter mode. GPCR's higher PDR than GPSR's verifies this claim.

As much as TO-GO shows good PDR and hop count, the end-to-end latency is the highest among all four protocols. The reason for the long delay is every TO-GO node broadcasts and since broadcast can be lost, retransmission of the same packet will inadvertently prolong the time it takes for the packet to reach the destination. In low node density such as 50, broadcast is resent because the sparsity of nodes decreases the opportunity for packets to reach nodes other than the target. If the target node failed to receive the broadcast, the sending node would not know until retransmission timer expires. This prolongs the latency. The delay also increases gradually from node density 75 to 150 because more nodes introduce collisions. Broadcast becomes less likely to reach the target node as it collides with periodic beacons and broadcast

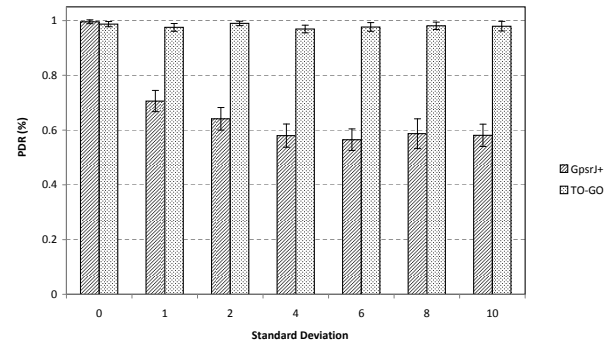


Fig. 9. PDR vs. Standard Deviation, σ in Wireless-Prone Channel

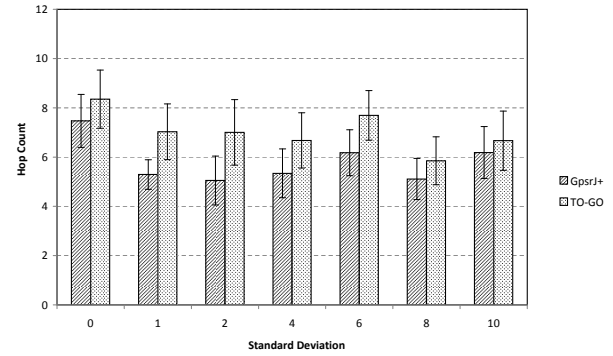


Fig. 10. Hop vs. Standard Deviation, σ in Wireless-Prone Channel

from other nodes.

3) *Error-Prone Wireless Channel*: Errors are introduced into the channel artificially by varying the standard deviation σ of the Gaussian distributed random variable X in eqn. (2). As the standard deviation becomes larger, the error in the channel increases. We only compared the performance of GpsrJ+ and CBF because GpsrJ+ is an enhancement of GPCR and GpsrJ+ outperformed GPCR.

Figure 9, 10, and 11 show the PDR, the hop count, and latency of GpsrJ+ and TO-GO. The x -axis indicates the standard deviation and relates directly to the error probability. When the error increases, TO-GO maintains the PDR above 96% but GpsrJ+ keeps on dropping. AT $\sigma = 10$, TO-GO's PDR remains at 98% while GpsrJ+'s PDR drops to 58%.

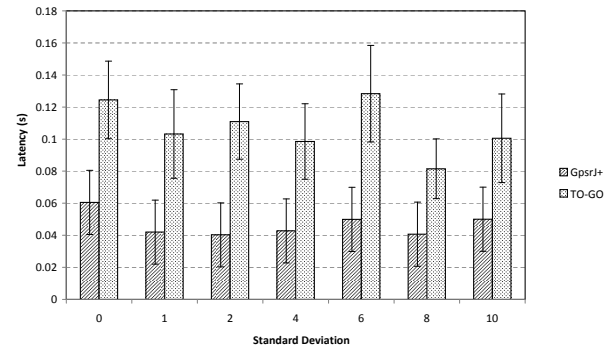


Fig. 11. Latency vs. Standard Deviation, σ in Wireless-Prone Channel

The hop count of TO-GO hovers between 5.8 and 8.4 and the latency of TO-GO hovers between 0.08s and 0.12s. In addition to stable PDR, the bounded hop count and latency further show TO-GO's robustness in using neighbors nearby the target to deliver opportunistically. When there are more nodes in the network, it helps TO-GO's cause because there are more opportunities for packets to be delivered to nodes closer to the target even though the target is not able to receive the packet.

The relatively higher hop count and latency in TO-GO from $\sigma = 1$ to $\sigma = 10$ is due to averaging these values which are not accounted for in GpsrJ+ because packets are dropped. Another reason for higher latency is because broadcast and periodic contentions cause frequent retransmissions that prolong end-to-end latency. This can be seen when $\sigma = 0$ where TO-GO's hop count is slightly higher than GpsrJ+ and its latency is twice as high while the two have similar PDR.

V. RELATED WORK

A. Geographic Routing

Lochert *et al.* [2] proposed GPCR, a solution that does not rely on planarization of nodes by taking note of the fact that an urban map naturally forms a planar graph. In the recovery mode, nodes simply forward along a road in the direction produced by the right-hand rule [1] until hitting nodes at a junction. The junction nodes will determine the next hop and the new direction of the perimeter to forward packets to.

