GPSR: Greedy Perimeter Stateless Routing for Wireless Networks*

Brad Karp
Harvard University / ACIRI
karp@eecs.harvard.edu

H. T. Kung
Harvard University
htk@eecs.harvard.edu

ABSTRACT
We present Greedy Perimeter Stateless Routing (GPSR), a novel routing protocol for wireless datagram networks that uses the *positions* of routers and a packet's destination to make packet forwarding decisions. GPSR makes *greedy* forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the *perimeter* of the region. By keeping state only about the local topology, GPSR scales better in per-router state than shortest-path and ad-hoc routing protocols as the number of network destinations increases. Under mobility's frequent topology changes, GPSR can use local topology information to find correct new routes quickly. We describe the GPSR protocol, and use extensive simulation of mobile wireless networks to compare its performance with that of Dynamic Source Routing. Our simulations demonstrate GPSR's scalability on densely deployed wireless networks.

1. INTRODUCTION
In networks comprised entirely of wireless stations, communication between source and destination nodes may require traversal of multiple hops, as radio ranges are finite. A community of ad-hoc network researchers has proposed, implemented, and measured a variety of routing algorithms for such networks. The observation that topology changes more rapidly on a mobile, wireless network than on wired networks, where the use of Distance Vector (DV), Link State (LS), and Path Vector routing algorithms is well-established, motivates this body of work.

DV and LS algorithms require continual distribution of a current map of the entire network's topology to all routers. DV's Bellman-Ford approach constructs this global picture transitively; each router includes its distance from all network destinations in each of its periodic beacons. LS's Dijkstra approach directly floods announcements of the change in any link's status to every router in the network. Small inaccuracies in the state at a router under both DV and LS can cause routing loops or disconnection [29]. When the topology is in constant flux, as under mobility, LS generates torrents of link status change messages, and DV either suffers from out-of-date state [4], or generates torrents of triggered updates.

*This research was supported in part by AFOSR MURI Grant F49620-97-1-0382, and NSF Grant CDA-94-0124, and in part by Microsoft Research, Nortel, Sprint, ISI, and ACIRI.

To appear in Proc. of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000).

The two dominant factors in the scaling of a routing algorithm are:

- The rate of change of the topology.
- The number of routers in the routing domain.

Both factors affect the message complexity of DV and LS routing algorithms: intuitively, pushing current state globally costs packets proportional to the product of the rate of state change and number of destinations for the updated state.

Hierarchy is the most widely deployed approach to scale routing as the number of network destinations increases. Without hierarchy, Internet routing could not scale to support today's number of Internet leaf networks. An Autonomous System runs an intra-domain routing protocol inside its borders, and appears as a single entity in the backbone inter-domain routing protocol, BGP. This hierarchy is based on well-defined and rarely changing administrative and topological boundaries. It is therefore not easily applicable to freely moving ad-hoc wireless networks, where topology has no well-defined AS boundaries, and routers may have no common administrative authority.

Caching has come to prominence as a strategy for scaling ad-hoc routing protocols. Dynamic Source Routing (DSR) [12], Ad-Hoc On-Demand Distance Vector Routing (AODV) [21], and the Zone Routing Protocol (ZRP) [10] all eschew constantly pushing current topology information network-wide. Instead, routers running these protocols request topological information in an *on-demand* fashion as required by their packet forwarding load, and cache it aggressively. When their cached topological information becomes out-of-date, these routers must obtain more current topological information to continue routing successfully. Caching reduces the routing protocols' message load in two ways: it avoids pushing topological information where the forwarding load does not require it (*e.g.,* at idle routers), and it often reduces the number of hops between the router that has the needed topological information and the router that requires it (*i.e.,* a node closer than a changed link may already have cached the new status of that link).

We propose the aggressive use of *geography* to achieve scalability in our wireless routing protocol, Greedy Perimeter Stateless Routing (GPSR). We aim for scalability under increasing numbers of nodes in the network, and increasing mobility rate. As these factors increase, our measures of scalability are:

- Routing protocol message cost: How many routing protocol packets does a routing algorithm send?
- Application packet delivery success rate: What fraction of applications' packets are delivered successfully by a routing algorithm?
- Per-node state: How much storage does a routing algorithm require at each node?

Networks that push on mobility, number of nodes, or both include:

- Ad-hoc networks: Perhaps the most investigated category, these mobile networks have no fixed infrastructure, and support applications for military users, post-disaster rescuers, and temporary collaborations among temporary associates, as at a business conference or lecture [10], [12], [20], [21], [22].
- Sensor networks: Comprised of small sensors, these mobile networks can be deployed with very large numbers of nodes, and have very impoverished per-node resources [6], [13]. Minimization of state per node in a network of tens of thousands of memory-poor sensors is crucial.
- "Rooftop" networks: Proposed by Shepard [24], these wireless networks are not mobile, but are deployed very densely in metropolitan areas (the name refers to an antenna on each building's roof, for line-of-sight with neighbors) as an alternative to wired networking offered by traditional telecommunications providers. Such a network also provides an alternate infrastructure in the event of failure of the conventional one, as after a disaster. A routing system that self-configures (without a trusted authority to configure a routing hierarchy) for hundreds of thousands of such nodes in a metropolitan area represents a significant scaling challenge.

Traditional shortest-path (DV and LS) algorithms require state proportional to the number of reachable destinations at each router. On-demand ad-hoc routing algorithms require state at least proportional to the number of destinations a node forwards packets toward, and often more, as in the case in DSR, in which a node aggressively caches all source routes it overhears to reduce the propagation scope of other nodes' flooded route requests.

We will show that geographic routing allows routers to be nearly stateless, and requires propagation of topology information for only a *single hop*: each node need only know its neighbors' positions. The self-describing nature of position is the key to geography's usefulness in routing. The position of a packet's destination and positions of the candidate next hops are sufficient to make correct forwarding decisions, without any other topological information.

We assume in this work that all wireless routers know their own positions, either from a GPS device, if outdoors, or through other means. Practical solutions include surveying, for stationary wireless routers; inertial sensors, on vehicles; and acoustic range-finding using ultrasonic "chirps" indoors [28]. We further assume bidirectional radio reachability. The widely used IEEE 802.11 wireless network MAC [11] sends link-level acknowledgements for all unicast packets, so that all links in an 802.11 network must be bidirectional. We simulate a network that uses 802.11 radios to evaluate our routing protocol. We consider topologies where the wireless nodes are roughly in a plane. Finally, we assume that packet sources can determine the locations of packet destinations, to mark packets they originate with their destination's location. Thus, we assume a location registration and lookup service that maps node addresses to locations [18]. Queries to this system use the *same* geographic routing system as data packets; the querier geographically addresses his request to a location server. The scope of this paper is limited to geographic routing. We argue for the eminent practicality of the location service briefly in Section 3.7. We adopt IP terminology throughout this paper, though GPSR can be applied to any datagram network.

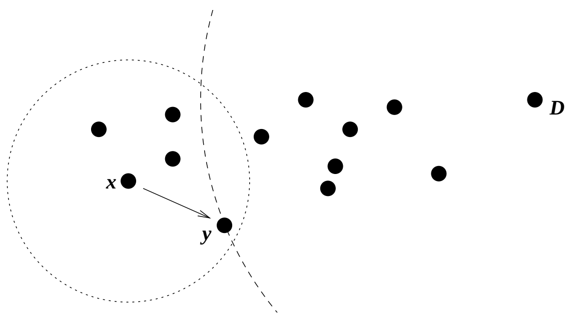

Figure 1: Greedy forwarding example. y **is** x**'s closest neighbor to** D**.**

In the following sections, we describe the algorithms that comprise GPSR, measure and analyze GPSR's performance and behavior in simulated mobile networks, cite and differentiate related work, identify future research opportunities suggested by GPSR, and conclude by summarizing our findings.

