Geographic Routing in Large-Scale MANETs

Dinesh Ramasamy and Upamanyu Madhow
University of California, Santa Barbara
Email:{dineshr, madhow}@ece.ucsb.edu

***Abstract*—In this paper, we propose and evaluate a geographic routing scheme, including an efficient position publish protocol and a routing protocol that can operate with imperfect information regarding the destination's location, that scales to large mobile ad hoc networks (MANETs). The traffic generated by our position publish protocol fits within the transport capacity of MANETs with constant communication bandwidth allocated for routing overhead, even as the network size increases. The routing protocol guarantees, with high probability, routes whose lengths are within a constant "stretch" factor of the shortest path from source to destination. The key idea underlying the scalability of the publish protocol is for each potential destination node to send location updates (with frequency decaying with distance) only to a subset of network nodes, structured as annular regions around it (the natural approach of updating circular regions in distance-dependent fashion does not scale). The routing protocol must therefore account for the fact that the source and/or relay nodes may not have estimates of the destination's location (or may have stale estimates). Spatial and temporal scaling of protocol parameters are chosen so as to guarantee scalability, route reliability and route stretch, and these analytical design prescriptions are verified using simulations.**

I. Introduction

Geographic routing is attractive for networks in which nodes know their own locations (e.g., using GPS) because a node only requires estimates of the locations of its immediate neighbors and of the destination node in order to forward a message. When the nodes in a network can move, a node can still maintain estimates of its neighbors' locations quite easily (the overhead for the local information exchanges for this purpose is small), but the bottleneck becomes global dissemination of information regarding the locations of moving destination nodes. As observed in prior work (discussed in more detail later), this bottleneck can be alleviated by structuring location updates such that distant nodes get fewer updates, and live with a fuzzier view of the destination's location without excessively compromising route quality. This intuition is the starting point for the present paper, which provides an approach for *provably* scaling geographic routing to large mobile ad hoc networks (MANETs), while providing performance guarantees on route suboptimality due to imperfect location information.

Scalability of any routing protocol demands that the traffic generated by routing updates be within the network transport capacity obtained with a fixed communication bandwidth, bounds on which have been established in the pioneering work of Gupta and Kumar [2]. In this paper, we provide a "position publish" scheme which potential destination nodes use for global location updates, and show that (a) if all network nodes use this publish scheme, the resulting overhead is within the Gupta-Kumar bounds, and (b) information from the publish scheme enables routing with paths of length within a constant "stretch'' factor of the shortest path from source to destination. Our contributions are summarized as follows:

- We show that, while bounded route stretch is feasible with uncertainty in destination location that scales with distance from the destination, the natural approach of simply reducing the frequency of location updates to distant nodes (which corresponds to updating *circular* regions), does not scale with network size. This implies that each potential destination must update a *subset* of network nodes regarding its locations.
- We show that scalability can be achieved by sending location updates to *annular* regions (the number of such regions scales as $\log n$, where n is the number of nodes). Counting outward, the inner radius and thickness of these rings increase exponentially with their index. Key parameters of the location publish protocol are these exponents, as well as those of parameters determining the spatial and temporal validity of a location update. We determine the constraints required on these parameters for achieving scalability.
- Our routing protocol employs greedy geographic forwarding, with relay nodes overwriting information regarding the destination's location if they have better information than what is contained in the packet. We determine constraints on the protocol parameters required to ensure that correct routing happens with high probability, and provide bounds on the worst-case route stretch in terms of these parameters.
- We provide simulation results showing that choosing protocol parameters as prescribed by our analysis provides performance within the bounds guaranteed by our analysis.

The mobility model used for our analysis is two-dimensional (2D) Brownian motion. However, we note that our scalability results are broadly applicable to a large class of mobility models: scalability requires controlling the volume of traffic due to position updates to distant nodes, and when viewed from "far enough" away, a broad class of random mobility models "look like" Brownian motion.

Before launching into a detailed exposition, we provide concrete insight into how the proposed protocol works via simulation results shown in Figure 1 (see Section VI for details). Note that,

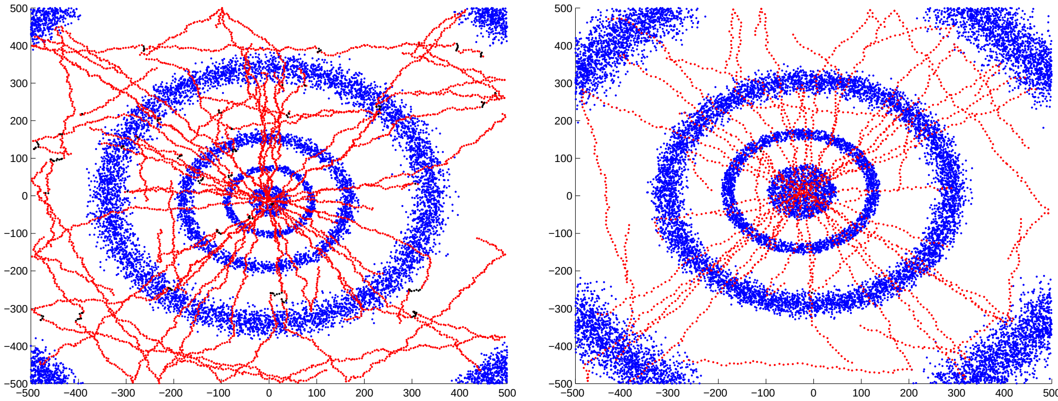

Figure 1. Overlaid routing trajectories (converging to a destination node at the center) at a snapshot of the network for the proposed routing protocol for two different communication radii. Blue dots indicate nodes with active position updates for the destination node. Red dots indicate relay nodes along packet trajectories using greedy geographic forwarding; black dots indicate greedy face traversal around voids [1]. The communication radius on the left is smaller, and is chosen to illustrate that geographic routing with imperfect location information can route around these using the same strategies as a standard setting where the destination location is perfectly known.

while our protocol is designed on the assumption that there are no voids in the network, voids can be handled using well-known techniques such as greedy face traversal [1].

A. Related Work

Since we are concerned with large-scale networks, our communication model and notion of scalability are guided by the relevant asymptotic results of Gupta and Kumar [3] [2]. We postpone detailed discussion of these to Section II.

The literature on MANET routing and on geographic routing (for stationary or mobile nodes) is vast, hence we restrict attention here to prior work that is most closely related to our approach (many of the references we cite provide good discussions on the state of the art). DREAM [4] considers geographic routing when the frequency of location updates is reduced as the distance from the updating node increases. While this intuition is the starting point for our scheme as well, we show that location updates made to all nodes as in DREAM are not scalable. A similar intuition is also behind the Hazy Sighted Link State (HSLS) algorithm in [5], in which link state updates are sent to less frequently to distant nodes. HSLS is designed based on minimization of the sum of the overhead due to route suboptimality and location updates. However, the overhead computations in [5] show that HSLS is not scalable. An intuitive reason for this is that all nodes must have a roughly consistent view of the network for successful link state routing, whereas geographic routing only requires that an appropriate subset of nodes have location updates from a given destination node. GLS [6] is a spatially hierarchical quorum based scheme for position lookups, but is not designed to work in networks with pervasive movement.

MLS[7] proposes a “lazy” hierarchical position lookup service in which updates are published to certain fixed geographical regions. It is similar in spirit to our scheme, in that it is able to guarantee a constant route stretch without requiring that all nodes in the network obtain location updates, but the updates in our scheme are published to regions which are different, in general, for different nodes. It is worth mentioning that MLS builds on an earlier scheme termed LLS[8], which structures location updates to areas centered around the destination node, as in our scheme. The key difference of [7], [8] from our work is they do not relate the routing overhead to network transport capacity, and do not provide means to vary the tradeoff between route stretch and overhead.

For the routing scheme in this paper and the preceding references, mobility is a nuisance that increases routing overhead. However, when delay in message delivery is not an issue, Grossglauser and Tse have shown in [9] that mobility can actually help us get around the transport capacity limits derived by Gupta and Kumar [2]. In a similar spirit, mobility can be exploited to reduce the overhead of location updates, as argued in [10][11]. However, this is not the regime of interest to us, since we are interested in delivering packets to their destinations with minimal delay.