GpsrJ+ enhances GPCR by noting that nodes do not necessarily need to stop at each junction node. GpsrJ+ [3] improves GPCR in delivery ratio and hop count by avoiding unnecessary forwarding to junction nodes. Since a city map is given, GpsrJ+ does not depend on the heuristics of low density in a junction to determine junction nodes.

Ma *et al.* [13] presented a path pruning algorithm that exploits the channel listening capability to reduce the number of hops in the perimeter mode. Since the approach requires planarization of nodes, it differs from GPCR and GpsrJ+ that use the underlying map as the planar graph. In GPCR and GpsrJ+, the perimeter forwarding is greedy (i.e., the furthest node that is along the road segment) in between junctions. Because of the greedy approach in the perimeter mode, there are not so many redundant paths that can be pruned. Near junctions, GPJR further determines whether it can pass through its junction neighbor if it is not at a critical junction. The hop saving at junction nodes happens in the first pass in GpsrJ+ as opposed to the second pass in [13]. Furthermore, unlike the path pruning algorithm, GpsrJ+ does not require each node to keep states. This may not work so well in highly mobile VANETs.

B. Opportunistic Routing

ExOR [4] operates on batches of packets. The source node includes a list of candidate forwarders in each packet, prioritized by ETX to the destination. Receiving nodes buffer successfully received packets and await the end of the batch. The highest priority forwarder then broadcasts the packets in

its buffer. The remaining forwarders then transmit in order, sending only packets which were not acknowledged by the higher priority nodes. LCOR [5] also uses ETX as the link cost to assign and prioritize a list of candidate forwarders such that the expected cost of forwarding a packet to the destination is minimized. While LCOR argues for a converging least cost opportunistic route, in VANET, finding such a route is overly expensive because of mobility. TO-GO combines opportunistic and geo-routing, relying on their robustness and delivery guarantee, respectively while keeping overhead at minimum.

LCOR [5] also uses ETX as the link cost to assign and prioritize a list of candidate forwarders such that the expected cost of forwarding a packet to the destination is minimized. While LCOR argues for a converging least cost opportunistic route, in VANET, finding such a route is overly expensive because of mobility. TO-GO combines opportunistic and geo-routing, relying on their robustness and delivery guarantee⁴, respectively while keeping overhead at minimum.

C. Geographic Opportunistic Routing

Unlike ExOR and LCOR, geographic opportunistic routing is stateless; i.e., it does not require global topology information (i.e., ETX values). Geographic opportunistic routing tries to maximize the expected advancement to the destination or the Expected Packet Advancement [9], and thus, minimizes the total number of transmissions. In CBF [7], data packets are broadcast to all direct neighbors and the neighbors themselves decide if they should forward the packet. The actual forwarder is selected by a distributed timer-based contention process which allows the most-suitable node to forward the packet and to suppress other potential forwarders. GeRaF [8] prioritizes forwarders based on their distance to the destination using RTS/CTS solicitation to candidate forwarders. Shah *et al.* used a simple RTS/CTS without geographic prioritization [14], which is a must to reduce the number of transmissions. Unlike these approaches, TO-GO sets its timer *not* based on its distance to the *destination* but the distance to the *target node* and the sending node. Thanks to enhanced beaconing with two-hop information, TO-GO does not rely on a complex radio geometry to determine an *approximate* forwarding set.

Zeng *et al.* [15] discusses Geographic Opportunistic Routing's potential in maximizing throughput. They propose a local metric named *expected one-hop throughput* (EOT) to balance the tradeoff between the benefit of packet advancement and transmission reliability and the cost of medium time delay. The EOT metric assumes that each node knows the packet reception ratio of its neighbors. The metric is formulated based upon the observations that that the candidate relay priority⁵, candidate set selection⁶, and candidate coordination⁷ impact throughput. By embracing EOT metric to refine the

⁴The delivery guarantee is under the assumption of a fully connected graph.

⁵Nodes with higher packet advancement having higher priority does not necessarily increase throughput.

⁶A large candidate set does not necessarily increase throughput.

⁷Setting inaccurate timer can reduce throughput.

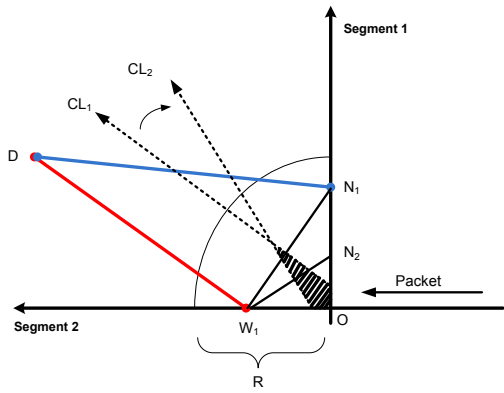


Fig. 12. $\overline{W_1D} < \overline{N_1D}$.

list of candidate forwarders, TO-GO can maximize end-to-end throughput.