2. ALGORITHMS AND EXAMPLES

We now describe the Greedy Perimeter Stateless Routing algorithm. The algorithm consists of two methods for forwarding packets: *greedy forwarding*, which is used wherever possible, and *perimeter forwarding*, which is used in the regions greedy forwarding cannot be.

2.1 Greedy Forwarding

As alluded to in the introduction, under GPSR, packets are marked by their originator with their destinations' locations. As a result, a forwarding node can make a locally optimal, greedy choice in choosing a packet's next hop. Specifically, if a node knows its radio neighbors' positions, the locally optimal choice of next hop is the neighbor geographically closest to the packet's destination. Forwarding in this regime follows successively closer geographic hops, until the destination is reached. An example of greedy next-hop choice appears in Figure 1. Here, x receives a packet destined for D. x's radio range is denoted by the dotted circle about x, and the arc with radius equal to the distance between y and D is shown as the dashed arc about D. x forwards the packet to y, as the distance between y and D is less than that between D and any of x's other neighbors. This greedy forwarding process repeats, until the packet reaches D.

A simple beaconing algorithm provides all nodes with their neighbors' positions: periodically, each node transmits a beacon to the broadcast MAC address, containing only its own identifier (*e.g.,* IP address) and position. We encode position as two four-byte floating-point quantities, for x and y coordinate values. To avoid synchronization of neighbors' beacons, as observed by Floyd and Jacobson [8], we jitter each beacon's transmission by 50% of the interval B between beacons, such that the mean inter-beacon transmission interval is B, uniformly distributed in $[0.5B, 1.5B]$.

Upon not receiving a beacon from a neighbor for longer than timeout interval T, a GPSR router assumes that the neighbor has failed or gone out-of-range, and deletes the neighbor from its table. The 802.11 MAC layer also gives direct indications of link-level retransmission failures to neighbors; we interpret these indications identically. We have used $T = 4.5B$, three times the maximum jittered beacon interval, in this work.

Greedy forwarding's great advantage is its reliance only on knowledge of the forwarding node's immediate neighbors. The state required is negligible, and dependent on the density of nodes in the wireless network, not the total number of destinations in the network.[1] On networks where multi-hop routing is useful, the number of neighbors within a node's radio range must be substantially less than the total number of nodes in the network.

The position a node associates with a neighbor becomes less current between beacons as that neighbor moves. The accuracy of the set of neighbors also decreases; old neighbors may leave and new neighbors may enter radio range. For these reasons, the correct choice of beaconing interval to keep nodes' neighbor tables current depends on the rate of mobility in the network and range of nodes' radios. We show the effect of this interval on GPSR's performance in our simulation results. We note that keeping current topological state for a one-hop radius about a router is the minimum required to do *any* routing; no useful forwarding decision can be made without knowledge of the topology one or more hops away.

This beaconing mechanism does represent pro-active routing protocol traffic, avoided by DSR and AODV. To minimize the cost of beaconing, GPSR piggybacks the local sending node's position on *all* data packets it forwards, and runs all nodes' network interfaces in promiscuous mode, so that each station receives a copy of all packets for all stations within radio range. At a small cost in bytes (twelve bytes per packet), this scheme allows all packets to serve as beacons. When any node sends a data packet, it can then reset its inter-beacon timer. This optimization reduces beacon traffic in regions of the network actively forwarding data packets.

In fact, we could make GPSR's beacon mechanism fully reactive by having nodes solicit beacons with a broadcast "neighbor request" only when they have data traffic to forward. We have not felt it necessary to take this step, however, as the one-hop beacon overhead does not congest our simulated networks.

The power of greedy forwarding to route using only neighbor nodes' positions comes with one attendant drawback: there are topologies in which the only route to a destination requires a packet move temporarily *farther* in geometric distance from the destination [7], [16]. A simple example of such a topology is shown in Figure 2. Here, x is closer to D than its neighbors w and y. Again, the dashed arc

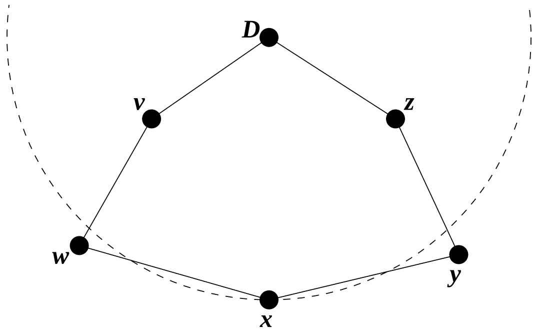

Figure 2: Greedy forwarding failure. x is a local maximum in its geographic proximity to D; w and y are farther from D.

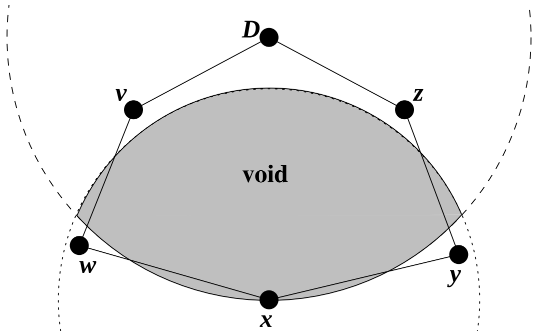

Figure 3: Node x's *void* with respect to destination D.

about D has a radius equal to the distance between x and D. Although two paths, $(x \to y \to z \to D)$ and $(x \to w \to v \to D)$, exist to D, x will not choose to forward to w or y using greedy forwarding. x is a local maximum in its proximity to D. Some other mechanism must be used to forward packets in these situations.

2.2 The Right-Hand Rule: Perimeters

Motivated by Figure 2, we note that the *intersection* of x's circular radio range and the circle about D of radius $|\overline{xD}|$ (that is, of the length of line segment $\overline{xD}$) is empty of neighbors. We show this region clearly in Figure 3. From node x's perspective, we term the shaded region without nodes a *void*. x seeks to forward a packet to destination D beyond the edge of this void. Intuitively, x seeks to route *around* the void; if a path to D exists from x, it doesn't include nodes located within the void (or x would have forwarded to them greedily).

The long-known *right-hand rule* for traversing a graph is depicted in Figure 4. This rule states that when arriving at node x from node y, the next edge traversed is the next one sequentially counterclockwise about x from edge (x, y). It is known that the right-hand rule traverses the interior of a closed polygonal region (a *face*) in clockwise edge order—in this case, the triangle bounded by the edges between nodes x, y, and z, in the order $(y \to x \to z \to y)$. The rule traverses an exterior region, in this case, the region *outside* the same triangle, in counterclockwise edge order.

We seek to exploit these cycle-traversing properties to route around voids. In Figure 3, traversing the cycle $(x \to w \to v \to D \to z \to y \to x)$ by the right-hand rule amounts to navigating *around the pictured void*, specifically, to nodes closer to the destination than x (in this case, including the destination itself, D). We call the sequence of

[1] The word "stateless" in GPSR's name is not meant literally, but refers to this small, purely local state.

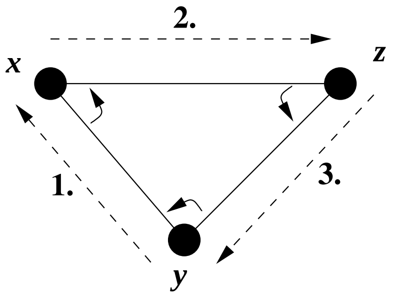

Figure 4: The right-hand rule (*interior* of the triangle). x receives a packet from y, and forwards it to its first neighbor counterclockwise about itself, z, *&c.*

edges traversed by the right-hand rule a *perimeter*.