II. System Model and Basic Computations

We describe the system model and provide some computations related to geographic routing with imperfect information. We consider a network of n nodes in the two-dimensional plane, with node density fixed at Λ as n gets large. Thus, the area of the deployment region $B_n = \frac{n}{\Lambda}$ grows with n.

Notation: Since we wish to prove that certain desirable properties hold even as the network size increases, we review standard notation for comparing the behavior of functions as some parameter tends to a finite or infinite limit (e.g., as the number of nodes n gets large). We say that $f(n) = O(g(n))$ if for some constant C, $f(n) \leq Cg(n)$ for sufficiently large n. We say that $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$ and $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$).

Connectivity: We assume that the communication radius for all nodes is fixed at $r = r(n)$, and that it is chosen so that the network is connected. As shown in [3], connectivity requires that $r(n)$ satisfies $\Lambda\pi r^2(n) = (1+\epsilon)\log n$, for some $\epsilon > 0$. Furthermore, it is shown in [12] that if $\epsilon > \epsilon_0$ (with $\epsilon_0 \approx 1.6$), then greedy geographic routing works with probability 1. While such a choice corresponds to a communication radius $r(n) = \Theta\left(\sqrt{\log n}\right)$, we note that we can scale down to a constant communication radius by scaling the deployment region as $\frac{n}{\log n}$ (along with suitably scaling other parameters) rather than as n.

Scalability: Using the protocol model in [2], each transmission precludes the reception of any other transmission within a disc of radius $(1+\Delta)r$. Thus, if the bandwidth available for communication is W, then the maximum number of simultaneous transmissions available per time slot, denoted by $T_A(n)$, scales as $\Theta\left(\frac{Wn}{\Lambda r^2(n)}\right)$. Denoting by $T_U(n)$ the average of the total number of simultaneous transmissions needed per unit time for the position update protocol across all nodes, we employ the following definition for the scalability of a protocol. Assuming that the transmissions are originated in a way that is uniform across nodes, we can also define the average number of transmissions on a per node basis as $t_U(n) = \frac{T_U(n)}{n}$ and $t_A(n) = \frac{T_A(n)}{n}$, respectively.

Definition 1. We define a protocol to be scalable if $T_U(n) = O(T_A(n))$, or equivalently, $t_U(n) = O(t_A(n))$. In this case, the overhead needed for the protocol be accommodated with a suitably chosen constant bandwidth W.

The preceding definition assumes that the load induced on the network as a result of position updates is uniform in space and time (this does hold for our mobility model, which is described next).

Mobility Model: Every node in the network is mobile, executing Brownian motion with reflection at the boundaries of the deployment region (assumed to be square for convenience). Thus, away from the boundaries, the two-dimensional displacement $(\Delta x, \Delta y)$ over a time interval Δt is modeled as follows: Δx and Δy are independent and identically distributed Gaussian random variables with mean zero and variance $\sigma^2\ \Delta t$. (Brownian motion is rotationally invariant, so that the preceding observations hold for any rotation of the x and y axes). The root mean square (RMS) velocity of each node is given by $v_{RMS}^2 = 2\sigma^2$. While we choose the Brownian motion model for its analytical tractability, we note that our scalability results hold more broadly: scalability depends on how distant nodes perceive the mobility of a destination node, and a large class of randomized models for local mobility look like Brownian motion when viewed from far away and at large time scales. For example, consider a version of the random

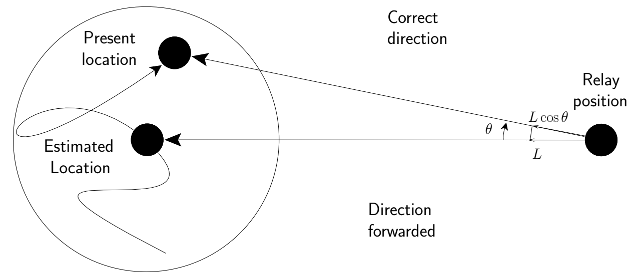

Figure 2. Rate of progress depends on the angle between the estimated and correct directions.

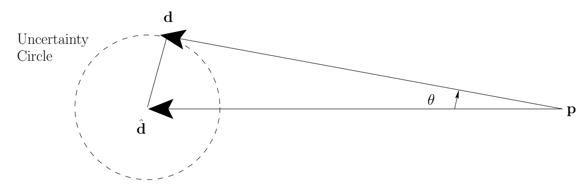

Figure 3. The destination $\mathbf{d}$ lies on an "uncertainty circle" of radius $\left\|\mathbf{d} - \hat{\mathbf{d}}\right\|$ around $\hat{\mathbf{d}}$. The worst-case angle θ occurs when $\mathbf{d}-\mathbf{p}$ is a tangent to the uncertainty circle.

waypoint model in which each node chooses a new speed V_n uniformly over $[0, v_{max}]$ and direction Φ_n uniformly over $[0, 2\pi]$ for a duration D_n, where the times $\{D_n\}$ are independent and identically distributed exponential random variables with mean $1/\lambda$. It is easy to show that, over large time scales, this model behaves like a Brownian motion random walk with $\sigma^2 = \frac{v_{max}^2}{3\lambda}$.

We now provide some basic computations that motivate the proposed protocol.

A. Geographic routing with location errors

When information about the destination's location is imperfect, the natural approach is to route the packet along the best estimate of the direction of the destination, possibly updating this estimate after each hop, until we get close enough that the destination is within the communication radius. If the angle between the correct and estimated directions is θ, then the progress towards the destination per unit distance traveled is $\cos\theta$, so that we would like θ to be small. This is depicted in Figure 2.

Suppose that a packet currently at location $\mathbf{p}$ is to be routed to a destination at location $\mathbf{d}$, and that the currently available estimate of the destination's location is $\hat{\mathbf{d}}$. We now observe that we can afford to be sloppier in our estimate of the destination's location when we are farther away. Assuming that $||\mathbf{p} - \hat{\mathbf{d}}|| > ||\mathbf{d} - \hat{\mathbf{d}}||$ (where $||\mathbf{v}||$ denotes the length of a vector $\mathbf{v}$), let us denote the *relative uncertainty* in location as

$$U = \frac{||\mathbf{d} - \hat{\mathbf{d}}||}{||\mathbf{p} - \hat{\mathbf{d}}||} \quad (1)$$

It is clear from Figure 3 that the worst-case (largest) value of θ

is given by

$$\sin\theta = \frac{||\mathbf{d}-\hat{\mathbf{d}}||}{||\mathbf{p}-\hat{\mathbf{d}}||} = U \tag{2}$$

Thus, for a given value of θ, the localization error $||\mathbf{d}-\hat{\mathbf{d}}||$ can be larger when the packet is further away (i.e., when $||\mathbf{p}-\hat{\mathbf{d}}||$ is larger.

Bounded relative uncertainty leads to bounded route stretch: Now, suppose that the relative uncertainty seen by a packet is always less than $U_0 < 1$, so that the worst-case angle between the correct and estimated directions always satisfies $\sin\theta \le U_0$. This implies that $\cos\theta \ge \sqrt{1-U_0^2}$, and that the route stretch is at most $\frac{1}{\sqrt{1-U_0^2}}$.

Thus, a natural approach to guarantee a worst-case route stretch is to employ a position update protocol that maintains relative uncertainty below a level $U_0 < 1$ throughout the network. We show in the next section that such protocols would not scale. Before doing that, we round out this section by quantifying the cost of multicasting information to a specific region (which is a basic building block for the position update protocols discussed here).