VI. CONCLUSION

In this paper, we proposed TO-GO, a geographic opportunistic routing protocol that exploits road-topology information and opportunistic packet reception to increase the packet delivery ratio. As the goal is to maximize the expected packet advancement to the destination, TO-GO defines a candidate forwarding set between the current sender and the target node selected by a simple junction prediction algorithm with topology information and enhanced beaconing. The forwarding set is then adjusted to reduce packet duplication and collision. We validated the robustness of TO-GO under severe wireless channel errors via extensive simulations.

APPENDIX

We prove formally why it is sufficient to know a junction node's furthest nodes on different road segments to determine the best forwarding node. This knowledge greatly reduces the overhead inherited from the two-hop neighbor beacons. Furthermore, it produces the accurate prediction strategy that mimics exactly the forwarding behavior of a junction node in GPCR.

The theorem can be applied repeatedly for more than two furthest nodes. Without loss of generality, assume that a junction node has two furthest nodes on two road segments. We introduce and prove the theorem below:

Theorem 1. *Let W_1 and N_1 be the furthest nodes of O shown in Figure 12. It is sufficient to compare $\overline{W_1D}$ and $\overline{N_1D}$ to determine which road segment O will forward packets to. If $\overline{W_1D} < \overline{N_1D}$, O will forward to Segment 1, otherwise, O will forward to Segment 2.*

Proof: Let CL_1 be the center line for $\overline{W_1N_1}$ as shown in Figure 12. If $\overline{W_1D} < \overline{N_1D}$, D must be on the left-hand side of CL_1 . Since N_1 is the furthest node on Segment 1, any node on that segment will be closer to O than N_1 . Let N_2 be such a node and $\overline{N_2O} < \overline{N_1O}$. We draw a center line CL_2 for $\overline{W_1N_2}$. We observe that D is on the left-hand side of CL_2 . In fact, for any N_i closer than N_1 to O on Segment 1, the

center line CL_i is in the *clockwise* rotation of CL_1 . In other words, if $\overline{W_1D} < \overline{N_1D}$, then $\overline{W_1D} < \overline{N_iD}$ for all N_i where $\overline{N_iO} < \overline{N_1O}$. Therefore, it is sufficient to pick the furthest node N_1 and compare its distance to D with $\overline{W_1D}$.

If D is in the shaded area in Figure 12, it does not matter which road segment O forwards packets to because O can reach D in one hop. Forwarding to O and then to D will take the same number of hops as forwarding to the furthest node of the current forwarding node and then to D . In other words, when D is in the shaded area, having prediction or not does not make a difference.

If $\overline{W_1D} > \overline{N_1D}$, one can also show symmetrically that it is sufficient to pick the furthest node W_1 and compare its distance to D with $\overline{N_1D}$. ■

REFERENCES

- [1] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Mobile Computing and Networking*, 2000, pp. 243–254.
- [2] C. Lochert, M. Mauve, H. Füssler, and H. Hartenstein, "Geographic routing in city scenarios," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 1, pp. 69–72, 2005.
- [3] K. C. Lee, J. Haerri, U. Lee, and M. Gerla, "Enhanced perimeter routing for geographic forwarding protocols in urban vehicular scenarios," *Globecom Workshops, 2007 IEEE*, pp. 1–10, 26–30 Nov. 2007.
- [4] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 133–144, 2005.
- [5] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Least-Cost Opportunistic Routing," Tech. Rep., 2007.
- [6] R. C. Shah, S. Wietholter, A. Wolisz, and J. M. Rabaey, "When does opportunistic routing make sense?" in *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 350–356.
- [7] H. Füssler, H. Hartenstein, J. Widmer, M. Mauve, and W. Effelsberg, "Contention-based forwarding for mobile ad-hoc networks."
- [8] M. Zorzi and R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *Mobile Computing, IEEE Transactions on*, vol. 2, no. 4, pp. 337–348, Oct.-Dec. 2003.
- [9] K. Zeng, W. Lou, J. Yang, and D. Brown, "On geographic collaborative forwarding in wireless ad hoc and sensor networks," *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on*, pp. 11–18, Aug. 2007.
- [10] H. Füssler, H. Hartenstein, J. Widmer, M. Mauve, and W. Effelsberg, "Contention-based forwarding for street scenarios."
- [11] J. Härrä, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: generating realistic mobility patterns for VANETs," in *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*. New York, NY, USA: ACM, 2006, pp. 96–97.
- [12] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
- [13] X. Ma, M.-T. Sun, X. Liu, and G. Zhao, "Improving geographical routing for wireless networks with an efficient path pruning algorithm," *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, pp. 246–255, Sept. 2006.
- [14] R. C. Shah, S. Wietholter, A. Wolisz, and J. M. Rabaey, "Modeling and analysis of opportunistic routing in low traffic scenarios," in *WIOPT '05: Proceedings of the Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 294–304.
- [15] K. Zeng, W. Lou, J. Yang, and I. Donald R. Brown, "On throughput efficiency of geographic opportunistic routing in multihop wireless networks," *Mob. Netw. Appl.*, vol. 12, no. 5, pp. 347–357, 2007.