In earlier work [15], [16], we propose mapping perimeters by sending packets on tours of them, using the right-hand rule. The state accumulated in these packets is cached by nodes, which recover from local maxima in greedy forwarding by routing to a node on a cached perimeter closer to the destination. This approach requires a heuristic, the *no-crossing heuristic*, to force the right-hand rule to find perimeters that enclose voids in regions where edges of the graph cross. This heuristic improves reachability results overall, but still leaves a serious liability: the algorithm does not always find routes when they exist. The no-crossing heuristic blindly removes whichever edge it encounters *second* in a pair of crossing edges. The edge it removes, however, may partition the network. If it does, the algorithm will not find routes that cross this partition.

2.3 Planarized Graphs

While the no-crossing heuristic empirically finds the vast majority of routes (over 99.5% of the $n(n-1)$ routes among n nodes [16]) in randomly generated networks, it is unacceptable for a routing algorithm persistently to fail to find a route to a reachable node in a static, unchanging network topology. Motivated by the insufficiency of the no-crossing heuristic, we present alternative methods for eliminating crossing links from the network.

A graph in which no two edges cross is known as *planar*. A set of nodes with radios, where all radios have identical, circular radio range r, can be seen as a graph: each node is a vertex, and edge (n,m) exists between nodes n and m if the distance between n and m, $d(n,m) \leq r$. Graphs whose edges are dictated by a threshold distance between vertices are termed *unit graphs*. In the sense that network radio hardware is traditionally viewed as having a nominal open-space range (*e.g.*, 250 meters for 900 MHz DSSS WaveLAN), this model is reasonable. We additionally assume that the nodes in the network have negligible difference in altitude, so that they can be considered roughly in a plane. We discuss these assumptions further in Section 5.

The *Relative Neighborhood Graph (RNG)* and *Gabriel Graph (GG)* are two planar graphs long-known in varied disciplines [9], [27]. An algorithm for removing edges from the graph that are not part of the RNG or GG would yield a network with no crossing links. For our application, the algorithm should be run in a distributed fashion by each node in the network, where a node needs information only about the local topology as the algorithm's input. However, for this strategy to be successful, one important property must be shown:

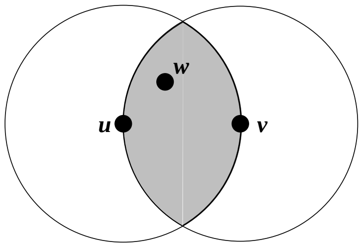

Figure 5: The RNG graph. For edge (u,v) to be included, the shaded lune must contain no witness w.

> Removing edges from the graph to reduce it to the RNG or GG must not disconnect the graph; this would amount to partitioning the network.

Given a collection of vertices with known positions, the RNG is defined as follows:

> An edge (u,v) exists between vertices u and v if the distance between them, $d(u,v)$, is less than or equal to the distance between every *other* vertex w, and whichever of u and v is farther from w. In equational form:
>
> $$\forall w \neq u,v : d(u,v) \leq \max[d(u,w), d(v,w)]$$

Figure 5 depicts the rule for constructing the RNG. The shaded region, the *lune* between u and v, must be empty of any witness node w for (u,v) to be included in the RNG. The boundary of the lune is the intersection of the circles about u and v of radius $d(u,v)$.

When we begin with a connected unit graph and remove edges not part of the RNG, note that we cannot disconnect the graph. (u,v) is only eliminated from the graph when there exists a w within range of both u and v. Thus, eliminating an edge requires an alternate path through a witness exist. Each connected component in an unobstructed radio network will not be disconnected by removing edges not in the RNG.

Under the previously described beaconing mechanism, through which all nodes know their immediate neighbors, if u and v can reach one another, they must both know all nodes with the lune. Starting from a full list of its neighbors, N, each node u can remove non-RNG links as follows:

for all $v \in N$ **do**
  for all $w \in N$ **do**
    if $w == v$ **then**
      continue
    else if $d(u,v) > \max[d(u,w), d(v,w)]$ **then**
      eliminate edge (u,v)
      break
    end if
  end for
end for

The GG is defined as follows:

> An edge (u,v) exists between vertices u and v if no

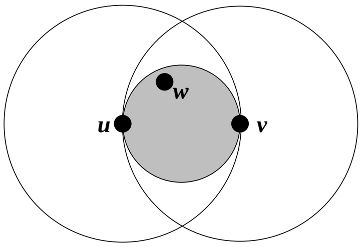

Figure 6: The GG graph. For edge (u,v) to be included, the shaded circle must contain no witness w.

other vertex w is present within the circle whose diameter is $\overline{uv}$. In equational form:

$$\forall w \neq u,v : d^2(u,v) < [d^2(u,w) + d^2(v,w)]$$

Figure 6 depicts the GG graph membership criterion.

As the midpoint of $\overline{uv}$ is the center of the circle with diameter $\overline{uv}$, a node u can remove its non-GG links from a full neighbor list N thus:

```
m = midpoint of uv
for all v ∈ N do
  for all w ∈ N do
    if w == v then
      continue
    else if d(m,w) < d(u,m) then
      eliminate edge (u,v)
      break
    end if
  end for
end for
```

Eliminating edges in the GG cannot disconnect a connected unit graph, for the same reason as was the case for the RNG. Both these algorithms for rendering the graph of the radio network planar take time $O(\deg^2)$ at each node, where deg is the node's degree in the full radio graph.

It has been shown in the literature [27] that the RNG is a subset of the GG. This is consistent with the smaller shaded region searched for a witness in the GG, as compared with in the RNG. Figure 7 shows a full unit graph corresponding to 200 nodes randomly placed on a 2000-by-2000-meter region, with radio ranges of 250 meters; the GG subset of the full graph; and the RNG subset of the full graph. Note that the RNG and GG offer different densities of connectivity by eliminating different numbers of links. Many MAC layers exhibit drastically reduced efficiency as the number of mutually reachable sending stations increases [1], [5]. Moreover, while any packet a node transmits monopolizes the shared channel within its radio range, MAC protocols that address the hidden terminal problem, including 802.11 [11], MACA [14], and MACAW [2], deliberately spread contention to the full radio ranges of *both* sender and receiver. Under such regimes, using fewer links in routing can improve spatial diversity.

2.4 Combining Greedy and Planar Perimeters

We now present the full Greedy Perimeter Stateless Routing algorithm, which combines greedy forwarding (Section 2.1) on the full network graph with perimeter forwarding on the planarized network graph where greedy forwarding is not possible. Recall that all nodes maintain a neighbor table, which stores the addresses and locations of their single-hop radio neighbors. This table provides all state required for GPSR's forwarding decisions, beyond the state in the packets themselves.

Field	Function
D	Destination Location
L_p	Location Packet Entered Perimeter Mode
L_f	Point on $\overline{xV}$ Packet Entered Current Face
e_0	First Edge Traversed on Current Face
M	Packet Mode: Greedy or Perimeter

Table 1: GPSR packet header fields used in perimeter mode forwarding.

The packet header fields GPSR uses in perimeter-mode forwarding are shown in Table 1. GPSR packet headers include a flag field indicating whether the packet is in greedy mode or perimeter mode. All data packets are marked initially at their originators as greedy-mode. Packet sources also include the geographic location of the destination in packets. Only a packet's source sets the location destination field; it is left unchanged as the packet is forwarded through the network.

Upon receiving a greedy-mode packet for forwarding, a node searches its neighbor table for the neighbor geographically closest to the packet's destination. If this neighbor is closer to the destination, the node forwards the packet to that neighbor. When no neighbor is closer, the node marks the packet into perimeter mode.