B. Cost of Multicast

We note here for future use that the minimum number of transmissions $C(A)$ needed to multicast a message to all nodes in a connected region A is $\Theta\left(\frac{|A|}{r^2(n)}\right)$. To see this, first note that, in order for every node in the region to have listened to the message at least once, $C(A) \ge \frac{|A|}{\pi r^2}$ as the space has to be "tiled" by message transmissions at least once. To provide an upper bound, we need a constructive scheme that multicasts messages to all nodes in A. If $\Lambda\pi r^2(n) = (1+\epsilon)\log n$, with ϵ a sufficiently large constant, we can use a result from [2] to tile the area *a priori* into $\Theta\left(\frac{|A|}{r^2}\right)$ tiles such that there exists at least one node per tile and every node in a tile can communicate with every other node in its tile and all the nodes in its neighboring tiles. Thus, the resultant network of tiles is connected. We designate one node per tile (say the one with the smallest node ID) to transmit and listen while others merely listen. The first instant when the designated node in each tile receives the multicast message, it airs the message to all nodes in its range in a collision free manner (using an appropriate MAC) by means of a wireless broadcast. Thus, the message is multicast to all nodes in A with $\Theta\left(\frac{|A|}{r^2}\right)$ transmissions, which proves that $C(A) = \Theta\left(\frac{|A|}{r^2}\right)$. Note that, even though nodes are mobile, the tiling of the network can be done *a priori* as in [2], and a node leader elected based on the node of smallest ID occupying the tile. (This only requires nodes to have information regarding their neighbors, which any position update protocol provides).

III. A Non-Scalability Result

While maintaining the relative uncertainty guarantees an upper bound on route stretch, we now show that maintaining a uniform relative uncertainty throughout the network, which requires updating all nodes in the network, does not scale. In order to maintain relative uncertainty of at most U_0, location updates from a particular node (say v) must reach all nodes that are a distance z away from it if it moves a distance $U_0 z$. For our two-dimensional Brownian motion model, the mean time to move this distance is $\frac{U_0^2 z^2}{2\sigma^2}$, and the average frequency of updates to these nodes is the reciprocal of this time: $\frac{2\sigma^2}{U_0^2 z^2}$. The area of a small ring at distance z is $2\pi z dz$ and, as shown in Section II-B, the minimum number of transmissions needed to inform all nodes in the region is $C(2\pi z dz) = \Theta\left(\frac{z\ dz}{r^2}\right)$. Let $t_U(v)$ be average number of transmissions allocated to v per unit time. Note that the diameter of the network is $\Theta(\sqrt{n})$.

$$\begin{aligned} t_U(v) &\ge \frac{2\sigma^2 C\left(\pi k_1^2 r^2\right)}{k_2^2 r^2} + \int_{k_1 r}^{k_3\sqrt{n}} \frac{2\sigma^2 C(2\pi z dz)}{\phi^2 z^2} \\ &= \sigma^2\frac{k_4}{r^2} + \sigma^2\frac{k_5}{r^2}\int_{k_1 r}^{k_3\sqrt{n}}\frac{dz}{z} \\ &= \sigma^2\frac{k_4}{r^2} + \sigma^2\frac{k_5}{r^2}\log\left(\frac{k_3\sqrt{n}}{k_1 r}\right) \end{aligned} \tag{3}$$

for some constants k_1, k_2, k_3, k_4 and k_5. The first term corresponds to broadcasts to a circle of radius bigger than r to ensure all nodes have accurate lists of neighbors, while the second term corresponds to the location updates to distant nodes aimed at preserving relative uncertainty. The inequality in (3) is because we have ignored the rate needed to preserve updates in space (other network nodes are mobile and so updates made to a certain region in space will not be alive in that region indefinitely). So $t_U(v) = \Omega\left(\frac{\sigma^2\log n}{r^2}\right)$. The average sum load on the network is $T_U = \Omega\left(\frac{\sigma^2 n\log n}{r^2}\right)$.

Maintaining uniform relative uncertainty does not scale: The ratio of required overhead to sustainable capacity is therefore given by

$$\frac{T_U(n)}{T_A(n)} = \Omega\left(\frac{\Lambda\sigma^2\log n}{W}\right)$$

which blows up for large n. Thus, a strategy of maintaining an upper bound on relative uncertainty throughout the network (and thereby an upper bound on route stretch) does not scale.

Clearly, in order to provide guarantees on route stretch, the angle between the true and estimated directions towards the destination cannot be too large. But what we have just shown implies that we must appropriately choose a *subset* of nodes to update in order to reduce the routing overhead enough that the protocol can scale. This observation motivates the proposed protocol described in the next section.

IV. Proposed Protocol

We now describe our position update protocol, and a corresponding geographic routing protocol that works with imperfect information.

A. Position update protocol

We first make the obvious observation that, in order to implement any geographic forwarding protocol, each node must maintain a list of its neighbors and their positions, which is essential for implementing any geographic forwarding protocol. Such neighbor lists can be maintained using periodic *local* broadcasts which can be accommodated within a constant bandwidth, and are therefore irrelevant to considerations of scalability. The main focus of our description here, and the overhead computations in the next section, therefore, is on how position updates to distant nodes are performed.

While neighbor lists must be maintained by all nodes, whether serving as source, destination or relay, position updates to distant nodes need only be performed by nodes that are potential destinations. Thus, the overhead of location updates can be significantly reduced for asymmetric networks in which only a few nodes are destinations. However, in our scalability analysis in the next section, we make the worst-case assumption that all nodes are potential destinations.

We now describe position updates performed by a typical node that is a potential destination (we call this the *destination node* henceforth). The destination node directs its updates to geographic regions structured as annular rings around its current position, indexed as $i = 0, 1, ..., K$ with increasing radii. The ith ring has inner radius of the r_i and thickness d_i centered at the current location of node i. The geographical region covered by a ring is specified by its center $\mathbf{c}$ of the ring and its index.

As described below, the validity of updates is constrained in both space and time in order to enable successful geographical forwarding. Let $\mathbf{d}(t)$ denote the position of the destination at time t. There are two kinds of location updates: normal and abnormal updates.

Normal update: A normal update at time t_0 to ring i specifies the center of the ring as $\mathbf{c} = \mathbf{d}(t_0)$, the current location as $\hat{\mathbf{d}} = \mathbf{d}(t_0)$, and has a lifetime of T_i. The update is understood to remain valid while the destination remains within a circle of radius βr_i around the current location $\hat{\mathbf{d}}$, termed the *confidence region* for the update.

Abnormal update: An abnormal update is sent when the destination leaves the confidence region for a prior update before the timer for the latter update expires. For example, for the normal update at time t_0 described above, if the destination node crosses the boundary of the confidence region at time $t_1 < t_0 + T_i$ (i.e., $||\mathbf{d}(t_1) - \mathbf{d}(t_0)|| > \beta r_i$), then we send an abnormal update to the ring centered at the *prior* update. That is, we send an update specifying the current location $\mathbf{d}(t_1)$ to a ring of index i centered at $\mathbf{d}(t_0)$ with a timer $T_i - (t_1 - t_0)$ (spanning the remaining lifetime of the invalidated update).

Ring Index	Destination Location	Ring Center	Time of Update	Lifetime of Update	Normal Update Flag
i	$\hat{\mathbf{d}}$	$\mathbf{c}$	t_U	TTL	F

Figure 4. List of active updates maintained by a destination node

As shown later, we choose our system parameters such that the probability of abnormal updates tends to zero as the ring index increases, so that they are essentially irrelevant from the point of view of scalability. However, inclusion of abnormal updates enhances routing performance by reducing the incidence of misrouting due to stale location updates.

Triggers for new updates: We focus on a specific ring index i, with the understanding that updates occur in parallel for all ring indices. A normal update is performed whenever the timer for a prior normal update expires. When a destination node moves out of the confidence region of a normal update whose timer has not expired, then *two* updates are performed: a normal update to a ring centered around the current location and an abnormal update centered around the old location (at which the invalidated normal update was made). The abnormal update, which lasts for the remaining lifetime of the invalidated update, prevents the invalidated update from influencing packet trajectories.

Spatial validity of updates: As we shall see in the description of the routing protocol, an update (whose timer has not yet expired) can only be used for geographical forwarding if the relay node is in the ring for which the update was destined. Thus, once a node moves out of that ring, it can no longer use the information it received about the destination's location when in the ring. It will use this information if it moves into the update ring again. This may appear to be overly restrictive: is it not better always to reveal to the packet the best (latest) update corresponding to the destination node? However, imposing this restriction is what enables us to guarantee that we can use abnormal updates to overwrite misleading information about the destination's location.

The destination node maintains a list of active updates published by it, so that it can publish new updates when these updates time out or become spatially invalid. Each entry in this list is of the form in the Figure 4. The kind of update is indicated by the flag F: $F = 1$ for a normal update and $F = 0$ for an abnormal update. The update procedure is summarized as Algorithm 1, which runs continuously at the destination node.