GPSR forwards perimeter-mode packets using a simple planar graph traversal. In essence, when a packet enters perimeter mode at node x bound for node D, GPSR forwards it on progressively closer *faces* of the planar graph, each of which is crossed by the line $\overline{xD}$. A planar graph has two types of faces. *Interior faces* are the closed polygonal regions bounded by the graph's edges. The *exterior face* is the one unbounded face outside the outer boundary of the graph. On each face, the traversal uses the right-hand rule to reach an edge that crosses line $\overline{xD}$. At that edge, the traversal moves to the adjacent face crossed by $\overline{xD}$. See Figure 8 for an example. Note that in the figure, each face traversed is pierced by $\overline{xD}$—the first two and last faces are interior faces, while the third is the exterior face.[2]

When a packet enters perimeter mode, GPSR records in the packet the location L_p, the site where greedy forwarding failed. This location is used at subsequent hops to determine whether the packet can be returned to greedy mode. Each time GPSR forwards a packet onto a new face, it records in L_f the point on $\overline{xD}$ shared between the previous and new faces. Note that L_f need not be located at a node; $\overline{xD}$ usually intersects edges, as in Figure 8. Finally, GPSR records e_0, the first edge (sender and receiver addresses) a packet crosses on a new face, in the packet.

Upon receiving a perimeter-mode packet for forwarding, GPSR first compares the location L_p in a perimeter-mode packet with the forwarding node's location. GPSR returns a packet to greedy

[2]Forwarding in Figure 8 is done in perimeter mode only for exposition; true GPSR forwards greedily when neighbors closer to the destination are available.

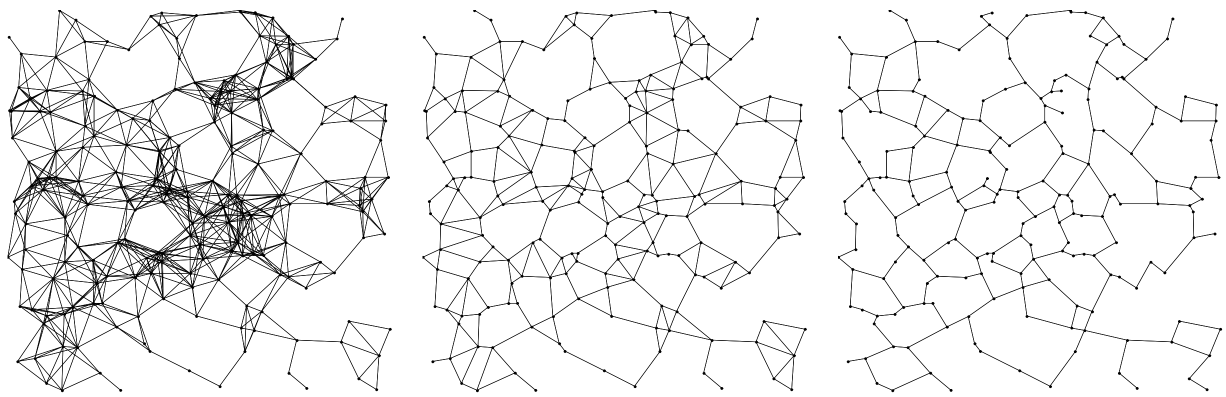

Figure 7: Left: the full graph of a radio network. 200 nodes, uniformly randomly placed on a 2000 x 2000 meter region, with a radio range of 250 m. Center: the GG subset of the full graph. Right: the RNG subset of the full and GG graphs.

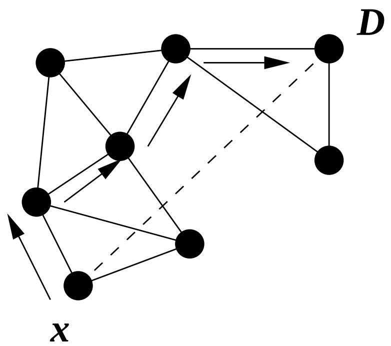

Figure 8: Perimeter Forwarding Example. D is the destination; x is the node where the packet enters perimeter mode; forwarding hops are solid arrows; the line $\overline{xD}$ is dashed.

mode if the distance from the forwarding node to D is less than that from L_p to D.[3] Perimeter forwarding is only intended to recover from a local maximum; once the packet reaches a location closer than where greedy forwarding previously failed for that packet, the packet can continue greedy progress toward the destination without danger of returning to the prior local maximum.

When a packet enters perimeter mode at x, GPSR forwards it along the face intersected by the line $\overline{xD}$. x forwards the packet to the first edge counterclockwise about x from the line $\overline{xD}$. This determines the first face over which to forward the packet. Thereafter, GPSR forwards the packet around that face using the right-hand rule. There are two cases to consider: either x and D are connected by the graph, or they are not.

When x and D are connected by the graph, traversing the face bordering x in either direction (we use the previously described right-hand rule) must lead to a point y at which $\overline{xD}$ intersects the far side of the face. This is the case whether the traversed face is interior or exterior. At y, GPSR has clearly reduced the distance between the packet and its destination, in comparison with the packet's start in perimeter mode at x.

While forwarding around a face, GPSR determines whether the edge to the chosen next hop n intersects $\overline{xD}$. GPSR has the information required to make this determination, as L_p and D are recorded in the packet, and a GPSR node stores its own position and those of its neighbors. If a node borders the edge where this intersection point y lies, GPSR sets the packet's L_f to y. At this point, the packet is forwarded along the *next* face bordering point y that is intersected by $\overline{xD}$. The node forwards the packet along the first edge of this next face—by the right-hand rule, the next edge counterclockwise about itself from n. This first edge on the new face is recorded in the packet's e_0 field.

This process repeats at successively closer faces to D. At each face, the packet progresses by the right-hand rule until reaching the edge that interesects with $\overline{xD}$ at a point y closer than the packet's L_f field to D. Finally, the face containing D is reached, and the right-hand-rule leads to D along that face.

When D is not reachable (*i.e.,* it is disconnected from the graph), two cases exist: the disconnected node lies either inside an interior face, or outside the exterior face. GPSR will forward a perimeter-mode packet until the packet reaches the corresponding face. Upon reaching this interior or exterior face, the packet will tour unsuccessfully around the entirety of the face, without finding an edge intersecting $\overline{xD}$ at a point closer to D than L_f. When the packet traverses the first edge it took on this face for the second time, GPSR notices the repetition of forwarding on the edge e_0 stored in the packet, and correctly drops the packet, as the destination is unreachable; the perimeter-mode graph traversal to a reachable destination never sends a packet across the same link in the same direction twice.

Note that GPSR will greedily forward a packet for potentially many hops, before the packet loops on an exterior or interior face and is recognized as undeliverable. If the majority of unreachable destinations lie beyond the boundary of a single face, undeliverable packets may concentrate at that face of the network graph. This behavior is a direct consequence of GPSR's avoidance of transitive routing protocol traffic across the many hops from a destination to a forwarding router. Other techniques for scaling routing have similar effects, however: the hierarchy used to scale routing on wired networks obscures intra-domain link failures from the backbone in the interest of scaling. Thus, the inter-domain routing system will push a packet a great distance, with the potential result that the packet will be dropped inside the destination AS.

[3] GPSR could also return the packet to greedy mode if any *neighbor* were closer to D than L_p. We have not implemented this variant.

By the end-to-end argument [23], the most logical place for routing unreachability to be determined, and the load on the network from undeliverable packets to be reduced, is at the sending end-system. Mechanisms from inside the network, like ICMP Unreachable, are hard to interpret at senders; it is hard to know on what timescale they indicate unreachability, for example. Applications running over a GPSR-routed network, or any other network, should offer a conforming load; senders should cut their transmission rate absent feedback from receivers.

2.5 Protocol Implementation

To make GPSR robust on a mobile IEEE 802.11 network, we made the following significant choices in our implementation:

- **Support for MAC-layer failure feedback**: As used in DSR [4], we receive notification from the 802.11 MAC layer when a packet exceeds its maximum number of retransmit retries. Barring congestive collapse, a retransmit retry exceeded failure indicates that the intended recipient has left radio range. Use of this feedback may inform GPSR earlier than otherwise possible through expiration of the neighbor timeout interval (4.5*B*).
- **Interface queue traversal:** Related to MAC-layer feedback, this implementation detail had a profound effect on our results. While an IEEE 802.11 interface repeatedly retransmits the packet at the head of its queue, it head-of-line blocks, waiting for a link-level acknowledgement from the receiver. This head-of-line blocking reduces the available transmit duty cycle of the interface significantly. For this reason, upon notification of a MAC retransmit retry failure, we traverse the queue of packets for the interface, and remove all packets addressed to the failed transmission's recipient. We pass these packets back to the routing protocol for re-forwarding to a different next hop. This change virtually eliminated what we'd previously thought to be MAC contention in high-mobility simulations where neighbors were lost frequently; the timeouts and head-of-line blocking were what really had been causing the drops at the interface queue. The implementation of DSR for ns-2 [25] implements this useful optimization, though we don't see it mentioned in the published work on DSR.
- **Promiscuous use of the network interface:** Also as used in DSR [4], GPSR disables MAC address filtering to receive copies of all packets for all stations within its radio range. As described in Section 2.1, all packets carry their local sender's position, to reduce the rate at which beacon packets must be sent, and to keep positions in neighbor lists maximally current in regions under traffic load.
- **Planarization of the graph:** Both the RNG and GG planarizations depend on having current position information for a node's current set of neighbors. We have implemented both planarizations, though the results we present in this paper use only the RNG. As nodes move, a planarization becomes stale, and less useful for accurate perimeter-mode packet forwarding. In our current implementation, we re-planarize the graph upon every acquisition of a new neighbor, and every loss of a former neighbor, as distinguishable by receipt of a beacon or data packet (promiscuously) from a previously unknown neighbor, and by a beacon timeout for a neighbor, or MAC transmit failure indication. However, this choice will not keep the planarization current if nodes only move *within* a node's radio range, but no nodes move into or out of it. In future, we will incrementally update the planarization upon receipt of every beacon (or promiscuous data packet) from a neighbor, to keep the planarized graph maximally up-to-date.

3. SIMULATION RESULTS AND EVALUATION

To measure our success in meeting the design goals for GPSR, we simulated the algorithm on a variety of static and mobile network topologies. We focus mainly on the mobile simulation results in this paper, as that part of the design space is more demanding of a routing protocol—link additions and removals are far more frequent under mobility. To compare the performance of GPSR with prior work in wireless routing, we also simulate Johnson *et al.*'s Dynamic Source Routing, DSR [12], [19], which has been shown to offer higher packet delivery ratios and lower routing protocol overhead than several other ad-hoc routing protocols [4].

3.1 Simulation Environment

We simulated GPSR in ns-2 [26], using the wireless extensions developed at Carnegie Mellon [25]. This simulation environment offers high fidelity, as it includes full simulation of the IEEE 802.11 physical and MAC layers. Moreover, by using the same simulation code base as the measurement study used to evaluate DSR [4], we ensure our results are directly comparable to those published previously.

The ns-2 wireless simulation model simulates nodes moving in an unobstructed plane. Motion follows the *random waypoint* model [4]: a node chooses a destination uniformly at random in the simulated region, chooses a velocity uniformly at random from a configurable range, and then moves to that destination at the chosen velocity. Upon arriving at the chosen waypoint, the node pauses for a configurable period before repeating the same process. In this model, the pause time acts as a proxy for the degree of mobility in a simulation; longer pause time amounts to more nodes being stationary for more of the simulation.

In the simulations where we compare GPSR with DSR, we use simulation parameters identical to a subset of those used by Broch *et al.* [4]. Our simulations are for networks of 50, 112, and 200 nodes with 802.11 WaveLAN radios, with a nominal 250-meter range. The nodes are initially placed uniformly at random in a rectangular region. All nodes move according to the random waypoint model, with a maximum velocity of 20 m/s. We simulate pause times of 0, 30, 60, and 120 seconds, the highest mobility cases, as they are the most demanding of a routing algorithm. Broch *at al.* also simulated 300-, 600-, and 900-second pause times, perhaps in large part because two of the routing algorithms they evaluated (DSDV and TORA) performed well in these cases. We simulate 30 CBR traffic flows, originated by 22 sending nodes. Each CBR flow sends at 2 Kbps, and uses 64-byte packets. Broch *et al.* simulated a wider range of flow counts (10, 20, and 30 flows); we simulate only the 30-flow case as this case makes the greatest demands on the routing protocols: the most data traffic to forward and most destinations to which to route. Each simulation lasts for 900 seconds of simulated time. We simulate at each pause time with six different randomly generated motion patterns, and present the mean of each metric over these six runs. Because we only simulate the high mobility cases, and motion patterns during each run are random, there

Nodes	Region	Density	CBR Flows
50	1500 m $\times$ 300 m	1 node / 9000 m^2	30
112	2250 m $\times$ 450 m	1 node / 9000 m^2	30
200	3000 m $\times$ 600 m	1 node / 9000 m^2	30

Table 2: Simulated Topology Characteristics

was little variance in the results among these runs. Runs with more static topologies would be much more sensitive to node placement. Table 2 summarizes the three network sizes we simulate.

These Broch *et al.* simulated networks are quite dense; the y dimension of the space in which nodes are distributed in their 50-node simulations is only 50 meters larger than the simulated radio range. On average, there is one node per 9,000 square meters in these simulations. A radio range is nearly 200,000 square meters. As a result, there are an average of approximately 20 neighbors within range of the average node in these networks. DSR's caching of overheard routes gives great benefit in such dense topologies. And GPSR can use greedy mode to forward the vast majority of packets.

Our simulations do not include a distributed location database for annotating packets with destinations' positions. Our results here argue that the GPSR approach to routing warrants investigation into efficient location databases, and related work is already underway in this area [18]. In these simulation results, we use an idealized location database: each source annotates packets it originates with the true location of the destination. In this sense, our results represent the lowest control packet load that can be expected from GPSR. Section 3.7 discusses GPSR's interaction with a location database further.

Before gathering the measurement results we present here, we validated the GPSR implementation extensively by running it on hundreds of *non-mobile* topologies, over an ideal MAC layer (the Null MAC [25]), a 2 Mbps, contention-free network. Our goal in these tests is to achieve 100% delivery success to demonstrate that the GPSR code makes correct forwarding decisions. After reaching this 100% goal on the Null MAC, we validated the GPSR implementation on these non-mobile topologies atop the ns 802.11 MAC layer, to verify GPSR's response to MAC transmit failure callbacks.

We evaluate GPSR and DSR using three metrics: packet delivery success rate, routing protocol overhead, and optimality of path lengths taken by data packets.

3.2 Packet Delivery Success Rate

Figure 9 shows how many application packets GPSR delivers successfully for varying values of B, the beaconing interval, as a function of pause time. The same figure for DSR is included for comparison. Note the narrow range of values on the y axis; all algorithms on this graph deliver over 97% of user packets. Only packets for which a path *exists* to the destination are included in the graph; delivery failure to a truly disconnected destination does not represent failure of a routing algorithm. However, as mentioned above, disconnection of a node is extremely rare in these simulations, as connectivity is dense. As one would expect, the decrease in precision of neighbor lists caused by the longer beaconing interval of 3 seconds results in a slightly reduced delivery success rate. But it appears that there is little added benefit, for the simulated mo-

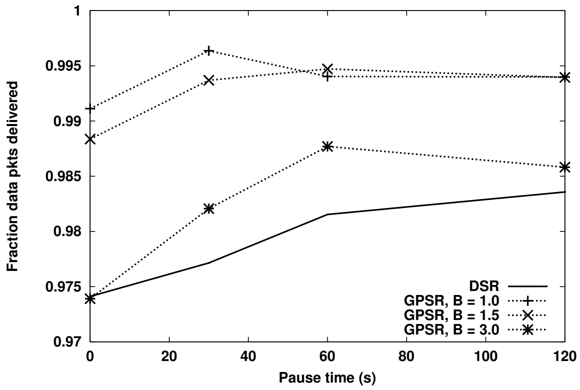

Figure 9: Packet Delivery Success Rate. GPSR with varying beacon intervals, *B*, compared with DSR. 50 nodes.

bility rates and radio ranges, in decreasing B beyond 1.5. At all pause times simulated, GPSR delivers a slightly greater fraction of packets successfully than DSR.