Update propagation: In order to limit the traffic generated by a location update, the destination node sends the update packet in a specified direction until it hits the ring it is intended for, at which point it "expands" into a multicast message. Specifically, the destination launches the packet in an arbitrarily chosen direction $\mathbf{u}$, which is indicated in the packet. Each intermediate node

Algorithm 1 Publish algorithm

for all updates $\left(i, \hat{\mathbf{d}}, \mathbf{c}, t_U, \mathsf{TTL}, F\right) \in$ update list **do**
  if $t > t_U + \mathsf{TTL}$ **then**
    if $F = 1$ **then**
      Publish new normal update to ring index i
      Add new update to list
    end if
    Delete the timed out update from list
  else if $\left\|\mathbf{d}(t) - \hat{\mathbf{d}}\right\| > \beta r_i$ **then**
    if $F = 1$ **then**
      Publish new normal update to ring index i
      Add new update to update list
    end if
    Publish a new abnormal update to ring index i with center $\mathbf{c}$, pointing to the current location $\mathbf{d}(t)$, and with lifetime $\mathsf{TTL} - (t - t_U)$
    Add new update to update list
    Delete the update whose guarantee is violated from the update list
  end if
end for

Node ID	Ring Index	Destin-ation Estimate	Ring Center	Inner radius of ring	Outer radius of ring	Time of Update	Lifetime of Update	Direction of relay
v	i	$\hat{\mathbf{d}}$	$\mathbf{c}$	r_i	$r_i + d_i$	t_U	TTL	$\mathbf{u}$

Figure 5. Position Update sent out to the network

examines the packet to see if it is in the specified ring. If not, it simply forwards the packet in the direction $\mathbf{u}$. Once the packet reaches a node in the update ring, that node repackages the update as a multicast packet for all nodes in the update ring. As shown later by our overhead computations, this approach leads to scalability when the protocol parameters are appropriately chosen.

B. Routing protocol

We now consider the problem of routing a packet to a destination which proactively publishes its location as described in Section IV-A. The packet contains a field indicating the destination identity, the "best" estimate of its location, and the ring index and time of update corresponding to this estimate. Intermediate nodes use this field for geographic forwarding, and are allowed to overwrite it if they have a "better" active estimate of the destination's location. An estimate is considered "better" only if its ring index is smaller than that of the packet or if its ring index is the same as that the packet, but the update is more recent than that of the packet. The ring index is given more importance than the time of update because of the guarantees given by the destination node through its layered update scheme.

If the source node does not possess an active update, then it chooses a random direction to relay the packet along: this is indicated in the packet by means of a vector indicating this direction, time of update ∞ and ring index ∞. Until the packet hits a node with an active update all intermediate nodes relay the packet along this direction. When the packet hits a node with an active estimate, it is said to have "bootstrapped". If the packet reaches the boundary of the network before bootstrap, it bounces off the boundary by reflection(this can be implemented if the boundary of the network is known to all nodes).

V. Scalability, Reliability and Stretch

We now evaluate the performance of the proposed protocol in terms of routing overhead and route stretch, with a view to extracting design guidelines for providing provable performance guarantees. The behavior of our protocol depends on four parameters: α, β, γ and μ. The inner radii of the rings scale up exponentially with ring index: $r_i = r_0 \alpha^i$, where $\alpha > 1$. So does the ring thickness, but at a slower rate: $d_i = d_0 \alpha^{\mu i}$, $0 < \mu < 1$. The timer durations also scale up exponentially: $T_i = T_0 \alpha^{\gamma i}$. Recall that the confidence region for an update to a ring of index i is given by βr_i. We now wish to determine what conditions the protocol parameters must satisfy in order to achieve scalability, reliable routing and to provide guarantees on route stretch.

Note that we must have $r_0 = r = \Theta(\sqrt{\log n})$ in order to maintain neighbor lists. To see how the number of rings scales with K, note that we need $\pi r_K^2 = \pi r_0^2 \alpha^{2K}$ roughly equal to the network area n/Λ, which yields $K = \Theta\left(\log \sqrt{\frac{n}{\Lambda r_0^2}}\right) = \Theta\left(\log n\right)$.

A. Routing Overhead

Computing the average cost of updates to a particular ring index i is the key step to computing the routing overhead. For proving scalability, it is the behavior for large i that is the most relevant. For 2D Brownian motion, it can be shown (details omitted due to lack of space) the probability of exiting a circle of radius βr_i within the timer duration T_i tends to zero for large i, as long as T_i grows slower than r_i^2. For $T_i = T_0 \alpha^{\gamma i}$ and $r_i = r_0 \alpha^i$, this is satisfied as long as $\gamma < 2$, which we henceforth take as a constraint. This implies that the rate of abnormal updates tends to zero, and that the update rate U_i for ring i is approximately $\frac{1}{T_i}$ for i large. It can in fact be shown in general that $U_i \sim k/T_i$, where k depends on $\frac{\beta r_0}{\sqrt{\sigma^2 T_0}}$, but we do not need explicit formulas for k for our purpose of evaluating scalability. For our purpose, it suffices to observe that $U_i = \Theta(1/T_i)$.

As described in Section IV-A, a position update to ring i goes on a straight line until it hits the ring, and then is multicast in the ring. The area of the ith ring ($i \geq 1$) is $A_i = \pi\left((r_i + d_i)^2 - r_i^2\right) = \pi d_i^2 + 2\pi r_i d_i = \Theta\left(r_i d_i\right)$ (since the radius r_i scales faster than the thickness d_i). For ring 0 (or nearest neighbors), we have $A_0 = \pi r_0^2$, but we can ignore this for scalability computations. From Section II-B, we know that the number of transmissions to multicast in this area is

$C(A_i) = \Theta\left(\frac{|A_i|}{r^2}\right)$. Proceeding along the straight line takes $\Theta\left(\frac{r_i}{r}\right)$ transmissions, which can be ignored in comparison to the preceding. Thus, the number of transmissions for an update to the ith ring is $\nu_i = \Theta(\frac{r_i d_i}{r^2})$ Note that the communication radius $r = r(n) = \Theta(\sqrt{\log n})$.

The average rate of transmissions corresponding to updates for a typical destination node, which we term the *average overhead rate,* is therefore given by

$$t_U = \sum_{i=1}^{K} U_i \nu_i = \Theta\left(\sum_{i=1}^{K} \frac{r_i d_i}{r^2 T_i}\right)$$

Plugging in the scaling for r_i, d_i and T_i, we obtain that the average overhead rate is given by

$$t_U = \Theta\left(\frac{1}{r^2}\sum_{i=1}^{K} \frac{r_0 \alpha^i d_0 \alpha^{\mu i}}{T_0 \alpha^{\gamma i}}\right)$$

As n gets large, so does the number of rings K, so that the preceding summation converges if and only if $\alpha^{1+\mu-\gamma} < 1$. Since $\alpha > 1$ in order to exponentially expand the rings, we must have $1 + \mu - \gamma < 0$ as a necessary condition. Further note that as the network size increases r_0 has be atleast $r(n)$ and to keep the probability of errors in the neighbor lists constant $\sigma\sqrt{T_0} = \Theta\left(r(n)\right)$. One also has choose $d_0 = \Theta\left(r(n)\right)$ for reliable routing as will be shown later. Once this condition is satisfied, we have that

$$t_U = \Theta\left(\frac{\sigma^2}{r^2}\right) = \Theta\left(\frac{\sigma^2}{\log n}\right)$$

which matches the throughput available per node t_S for a fixed bandwidth. Notice that $\frac{t_U}{t_S} = \Theta\left(\frac{\Lambda\sigma^2}{W}\right)$ We have therefore obtained the following condition for scalability.

Condition for scalability: $1+\mu < \gamma < 2$, where the upper bound on γ was needed to drive to zero the rate of abnormal updates.

B. Routing Reliability

We have analyzed the update protocol to determine conditions for scalability. We now analyze the routing protocol to determine conditions that ensure correct routing with high probability. After an update is made to nodes in a ring, some of these nodes may leave the ring. When a packet being routed to the destination hits the ring, therefore, the relay nodes it sees may be ones which moved in after the currently active update was made. According to our routing protocol, when the packet meets such nodes which have estimates of the destination's location worse than its own (including not having any estimate of the destination's location), it simply continues in the direction it is going. Thus, in order for a packet to take advantage of an active update for ring i once it hits it, it suffices that at least one of the nodes it meets as it is cutting through the ring has an active update corresponding to ring i. If this does not happen, we say that the packet has missed the ith ring. The lifetime T_i of normal updates must be short enough that the probability of a miss tends to zero, which imposes additional conditions on the protocol parameters, as we show here.

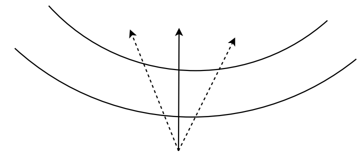

Figure 6. Different incidences of a packet on a ring

From Figure 6, the worst-case scenario for missing a ring is when the packet is relayed radially across it, since it meets fewer relay nodes along the ring, and hence a smaller probability of meeting a node with an active update. Furthermore, the packet is least likely to meet a relay node with an active update just before the update is about to expire, at time T_i after it was made. The density of nodes in the network, and in the ring, is Λ, and this remains invariant under our Brownian motion model. For $r_i \le a \le r_i + d_i$, let us denote by $\Lambda_U(a,t)$ the "update density", or the density of nodes in the ring with active updates, where $t = 0$ is the time of the update. At $t = 0$, all nodes in the ring have active updates, so that $\Lambda_U(a,0) = \Lambda I_{[r_i, r_i+d_i]}(a)$, where I_B denotes the indicator function of a set B. As time proceeds, the positions of the nodes with active updates is smeared out using a Gaussian kernel induced by 2D Brownian motion, so that

$$\Lambda_U\left(a,t\right) = \Lambda_U\left(a,0\right) \otimes \mathcal{N}\left(\mathbf{0}, \sigma^2 t I_{2\times 2}\right) \tag{4}$$

where $\otimes$ stands for 2D convolution. Let $\Lambda^*(a) = \Lambda_U(a, T_i)$ be the worst-case update density (just before the update expires). When a packet meets a node at radius a at time t, the worst-case probability that it does not get an active update is therefore given by $1 - \frac{\Lambda^*(a)}{\Lambda}$. Assuming a radial cut through, the packet meets $\frac{d_i}{r}$ such nodes (where r is the communication radius), and a miss occurs if none of these have an active update. We therefore obtain that the miss probability for the ith ring satisfies

$$P_{miss}(i) \le \prod_{\ell=1,\ldots,\frac{d_i}{r}} \left(1 - \frac{\Lambda^*\left(r_i + \ell r\right)}{\Lambda}\right)$$

For i large, $r/d_i \to 0$. Taking logarithms, the product becomes a sum which we then approximate as an integral, using $r/d_i \to 0$ for i large. Further since the ring thickness scales very slowly $d_i/\sigma\sqrt{T_i} \to 0$ for large i. So the probability that any single relay node in the ring contains an active update is very small. So we approximate $\log(1 - \Lambda^*/\Lambda)$ as $-\Lambda^*/\Lambda$.

$$\log P_{miss}(i) \leq \frac{1}{r} \sum_{l=1,\ldots,\frac{d_i}{r}} r \log\left(1 - \frac{\Lambda^*(r_i + lr)}{\Lambda}\right)$$
$$\overset{\frac{r}{d_i}\downarrow}{\approx} \frac{1}{r}\int_{r_i}^{r_i+d_i} \log\left(1 - \frac{\Lambda^*(a)}{\Lambda}\right) da$$
$$\overset{\frac{d_i}{\sigma\sqrt{T_i}}\downarrow 0}{\approx} -\frac{1}{r}\int_{r_i}^{r_i+d_i} \frac{\Lambda_U(a)}{\Lambda} da \quad (5)$$

From (4), the worst-case update density is given by

$\Lambda^*(a)$

$$= \Lambda \int_0^{2\pi} \int_{r_i}^{r_i+d_i} \exp\left(\frac{-a^2 - \rho^2 + 2a\rho\cos(\theta - \phi)}{2\sigma^2 T_i}\right) \frac{\rho\, d\rho\, d\theta}{2\pi\sigma^2 T_i}$$
$$= \frac{\Lambda}{\sigma^2 T_i} \int_{r_i}^{r_i+d_i} \exp\left(\frac{-a^2 - \rho^2}{2\sigma^2 T_i}\right) I_0\left(\frac{a\rho}{\sigma^2 T_i}\right) \rho\, d\rho$$

where $I_0(\cdot)$ denotes the zeroth order modified Bessel function of the first kind. We now plug this into (5) and use the following asymptotics to evaluate the logarithm of the miss probability: $r_i/\sqrt{T_i} \to \infty$, $d_i/r_i \to 0$ and $d_i/\sqrt{T_i} \to 0$ to obtain

$$\log P_{miss}(i)$$
$$\leq \frac{-1}{\sigma^2 r T_i} \int_{r_i}^{r_i+d_i}\int_{r_i}^{r_i+d_i} \exp\left(\frac{-a^2 - \rho^2}{2\sigma^2 T_i}\right) I_0\left(\frac{a\rho}{\sigma^2 T_i}\right) a\, da\, d\rho$$
$$\overset{\frac{r_i}{\sigma\sqrt{T_i}}\uparrow\infty}{\approx} -\frac{1}{\sigma r\sqrt{2\pi T_i}} \int_{r_i}^{r_i+d_i}\int_{r_i}^{r_i+d_i} \sqrt{\frac{a}{\rho}} \exp\left(\frac{-(a-\rho)^2}{2\sigma^2 T_i}\right) da\, d\rho$$
$$\overset{\frac{d_i}{r_i}\downarrow 0}{\approx} -\frac{\sigma\sqrt{T_i}}{r\sqrt{2}} \int_{R_{i,1}}^{R_{i,2}}\int_{R_{i,1}}^{R_{i,2}} \frac{2}{\sqrt{\pi}} \exp\left(-(a-\rho)^2\right) da\, d\rho$$
$$= -\frac{\sigma\sqrt{T_i}}{r\sqrt{2}} \times 4 \times \int_0^{\frac{d_i}{2\sigma\sqrt{T_i}}} \operatorname{erf}(x)\, dx$$
$$\overset{\frac{d_i}{2\sigma\sqrt{T_i}}\downarrow 0}{\approx} -\frac{4\sigma\sqrt{T_i}}{r\sqrt{2}} \int_0^{\frac{d_i}{2\sigma\sqrt{T_i}}} \frac{2}{\sqrt{\pi}} x\, dx$$
$$= -\frac{d_i^2}{\sigma r\sqrt{2\pi T_i}} \quad (6)$$

where we want the right-hand side to go to $-\infty$ to drive the probability of miss to zero. This will happen for the outer rings naturally if we let $d_i^2/\sqrt{T_i}$ to blow up as i increases. However for reliability, we need guarantees on the smaller indexed rings as well. In Section V-A, we already established that $r_0 = \Theta(r)$ and $\sigma\sqrt{T_0} = \Theta(r)$. So if $d_0 = \Theta(r(n))$ then reliability guarantees can be given for all network sizes.

Condition for reliable routing: $\gamma < 4\mu$.

C. Route Stretch

In order to bound the route stretch, we must account for the fact that, since location updates are sent to only a subset of nodes, the source node need not have an active update for the destination. In this case, the packet travels an additional distance in a random direction until it hits a node with an active update. Our bound on route stretch must account for this additional distance. Once the packet does encounter a node with an active update, we use uncertainty-like measures to bound the route stretch provided the packet does not miss smaller indexed rings thereafter. Using these ideas, we can bound the route stretch as follows:

$$\mathsf{S} = \sqrt{1 + \left(\sqrt{\alpha^2\left(\frac{1+\beta}{1-\beta}\right)^2 - 1} + \sqrt{\frac{\alpha^2\left(\frac{1+\beta}{1-\beta}\right)^2}{1 - \left(\frac{\alpha\beta}{1-\beta}\right)^2}}\right)^2} \quad (7)$$

Details of the derivation can be found in the appendix.