3.3 Routing Protocol Overhead

Figure 10 shows the routing protocol overhead, measured in total number of routing protocol packets sent network-wide during the entire simulation, for GPSR with varying B and for DSR. Because GPSR's beacons are sent pro-actively (modulo data traffic with piggybacked position information), each beaconing interval results in a constant level of routing protocol traffic, independent of pause time (and though we didn't simulate it, number of traffic flows, until application traffic becomes heavy enough to allow nodes never to send beacon packets). Because DSR is a reactive routing protocol, it generates increased routing protocol traffic as mobility increases.

We note with puzzlement that while we believe we run the exact same DSR simulator code as Broch *et al.*, we observe somewhat greater traffic load from DSR than they did in the 30-flow DSR simulations in [4]. To compare with these prior published results, we include a *second* DSR curve, DSR-Broch, in Figure 10. Again, our results, both for GPSR and DSR, represent means of 6 simulation runs. We see little variance in the individual run results; at these four shortest pause times, there is less simulation sensitivity to the particular random node placement than there is in longer-pause-time simulations. In any event, the contour of their reported curve is the same as that of our DSR curve, and GPSR with $B = 1.5$ offers between a threefold and fourfold overhead reduction under DSR. The contour of the DSR and GPSR curves suggests that as mobility increases further, GPSR may offer greater savings in routing protocol overhead.

3.4 Path Length

Figure 11 gives a histogram of the number of hops *beyond* the ideal true shortest path length in which GPSR and DSR deliver all successfully delivered packets. The data are presented as percentages of all packets delivered across all six 50-node simulations of GPSR ($B = 1.5$) and DSR at pause time zero, where topological information available to both algorithms is least current. Here, the "0" bin counts packets delivered in the optimal, true-shortest-path number of hops, and successive bins count packets that took one hop longer, two hops longer, *&c*.

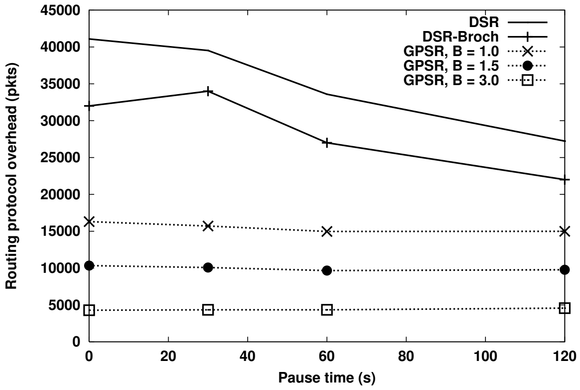

Figure 10: Routing Protocol Overhead. Total routing protocol packets sent network-wide during the simulation for GPSR with varying beacon intervals, *B*, compared with DSR. 50 nodes.

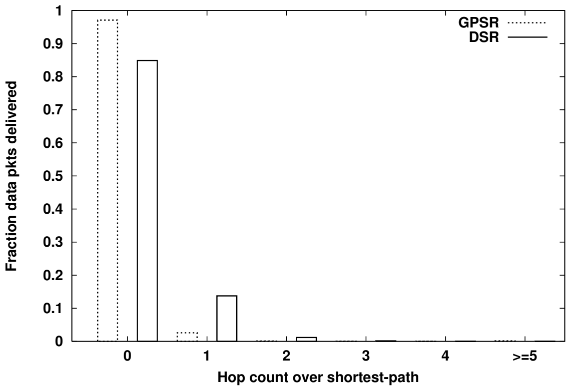

Figure 11: Path length beyond optimal for GPSR's and DSR's successfully delivered packets. 50 nodes.

GPSR delivers the vast majority of packets in the optimal number of hops. Intuitively, on a dense radio network, greedy forwarding approximates shortest-path routing. GPSR delivers 97% of its packets along optimal-length paths, *vs.* 84.9% for DSR. This difference is attributable to DSR's caching, which reduces the propagation of route requests, but causes sub-optimal cached paths to be used for forwarding until the cached route breaks.

3.5 Effect of Network Diameter

Figures 12 and 13 present packet delivery ratio and overhead results for larger-scale, 112- and 200-node networks with identical traffic sources and node density. The 200-node results include only one data point each (still the average of six runs with different randomly generated motion patterns), at pause time 0, because simulating 200-node networks is so computationally expensive. In these simulations, the regions on which nodes move are 2250 by 450 meters and 3000 by 600 meters, respectively, such that the number of square meters per node (9000 m^2/node) remains the same as that in the 50-node simulations. The intent in these simulations is to evaluate the scaling of DSR and GPSR as network diameter increases. When routes are longer, the probability of a route's breaking increases. The traffic sources are the same as in the smaller network simulations: 30 CBR sources of 2 Kbps each, transmitting 64-byte

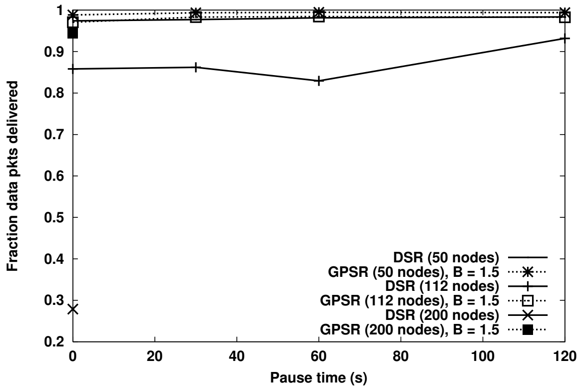

Figure 12: Packet Delivery Success Rate. For GPSR with $B = 1.5$ compared with DSR. 50, 112, and 200 nodes.

packets. We also include the same performance curves for the 50-node network, for comparison.

Note that in Figure 13, the *y* axis is log-scaled. For each number of nodes, GPSR's traffic overhead once again remains flat, as it is a non-reactive protocol. At a constant node density, network diameter has no effect on GPSR's *local* routing protocol message traffic, since GPSR never sends routing packets beyond a single hop. This particular metric, *network-wide* count of routing protocol packets, shows the GPSR beacon traffic to be linear in node count, as compared with the 50-node simulations. DSR's traffic overhead is significantly larger on the wider-diameter, 112- and 200-node networks, as the protocol must propagate source route information along the full length of a route. DSR's caching of routes does not avoid this significant message complexity increase.

GPSR's traffic delivery ratio remains high at all pause times on these larger-scale networks. It is GPSR's use of only local topology information that allows the protocol to maintain this delivery ratio; there is no penalty for GPSR as the path length from source to destination lengthens. Moreover, GPSR recovers from loss of a neighbor by greedily forwarding to another appropriate neighbor; this failover is instantaneous. DSR's delivery ratio decreases considerably in the wider-diameter network, owing to DSR's need to maintain full, end-to-end source routes.

Note that the maximum path lengths between nodes in these wider-diameter simulations are still under 16 nodes. We mention this fact as the DSR simulator code uses a compile-time constant for the maximum length of a route it will discover, and maximum propagation distance for route requests.

In these 112- and 200-node runs, DSR's 64-route cache is full at virtually every node. While the number of destinations in the network is only 30 in our simulations, DSR caches multiple routes per destination, and might profit from being able to cache more routes, though at the expense of increased per-router state (see the next section).