D. Protocol parameter choices

The protocol parameters are $\alpha > 1$, $0 < \beta < 1$, $\gamma > 0$ and $0 < \mu < 1$. We have now shown that scalability requires that $1 + \mu < \gamma < 2$, that reliable routing requires that $\gamma < 4\mu$, and that the worst-case route stretch is bounded if $\alpha\beta < 1$. In our simulation results in the next section, we use the following example set of parameters satisfying the preceding conditions: $\alpha = 2$, $\beta = 0.2$, $\mu = 0.55$, $\gamma = 1.95$.

VI. Simulation results

We perform simulations of the position publish and routing protocol for a particular destination node, for the following parameters:

- Number of nodes $n = 10^5$
- Node density $\Lambda = 0.1$.
- Mobility model used is 2D Brownian motion with parameter $\sigma^2 = 12$.
- Area of the square deployment area n/Λ
- Communication radius $r = \sqrt{(1+\epsilon)\log n/\Lambda\pi}$. Both $\epsilon = 0$ and $\epsilon = 2$ were simulated
- Parameters of order-zero ring $r_0 = 6.66 \times r$, $d_0 = 1.6 \times r$, $r/\sigma\sqrt{T_0} = 2.16$. Separate broadcasts to discs of radii r with lifetimes smaller than r^2/σ^2 have to be made so that the neighbor tables can be maintained. But the cost incurred due to these local broadcasts is small and can be accommodated with some additional bandwidth
- Protocol parameters $\alpha = 2$, $\beta = 0.2$, $\mu = 0.55$, $\gamma = 1.95$
- Simulation time step $\Delta t = T_0/3$

To get a concrete sense of what these numbers mean, suppose that we set the communication radius to $r = 100$m for $\epsilon = 0$ (which means that $r = 173$m for $\epsilon = 2$). The unit of length can then be computed to be 16.5 m, and the deployment area is 16.5

km by 16.5 km, with a node density of 366.4 nodes per square km. The rms motion is $\sigma\sqrt{t}$ units of length. Choosing the unit of time to be 25 seconds gives us that the rms motion over a second is about 10 meters, which is consistent with vehicular speeds. We omit further elaboration of physical interpretations due to lack of space, except to note that the results in this section imply that the protocol can handle fairly rapid mobility with relatively small overhead.

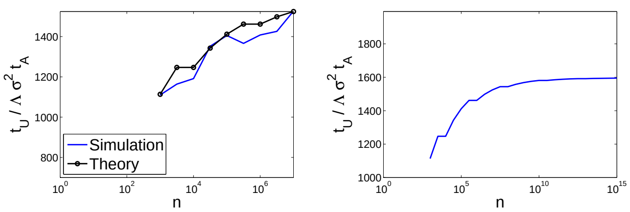

Figure 7. Scaled ratio of required to available throughput, given by $\frac{t_U(n)}{\Lambda\sigma^2 t_A(n)}$. We compare the theoretically computed value with the simulation for moderately large n (left), and show that the theoretically computed value plateaus for very large n (right).

Scalability: We first simulate the execution of the position publish scheme for a moving destination node, computing the number of transmissions per unit time as a function of the number of other nodes in the network. We estimate the rate of updates $t_U(n)$ by averaging the transmissions needed over time. For a fixed bandwidth, the ratio of update rate to available throughput per node, $\frac{t_U(n)}{t_A(n)}$ has been shown to scale with $\Lambda\sigma^2$ in Section V-A. Thus, we plot $\frac{t_U(n)}{\Lambda\sigma^2 t_A(n)}$ in Figure 7 (communication radius set according to $\epsilon = 2$).

Probability of a packet missing updated nodes in a ring: We now compare the asymptotic estimate of mis probability in (6) with empirical values of $P_{miss}(i)$ obtained from simulations. Note that the match is better for $\epsilon = 0$, because the average number of relay nodes seen by a packet decreases with an increase in ϵ and approaches the lower bound of d_i/r used in the derivation of (6). While the probability of miss decreases quite slowly with ring index, as we discuss next, our routing protocol is robust enough to handle misses.

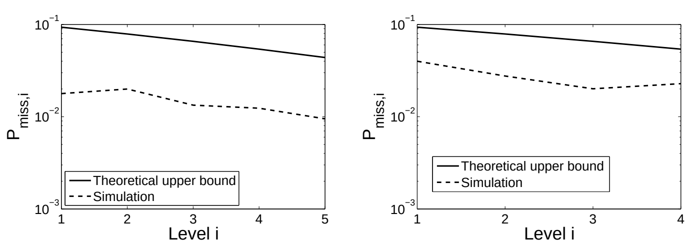

Figure 8. Probability of missing a ring for r corresponding to $\epsilon = 0$ (left) and $\epsilon = 2$ (right)

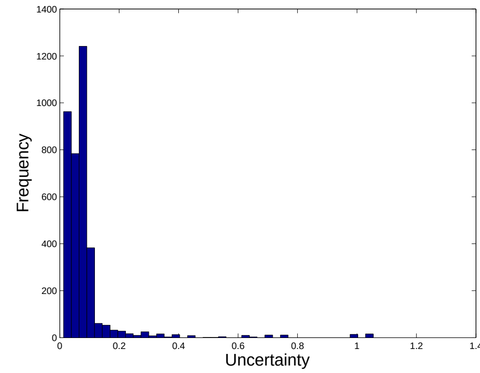

Figure 9. Uncertainty seen by packets as they cut through the network for $\epsilon = 2$

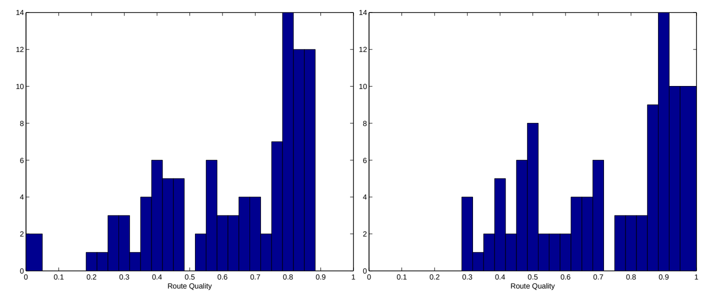

Figure 10. Histograms of reciprocal of route stretch, $\frac{1}{S}$ for communication radius r corresponding to $\epsilon = 0$ on the left and $\epsilon = 2$ on the right

Trajectories: We now refer back to the simulated trajectories previewed in Figure 1. We see that, for larger communication radius ($\epsilon = 2$), the trajectories are straighter, as there are many nodes available in each direction around a relay node. For $\epsilon = 0$, which is below the threshold [12] for asymptotic success of greedy geographic forwarding, trajectories do hit voids. However, using the standard technique of greedy left hand traversal of voids [1] (these segments of the trajectory are marked in black), route failure rates are reduced to a relatively small level.

Relative location uncertainty: The nominal relative location uncertainty defined in (1) is designed to be less than $\frac{\alpha\beta}{1-\beta} = 0.5$. From our simulations, we find that the uncertainty seen by a packet remains smaller than this value at large distances from the destination. The uncertainty can increase beyond the nominal value close to the destination, since small movements can cause large changes in the relative uncertainty, but it is the behavior at large distances from the destination that impacts scalability and route stretch. These results are shown in Figure 9.

Route Stretch: Figure 10 plots histograms of the *reciprocal* of the route stretch attained, with the bin corresponding to 0 indicating a routing failure. From (7), the worst case reciprocal of route stretch is 0.16. Note that all successful routes satisfy this guarantee, and that all routes are successful for $\epsilon = 2$. For $\epsilon = 0$ (greedy geographic forwarding not guaranteed to work), a small but significant fraction of route failures do occur.

VII. Conclusions

To the best of our knowledge, this is the first work that provides a provably scalable geographic routing protocol while providing guarantees on route stretch. Key to scalability is a probabilistic approach to updating a *subset* of nodes, and to geographic routing with imperfect information. Our emphasis here was on providing analytical insight and design criteria, verified by simulations. Mapping our ideas to practice require detailed protocol specifications at the level of packet level format and processing, and more extensive simulations for a wide variety of mobility models. In addition, while we focus on distant nodes in proving scalability, it may be possible to significantly optimize our protocol as the distance to the destination decreases. Finally, it is interesting to note that, while we have assumed a uniform set of protocol parameters for all destination nodes to prove scalability, in practice, each potential destination can choose its parameters differently, depending on the tradeoffs between routing overhead, reliability and stretch that it desires to obtain.