3.6 State per Router

When measuring state per router, the relevant metric is the number of *nodes* in a router's tables—not the number of routes. Because DSR uses source routes, each route stored by a DSR router requires

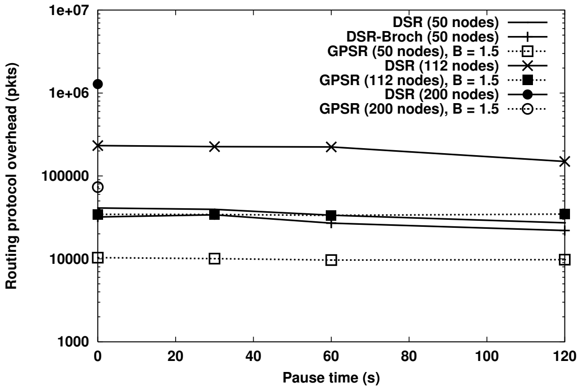

Figure 13: Routing Protocol Overhead. Total routing protocol packets sent network-wide during the simulation for GPSR with $B = 1.5$ compared with DSR. y axis log-scaled. 50, 112, and 200 nodes.

storage for each node along the route.

We measure DSR's average per-node state for the set of 200-node simulations with pause time 0. Because the state maintained by a node in these networks changes constantly, we take a snapshot at time 300.0 seconds in each of our 900-second simulations, and measure the state in use by each node at that instant. A GPSR node stores state for 26 nodes on average in the pause-time-0, 200-node simulations. This figure depends on node density, as the only state a GPSR router keeps is an entry for each of its single-hop radio neighbors.

In comparison, the average DSR node in our 200-node, pause-time-0 simulation stores state for 266 nodes. It should be noted that this value for DSR is clamped by the fixed-size route cache in the DSR simulator's implementation; this cache is limited to 64 routes. While DSR might profit in robustness from a larger route cache, the state cost per node will increase dramatically as the network size increases, and increasingly many more diverse routes are discovered. A DSR larger route cache may also store more broken routes, as mobility and network diameter increase.

Each node stored in a GPSR router's neighbor table arguably requires more storage than a node stored in a DSR router's table, as GPSR routers must track the positions and addresses of their neighbors, while DSR routers need only track the addresses of hops in a source route. GPSR uses 12 bytes for each neighbor in its table; two 4-byte floating point values for position coordinates, and 4 bytes for address. DSR uses 4 bytes per address. However, this is a constant factor difference, dominated by far by the number of nodes stored.

3.7 Location Database Overhead

The addition of location registration and lookup traffic for a location database will increase GPSR's overhead. For bidirectional traffic flows between end nodes, a location database lookup will often need only be performed by the connection initiator at the start of a connection; thereafter, both connection endpoints keep one another apprised of their changing locations by stamping their current locations in each data packet they transmit. In this case, the actual location database lookup is a one-time, DNS-like lookup.

It is important to note that GPSR decouples participation in routing as a forwarder from participation in the location database. Only nodes that are traffic destinations need send location updates to the database, and only nodes that originate traffic need send location queries to it. In a dense sensor network [13], it is easy to imagine configuring only a small subset of sensor nodes to take measurements at only the current points of interest, by flooding a few configuration packets through the network. The remainder of the sensor network can provide a robust transit network for the collection of measurements from sensors to the measurement point, with GPSR's beacons as their only routing protocol traffic—*without* generating any traffic to and from the location database.

In some networks, a destination may inherently have a well-known location. For example, the position of one or more fixed data collection points for a sensor network may be known to all sensors, in which case no location database is needed.

It is also important to note that queries and registrations for the location database are routable using GPSR itself; the queries and registrations are geographically addressed. In the next section, we cite a location database system built on geographic addressing.

4. RELATED WORK

Finn [7] is the earliest we know to propose greedy routing using the locations of nodes. He recognizes the small forwarding state greedy forwarding requires, and observes the failure of greedy forwarding upon reaching a local maximum. He proposes flooding search for a closer node as a strategy for recovering from local maxima.

We first propose greedy forwarding and perimeter traversal in [16], as briefly discussed in Section 2.2. This work simulates this older algorithm on static networks, in a very idealized (contentionless, infinite bandwidth) simulator, and presents the state per node (including perimeter node lists, notably absent from the current work), message cost from cold start to convergence, and frequency with which routes are not found, because of the imperfect no-crossing heuristic. This prior work does not offer any mobile simulation results, and the earlier algorithm suffers in many ways from its maintenance of state beyond neighbor lists at all routers: increased state size for perimeter lists at all nodes, periodic pro-active routing protocol traffic that perimeter probes generate, and staleness of perimeter lists that would occur under mobility. The unreachability of even a small fraction of destinations on *static* networks because of the failure of the no-crossing heuristic is also problematic; such routing failures are permanent, not transitory.

Johnson and Maltz [12] propose the Dynamic Source Routing (DSR) protocol. DSR generates routing traffic reactively: a router floods a route request packet throughout the network. When the request reaches the destination, the destination returns a route reply to the request's originator. Nodes aggressively cache routes that they learn, so that intermediate nodes between a querier and destination may subsequently reply on behalf of the destination, and limit the propagation of requests.

Broch *et al.* [4] compare the performance of the DSDV, TORA, DSR, and AODV routing protocols on a simulated mobile IEEE 802.11 network. They simulate networks of 50 nodes, under a range of mobility rates and traffic loads. Their measurements show the effectiveness of DSR's caching in minimizing DSR's routing protocol traffic on these 50-node networks. In the interest of comparability of results, we use this work's simulation environment for

IEEE 802.11, a two-ray ground reflection model, and DSR.

Ko and Vaidya [17] describe Location Aided Routing (LAR), an optimization to DSR in which nodes limit the propagation of route request packets to the geographic region where it is most probable the destination is located. LAR uses base DSR to establish first connectivity with a destination; thereafter, a route querier learns the destination's location directly from the destination node, and uses this information to mark route requests for propagation only within a region of some size about the destination's last known position. Like DSR's caching, LAR is a strategy for limiting the propagation of route requests. When a circuitous path, outside the region LAR limits route request propagation within, becomes the only path to a destination, LAR reverts to DSR's flooding-with-caching base case. Under LAR, DSR's routes are still end-to-end source routes. Geography is not used for data packet forwarding decisions under LAR; only to scope routing protocol packet propagation.

Li *et al.* [18] propose GLS, a scalable and robust location database that geographically addresses queries and registrations. Their system dynamically selects multiple database servers to store each node's location, for robustness against server failure. This property also ensures that a cluster of nodes partitioned from the remainder of the network continues to have location database service, provided by nodes inside the cluster. GLS uses a geographic hierarchy to serve queries at a server topologically close to the querier.

Bose *et al.* [3] independently investigated the graph algorithms for rendering a radio network's graph planar. They suggest the Gabriel Graph, and analyze the increase in path length over shortest paths when traversing a graph using *only* perimeters. Motivated by the longer-than-optimal paths perimeter traversal alone finds, they suggest combining planar graph traversal with greedy forwarding, and verify that this combination produces path lengths closer to true shortest paths. They do not present a routing protocol, do not simulate a network at the packet level, and assume that all nodes are stationary and reachable.

5. FUTURE WORK

One assumption in the use of planar perimeters we would like to investigate further is that a node can reach all other nodes within its radio range. The GG and RNG planarizations both rely on a node's ability to accurately know if there is a witness w within radio range, when considering elimination of an edge to a known neighbor. Our use of the GG and RNG can disconnect a graph with particular patterns of obstacles between nodes. This disconnection is easily avoided by forcing the pair of nodes bordering an edge to agree on the edge's fate, with the rule that both nodes must decide to eliminate the edge, or neither will do so. However, this modification to the planarization algorithms will make the RNG and GG planarizations leave one or more crossing edges in these regions with obstacles. We intend to study these cases further. One promising approach in dealing with such obstacles may be to have obstructed nodes choose a reachable *partner* node elsewhere in the network, and route via the partner for destinations that are unreachable because of local failure of the planarization.

While we have shown herein the benefits of geography as a tool for scalable routing systems, measuring the combined behavior of GPSR and a location database system will reveal more about the costs of using geography for routing. An efficient distributed location database would provide a network service useful in many other location-aware computing applications.