Appendix

To provide bounds on the worst case route stretch, we need to understand the geometry of the update rings surrounding a destination node that executes the publish algorithm. The publish algorithm ensures that at all times, there exists exactly one set of active normal updates corresponding to each ring index. All these active normal updates satisfy their confidence region guarantees. All other active updates are abnormal updates that are present to prevent older updates that were made stale by atypically large movements of the destination node from misdirecting the routing algorithm. Denote the center of the active normal update corresponding to ring index i as $\mathbf{c}_i$. This is also the position estimate of the destination given by this update. Note that all active abnormal updates(if any) of ring index i also point to $\mathbf{c}_i$. These abnormal updates are issued to past update rings whose updates are stale but have not yet timed out. When a packet is relayed through an update ring to which updates that were issued earlier have become stale, these abnormal updates ensure that the packet latches on to the newer estimate that they possess rather than the stale estimate in the nodes that posses the older update whose guarantee has been violated. This is because the routing algorithm prefers newer updates in time when it is presented with a tie in terms of the ring indices. The analysis in the derivation of $P_{miss}(i)$ holds here for the probability of missing the *newer* updates(in regions with stale updates that also have the same spatial validity). So in effect the role of abnormal updates is to prevent routing failures as a result of misdirection from stale updates. They also have an impact in increasing routing performance in the following ways: They increase the number of nodes with updates corresponding to their ring index. This decreases the probability of a packet not obtaining updates corresponding to this ring index when being routed through the network. They also increase the region where the updates of a particular ring index are available to packets. As a consequence a packet routed to the destination may potentially have the update corresponding to a ring index even before it reaches the normal update ring. This is expected to improve routing performance as outer rings have coarser guarantees in general and so, if a packet latches to a smaller indexed ring earlier, it is expected to improve routing performance. As a result of the above reasons, we can assume that abnormal updates and stale updates which were compensated for using abnormal updates are absent in a discussion of worst case routing stretch guarantees.

Consider a destination node which is surrounded by only a normal update ring of each ring index. All of these update rings satisfy their confidence region guarantees. Suppose the destination node is at the origin $\mathbf{0}$. The centers(and therefore the corresponding position updates) of the i-th update ring is $\mathbf{c}_i$. Since these position update rings all satisfy their confidence region guarantees, $\|\mathbf{c}_i\| \leq \beta r_i$. Suppose a packet originates for the destination node at $\mathbf{s}$. If the source node has an active estimate of the destination node, then the packet proceeds towards the destination. However if it does not posses an active update, the source launches the packet in some direction $\mathbf{u}$ and the packet bootstraps at $\mathbf{b}$. This process of bootstrapping can increase the route stretch. Firstly, we will examine the contribution of the lazy position updates to the route stretch after bootstrap. An upper bound on this is given via the worst case uncertainty seen by a packet. Then we provide an upper bound on the worst case contribution to route stretch due to the bootstrap process which corresponds to a confluence of unfavorable geometric configurations of the immediate inner and outer rings surrounding the packet source and launch direction chosen by it.

A. Inner and outer envelope of updates

In the following computations it is useful to define the concept of inner and outer envelope of an update of ring index i. An update of ring index i has as its center $\mathbf{c}_i$ such that $\|\mathbf{c}_i\| \leq \beta r_i$ and radius r_i. The outer envelope of updates of ring index i is defined as the set of points farthest from the source node that can possess a spatially active update of ring index i. We neglect d_i in this computation as $\frac{d_i}{r_i} \lll 1$. Similarly the inner envelope of updates of ring index i is defined as the set of points closest to the source node that can possess a spatially active update of ring index i. In each direction the farthest spatially valid update can be $(1+\beta)\, r_i$ away corresponding to $\|\mathbf{c}_i\| = \beta r_i$ and $\mathbf{c}_i$ in the same direction. So the outer envelope of updates of ring index i is the circle of radius $(1+\beta)\, r_i$ centered around the destination node at $\mathbf{0}$. Similarly the closest spatially valid update of ring index i in any direction cannot be closer than $(1-\beta)\, r_i$. This corresponds to $\|\mathbf{c}_i\| = \beta r_i$ and $\mathbf{c}_i$ in the opposite direction. So the inner envelope of updates is the circle of radius $(1+\beta)\, r_i$ centered around the destination node at $\mathbf{0}$.

B. Worst cast stretch after bootstrap

What is the worst case uncertainty seen by a packet after it has bootstrapped? To answer this question consider a segment of any packet trajectory from just after it has acquired the update for the $i+1$-th ring till it acquires the update for the i-th ring. In this segment of the packet trajectory, the packet is routed using the estimate from the $i+1$-th ring and so the estimate $\mathbf{c}_{i+1}$ can disagree from the true destination address by not more than $\beta r_{i+1} = \alpha\beta r_i$. Consider the scenario in Figure 3. Instead of uncertainty, consider the ratio $R = \frac{\left\|\mathbf{d}-\hat{\mathbf{d}}\right\|}{\|\mathbf{p}-\mathbf{d}\|}$. Even though it is not possible for a packet to discern this ratio as it can for uncertainty(the packet only needs to know the radius of the confidence region, βr_i, to compute uncertainty, which can be appended to the position update), this ratio can also establish a bound for the packet stretch. Let θ be the angle between the direction of relay and the true direction of the source node. Proceeding along similar lines to the earlier discussion on uncertainty, we arrive at the following: For a particular R, $\sin\theta_{worst} = R$. So if $R \leq R_0$, the route stretch will be less than $\frac{1}{\cos\theta_{worst}} = \sqrt{\frac{1}{1-R_0^2}}$. The farthest from the destination node that an update of index $i+1$ can be obtained by the packet is given by $(1+\beta)\, r_{i+1}$, the radius of the outer envelope of the $i+1$th ring and the closest the packet can get to the destination without latching on to the ith ring update is $(1-\beta)\, r_i$, the radius of inner envelope of updates of the ith ring. This is the region where a packet can use the $i+1$-th ring of updates. In this region, if a packet uses the update corresponding to the $i+1$-th ring, the ratio R satisfies:

$$\begin{aligned} R &\leq \max_{(1-\beta)r_i \leq \|\mathbf{p}-\mathbf{d}\| \leq (1+\beta) r_i} \frac{\beta r_{i+1}}{\|\mathbf{p}-\mathbf{d}\|} \\ &= \frac{\alpha\beta}{1-\beta} \end{aligned}$$

So $R \leq R_0 = \frac{\alpha\beta}{1-\beta}$. Note that this is independent of the ring index i. As long as the packet does not "miss" any update ring that it is relayed through, the route stretch after bootstrap is atmost $\sqrt{\frac{1}{1-\left(\frac{\alpha\beta}{1-\beta}\right)^2}}$.

For a fixed R,

$$U \leq \max_{\hat{\mathbf{d}}:\left\|\mathbf{d}-\hat{\mathbf{d}}\right\| = R\|\mathbf{p}-\mathbf{d}\|} \frac{\left\|\mathbf{d}-\hat{\mathbf{d}}\right\|}{\left\|\mathbf{p}-\hat{\mathbf{d}}\right\|} = \frac{R}{1-R}$$

So an upper bound on uncertainty is given by $U_0 = \frac{\alpha\beta}{1-(\alpha+1)\beta}$. But we use $\sqrt{\frac{1}{1-R_0^2}}$ in the worst case stretch computations as it is a tighter bound.