A comparison of the behavior of GPSR using the RNG and GG planarizations would reveal the performance effects of the tradeoff between the greater traffic concentration that occurs in perimeter forwarding on the sparser RNG, *vs.* the increased spatial diversity that the RNG offers by virtue of its sparsity. Even outside the context of GPSR, it may be the case that limiting edges used for forwarding in a radio network to those on the RNG or GG may reduce contention and improve efficiency on MAC protocols sensitive to the number of sending stations in mutual range.

We hope to extend GPSR for hosts placed in three-dimensional space, beyond the flat topologies explored in this paper. A promising approach is to implement perimeter forwarding for 3-D *volumes* rather than 2-D faces.

6. CONCLUSION

We have presented Greedy Perimeter Stateless Routing, GPSR, a routing algorithm that uses geography to achieve small per-node routing state, small routing protocol message complexity, and extremely robust packet delivery on densely deployed wireless networks. Our simulations on mobile networks with up to 200 nodes over a full IEEE 802.11 MAC demonstrate these properties: GPSR consistently delivers upwards of 94% of data packets successfully; it is competitive with DSR in this respect on 50-node networks at all pause times, and increasingly more successful than DSR as the number of nodes increases, as demonstrated on 112-node and 200-node networks. GPSR generates routing protocol traffic in a quantity independent of the length of the routes through the network, and therefore generates a constant, low volume of routing protocol messages as mobility increases, yet doesn't suffer from decreased robustness in finding routes. DSR must query longer routes as the network diameter increases, and must do so more often as mobility increases, and caching becomes less effective. Thus, DSR generates drastically more routing protocol traffic in our 200-node and 112-node simulations than it does in our 50-node ones. Finally, GPSR keeps state proportional to the number of its neighbors, while both traffic sources and intermediate DSR routers cache state proportional to the product of the number of routes learned and route length in hops.

GPSR's benefits all stem from geographic routing's use of only immediate-neighbor information in forwarding decisions. Routing protocols that rely on end-to-end state concerning the path between a forwarding router and a packet's destination, as do source-routed, DV, and LS algorithms, face a scaling challenge as network diameter in hops and mobility increase because the product of these two factors determines the rate that end-to-end paths change. Hierarchy and caching have proven successful in scaling these algorithms. Geography, as exemplified in GPSR, represents another powerful lever for scaling routing.

7. ACKNOWLEDGEMENTS

We thank Robert Morris, whose insight greatly benefitted this work, and the anonymous reviewers for their helpful comments. Dick Karp first suggested investigating planar graphs. Brad Karp also had fruitful discussions with Mark Handley, Scott Shenker and the rest of ACIRI.

8. REFERENCES

[1] ABRAMSON, N. The ALOHA system - another alternative for computer communications. *AFIPS 37* (1970), 281–285.

[2] BHARGHAVAN, A., DEMERS, S., SHENKER, S., AND ZHANG, L. MACAW: A media access protocol for wireless LANs. In *Proceedings of the SIGCOMM '94 Conference on Communications, Architectures, Protocols, and Applications* (Sept. 1994), pp. 212–225.

[3] BOSE, P., MORIN, P., STOJMENOVIĆ, I., AND URRUTIA, J. Routing with guaranteed delivery in *ad hoc* wireless networks. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM '99), Aug. 1999.

[4] BROCH, J., MALTZ, D., JOHNSON, D., HU, Y., , AND JETCHEVA, J. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)* (Dallas, Texas, USA, Aug. 1998).

[5] CALI, F., CONTI, M., AND GREGORI, E. IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement. In *Proceedings of IEEE INFOCOM 1998* (San Francisco, California, March/April 1998), p. 142.

[6] CHANDRAKASAN, A., AMIRTHARAJAH, R., CHO, S., GOODMAN, J., KONDURI, G., KULIK, J., RABINER, W., AND WANG, A. Design considerations for distributed microsensor systems. In *Proceedings of the IEEE 1999 Custom Integrated Circuits Conference (CICC '99)* (May 1999), pp. 279–286.

[7] FINN, G. G. Routing and addressing problems in large metropolitan-scale internetworks. Tech. Rep. ISI/RR-87-180, Information Sciences Institute, Mar. 1987.

[8] FLOYD, S., AND JACOBOSON, V. The synchronization of periodic routing messages. *IEEE/ACM Transactions on Networking 2*, 2 (April 1994), 122–136.

[9] GABRIEL, K., AND SOKAL, R. A new statistical approach to geographic variation analysis. *Systematic Zoology 18* (1969), 259–278.

[10] HAAS, Z., AND PEARLMAN, M. The performance of query control schemes for the zone routing protocol. In *Proceedings of the SIGCOMM '98 Conference on Communications Architectures, Protocols and Applications* (Sept. 1998).

[11] IEEE COMPUTER SOCIETY LAN MAN STANDARDS COMMITTEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 802.11-1997, 1997.

[12] JOHNSON, D. B., AND MALTZ, D. B. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, 1996, ch. 5, pp. 153–181.

[13] KAHN, J. M., KATZ, R. H., AND PISTER, K. S. J. Mobile networking for smart dust. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)* (Seattle, WA, USA, Aug. 1999).

[14] KARN, P. MACA—a new channel access method for packet radio. In *Proceedings of the 9th Computer Networking Conference* (Sept. 1990), pp. 134–140.

[15] KARP, B. Geographic routing for wireless networks. Presentation at AFOSR MURI ACTCOMM Research Review Meeting, Oct. 1998.

[16] KARP, B. Greedy perimeter state routing. Invited Seminar at the USC/Information Sciences Institute, July 1998.

[17] KO, Y., AND VAIDYA, N. Location-aided routing in mobile ad hoc networks. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)* (Dallas, Texas, USA, Aug. 1998).

[18] LI, J., JANNOTTI, J., DECOUTO, D., KARGER, D., AND MORRIS, R. A scalable location service for geographic ad-hoc routing. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)* (Boston, MA, USA, Aug. 2000).

[19] MALTZ, D., BROCH, J., JETCHEVA, J., AND JOHNSON, D. The effects of on-demand behavior in routing protocols for multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications 17*, 8 (Aug. 1999), 1439–1453.

[20] PARK, V., AND CORSON, M. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)* (Kobe, Japan, Apr. 1997), pp. 1405–1413.

[21] PERKINS, C. Ad hoc on demand distance vector (AODV) routing. Internet-Draft, draft-ietf-manet-aodv-04.txt, Oct. 1999.

[22] PERKINS, C., AND BHAGWAT, P. Highly-dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the SIGCOMM '94 Conference on Communications, Architectures, Protocols, and Applications* (London, UK, Sept. 1994), pp. 234–244.

[23] SALTZER, J., REED, D. P., AND CLARK, D. End-to-end arguments in system design. *ACM Transactions on Computer Systems 2*, 4 (Nov. 1984), 277–288.

[24] SHEPARD, T. A channel access scheme for large dense packet radio networks. In *Proceedings of the SIGCOMM '96 Conference on Communications Architectures, Protocols and Applications* (Aug. 1996).

[25] THE CMU MONARCH GROUP. Wireless and Mobility Extensions to ns-2. http://www.monarch.cs.cmu.edu/cmu-ns.html, Oct. 1999.

[26] THE VINT PROJECT. The UCB/LBNL/VINT Network Simulator—ns (version 2). http://mash.cs.berkeley.edu/ns.

[27] TOUSSAINT, G. The relative neighborhood graph of a finite planar set,. *Pattern Recognition 12*, 4 (1980), 261–268.

[28] WARD, A., JONES, A., AND HOPPER, A. A new location technique for the active office. *IEEE Personal Communications 4*, 5 (Oct. 1997), 42–47.

[29] ZAUMEN, W., AND GARCIA-LUNA ACEVES, J. Dynamics of distributed shortest-path routing algorithms. In *Proceedings of the SIGCOMM '91 Conference on Communications Architectures, Protocols and Applications* (Sept. 1991), pp. 31–42.