C. Bootstrapping cost

Consider a packet originating from $\mathbf{s}$ for the destination at the origin. Suppose the rings that surround the source are the i-th and $i+1$-th rings. There will always exist an inner and outer update ring corresponding to the destination node around any location in space. Further a packet launched in any direction first "hits" one of these rings. Since we are interested in route stretch, which is a ratio of distances, we can scale all distances by r_i. Let $\|\mathbf{s}\| = x$. $1-\beta \leq x \leq \alpha(1+\beta)$ as the source is surrounded by the i-th and $i+1$-th ring and this is possible only if the source is inside the region between the inner envelope of the i-th ring the outer envelope of the $i+1$-th ring. The worst case scenario for a packet launched by the source node is the result of multiple unfavorable scenarios. We find an upper bound on the effect of the worst case configuration using the idea of envelopes of possible update rings of a particular index. The update ring to the inner ring lies between its inner and outer envelope rings of radii $1-\beta$ and $1+\beta$ as distances are scaled by r_i. Similarly the outer ring lies between its inner and outer envelope rings of radii $\alpha(1-\beta)$ and $\alpha(1+\beta)$. When the packet is launched from the source in an arbitrary direction, the scenario where the packet latches to the ring closer to the destination is a better case. So consider the case where the packet bootstraps at the outer ring. We note that for all distances of the source node from the destination node the packet travels the farthest distance before bootstrapping if it is launched tangential to the inner envelope of the inner ring and it hits the outer envelope of the outer ring. Let the point where the trajectory is a tangent to the inner envelope of the inner ring be $\mathbf{t}$. This is the worst case scenario and is depicted in Figure 11. We denote the point on the outer envelope of the outer ring where the packet bootstraps as $\mathbf{b}$. The packet travels $\|\mathbf{b}-\mathbf{s}\|$ before bootstrap and after bootstrap it travels atmost $\|\mathbf{b}\| / \sqrt{1-R_0^2}$. Let θ be the angle between the launching direction and the direction of the destination. i.e., the angle between the vectors $\mathbf{b}-\mathbf{s}$ and $-\mathbf{s}$. Then $\sin\theta = \frac{1-\beta}{x}$. Let ϕ be the angle between the vectors $\mathbf{t}$ and $\mathbf{b}$ as shown in Figure 11. Then we have $\cos\phi = \frac{1-\beta}{\alpha(1+\beta)}$

From the cosine formula,

$$\begin{aligned} &\|\mathbf{b}-\mathbf{s}\|^2 \\ &= x^2+\alpha^2(1+\beta)^2 - 2x\alpha(1+\beta)\cos\left(\tfrac{\pi}{2}-\theta+\phi\right) \\ &= x^2+\alpha^2(1+\beta)^2 - 2x\alpha(1+\beta)\sin(\theta-\phi) \\ &= x^2+\alpha^2(1+\beta)^2 - 2x\alpha(1+\beta)\sin\theta\cos\phi \\ &\qquad +2x\alpha(1+\beta)\sin\phi\cos\theta \\ &= x^2+\alpha^2(1+\beta)^2 - 2(1-\beta)^2 \\ &\qquad +2\sqrt{\left(x^2-(1-\beta)^2\right)\left(\alpha^2(1+\beta)^2-(1-\beta)^2\right)} \\ &= \left(\sqrt{x^2-(1-\beta)^2}+\sqrt{\alpha^2(1+\beta)^2-(1-\beta)^2}\right)^2 \end{aligned}$$

So the worst case distance traveled before bootstrap is $\sqrt{x^2-(1-\beta)^2}+\sqrt{\alpha^2(1+\beta)^2-(1-\beta)^2}$. After bootstrap at

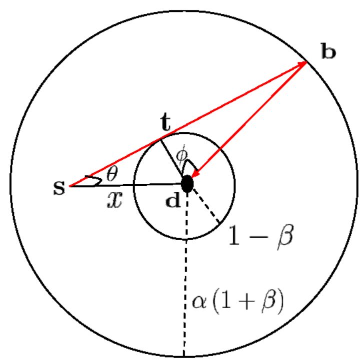

Figure 11. Contribution of bootstrapping to stretch via envelopes

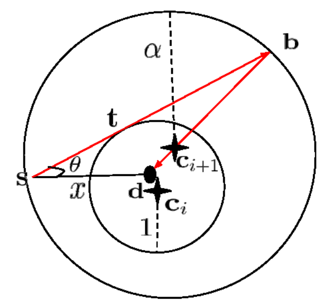

Figure 12. A configuration of ring centers $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$ marked as $+$ which satisfies the confidence region guarantees and has the same bootstrapping cost as the envelope based calculations in Figure 11

the outer ring, the packet travels atmost $\frac{\|\mathbf{b}\|}{\sqrt{1-R_0^2}} = \frac{\alpha(1+\beta)}{\sqrt{1-\left(\frac{\alpha\beta}{1-\beta}\right)^2}}$. Since $\|\mathbf{s}\| = x$, the route stretch is atmost S

$$\begin{aligned}\mathsf{S} &= \max_{1-\beta\le x\le\alpha(1+\beta)} \frac{\sqrt{x^2-(1-\beta)^2}}{x} \\ &\quad +\frac{1}{x}\left(\sqrt{\alpha^2(1+\beta)^2-(1-\beta)^2}+\frac{\alpha(1+\beta)}{\sqrt{1-\left(\frac{\alpha\beta}{1-\beta}\right)^2}}\right) \\ &= \sqrt{1+\left(\sqrt{\alpha^2\left(\frac{1+\beta}{1-\beta}\right)^2-1}+\sqrt{\frac{\alpha^2\left(\frac{1+\beta}{1-\beta}\right)^2}{1-\left(\frac{\alpha\beta}{1-\beta}\right)^2}}\right)^2}\end{aligned}$$

Note that even though we consider worst case envelopes of rings for this discussion, this scenario can be mapped to a configuration of inner and outer update rings as is shown in Figure 12 because each point on the envelopes correspond to a certain choice of the corresponding ring center. The center of the inner ring is $\mathbf{c}_i = -\frac{\beta}{1-\beta}\mathbf{t}$ and that of the outer ring $\mathbf{c}_{i+1} = \frac{\beta}{1+\beta}\mathbf{b}$. Note that the same launch trajectory is now a tangent to the inner ring with center as specified. Further $\|\mathbf{c}_i\| = \beta$ and $\|\mathbf{c}_{i+1}\| = \alpha\beta$, which satisfy their confidence region guarantees(as after scaling space, $r_i = 1$ and $r_{i+1} = \alpha$).

REFERENCES

[1] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254. [Online]. Available: http://doi.acm.org/10.1145/345910.345953

[2] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388 –404, mar 2000.

[3] ——, "Critical power for asymptotic connectivity," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 1, 1998, pp. 1106 –1110 vol.1.

[4] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '98. New York, NY, USA: ACM, 1998, pp. 76–84. [Online]. Available: http://doi.acm.org/10.1145/288235.288254

[5] C. A. Santiváñez, R. Ramanathan, and I. Stavrakakis, "Making link-state routing scale for ad hoc networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, ser. MobiHoc '01. New York, NY, USA: ACM, 2001, pp. 22–32. [Online]. Available: http://doi.acm.org/10.1145/501417.501420

[6] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 120–130. [Online]. Available: http://doi.acm.org/10.1145/345910.345931

[7] R. Flury and R. Wattenhofer, "MLS: an efficient location service for mobile ad hoc networks," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 226–237. [Online]. Available: http://doi.acm.org/10.1145/1132905.1132931

[8] I. Abraham, D. Dolev, and D. Malkhi, "LLS: a locality aware location service for mobile ad hoc networks," in *Proceedings of the 2004 joint workshop on Foundations of mobile computing*, ser. DIALM-POMC '04. New York, NY, USA: ACM, 2004, pp. 75–84. [Online]. Available: http://doi.acm.org/10.1145/1022630.1022643

[9] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, pp. 477–486, August 2002. [Online]. Available: http://dx.doi.org/10.1109/TNET.2002.801403

[10] M. Grossglauser and M. Vetterli, "Locating mobile nodes with EASE: learning efficient routes from encounter histories alone," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 457–469, June 2006. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.876204

[11] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *Proceedings of the 4th ACM international symposium on*

Mobile ad hoc networking & computing, ser. MobiHoc '03. New York, NY, USA: ACM, 2003, pp. 257–266. [Online]. Available: http://doi.acm.org/10.1145/778415.778446

[12] P.-J. Wan, C.-W. Yi, F. Yao, and X. Jia, "Asymptotic critical transmission radius for greedy forward routing in wireless ad hoc networks," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 25–36. [Online]. Available: http://doi.acm.org/10.1145/1132905.1132909