

Can geographic routing scale when nodes are mobile?

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

Abstract—We begin by asking whether geographic routing can scale when nodes are mobile; that is, can the overhead involved in tracking node locations be accommodated within the transport capacity of large-scale mobile ad hoc networks (MANETs)? We answer this question in the affirmative by proposing an efficient position publish protocol which fits within the transport capacity and a routing protocol that operates with imperfect information of the destination’s location. 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 then 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.

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, 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,

This work was supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

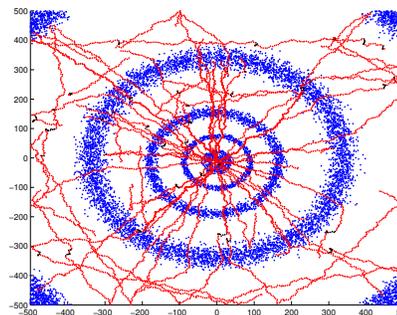


Fig. 1. Overlaid routing trajectories (converging to a destination node at the center) at a snapshot of the network for the proposed routing protocol. Blue dots indicate nodes with active position updates for the destination node. Red dots indicate relay nodes along packet trajectories using greedy geographic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

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. Scalability is achieved by sending location updates to *annular* (rather than circular) 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. 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 so as to achieve scalability and correct routing, and provide bounds on the worst-case route stretch.

Typical paths taken by a packet to a destination using our protocol are shown in Fig. 1 as a preview, prior to a more detailed discussion. Note that, 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]. Mobility can improve scalability if large delays are allowed [4] and in a similar spirit, mobility can be exploited to reduce the overhead of location updates, as argued in [5], [6]. This is not the regime of interest to us: we are interested in delivering packets to their destinations with minimal delay (quantified via the notion of route stretch), hence mobility is a nuisance for us.

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. With the aim of reducing the position publish overhead, DREAM [7] proposes a publish scheme in which frequency of updates is reduced as the distance from the updating node increases. While this intuition is the starting point for our scheme as well, location updates made to all nodes as in DREAM are not scalable. MLS[8] proposes a “lazy” hierarchical position lookup service in which updates are published to certain fixed geographical regions. The routing algorithm 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. MLS builds on an earlier scheme termed LLS[9] and improves the stretch of the routing protocol to a constant stretch. The key difference of [8], [9] from our work is they do not relate the publish overhead due to persistent mobility to network transport capacity, and also do not provide means to vary the tradeoff between route stretch and publish overhead.

II. SYSTEM MODEL AND BASIC COMPUTATIONS

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 n/Λ grows with n .

Connectivity: We assume that the communication radius for all nodes is fixed at $r(n)$, satisfying $\Lambda\pi r^2(n) = (1 + \epsilon) \log n$ [3] to ensure connectivity. We will denote the communication radius $r(n)$ by r henceforth. We assume $\epsilon > \epsilon_0$ (with $\epsilon_0 \approx 1.6$), so that greedy geographic routing works with probability 1 [10].

Scalability: Using the protocol model in [2], the maximum number of simultaneous transmissions available per time slot, denoted by $T_A(n)$, scales as $\Theta\left(\frac{Wn}{\Lambda r^2}\right)$, where W denotes the communication bandwidth. Denote 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. It is also useful to define the average number of transmissions on a per node basis as $t_U(n) = T_U(n)/n$ and $t_A(n) = T_A(n)/n$, respectively. We define a protocol to be scalable if $T_U(n) = O(T_A(n))$, or equivalently, $t_U(n) = O(t_A(n))$.

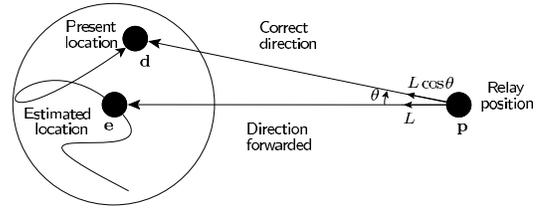


Fig. 2. Rate of progress depends on the angle between the estimated and correct directions.

This 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 the protocol we propose under our mobility model, which is described next.

Mobility Model: Every node in the network is mobile, executing Brownian motion (MS velocity $2\sigma^2$) with reflection at the boundaries of the deployment region (assumed to be square for convenience). This model is well-suited for investigating scalability, which depends on how distant nodes perceive the mobility of a destination node, since a large class of randomized models for *local* mobility “look like” Brownian motion when viewed from *far away* and at *large time scales*.

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 Fig. 2, where \mathbf{p} denotes the packet’s current location, \mathbf{d} denotes the destination’s location, and \mathbf{e} the currently available estimate of the destination’s location. Denoting the *relative uncertainty* in location as $U = \|\mathbf{d} - \mathbf{e}\| / \|\mathbf{p} - \mathbf{e}\|$, we note that, for a fixed relative uncertainty, the localization error can be larger when the packet is further away from the destination. Further, bounding the relative uncertainty leads to bounded route stretch: if the relative uncertainty is always less than $U_0 < 1$, then the worst-case angle between the correct and estimated directions always satisfies $\sin \theta \leq U_0$. This implies that $\cos \theta \geq \sqrt{1 - U_0^2}$, and that the route stretch is at most $1/\sqrt{1 - U_0^2}$. 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. For fixed relative uncertainty, the localization error can be larger when the packet is further away from a destination, which means we can reduce the update frequency to far-away nodes. However, we observe next that such a reduction, by itself, is not enough to provide scalability.

B. Update Costs

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 of area $|A|$ is $\Theta(|A|/r^2)$. Both upper and lower bounds use tiling arguments similar to those in [2], omitted here due to lack of space.

In order to maintain a relative uncertainty of at most U_0 , location updates from a particular node must reach all nodes that are a distance z away from it if it moves a distance $U_0 z$. Estimating the average frequency of these updates, and using the cost of multicast, we can show that the per-node overhead is given by $t_U(n) = \Omega(\sigma^2 \log n / r^2)$. This does not scale as the available number of transmissions per node, scales as $t_A(n) = \Theta(W/(\Lambda r^2))$.

Thus, while the angle between the true and estimated directions towards the destination cannot be too large if we wish to provide guarantees on route stretch, the preceding overhead computations imply that we must appropriately choose a *subset* of nodes to update in order to reduce the routing overhead enough that the protocol can scale.

III. PROPOSED PROTOCOL

Given space limitations, we discuss only the key ingredients of the proposed protocol.

A. Publish 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. 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. Our main focus, therefore, is on how position updates to distant nodes are performed. 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 typically directs its updates to geographic regions structured as annular rings around its current position, indexed as $i = 0, 1, \dots, K$ with increasing radii. The i -th ring has a fixed inner radius of r_i and thickness d_i . So the geographical region covered by a ring is specified by its center \mathbf{c} and its index i . Updates published to the i -th ring have a lifetime, which is at most T_i , after which they are discarded by the relay nodes which received them. We refer to this as the *temporal validity* of an update. Updates are also constrained in space and can only be used by relay nodes when they are in the ring to which the update was made. We refer to this as the *spatial validity* of an update.

Let $\mathbf{d}(t)$ denote the position of the destination node at time t . There are two kinds of location updates: normal and abnormal updates. While the two kinds of updates use the location of the destination $\mathbf{d}(t_0)$ at the time of dispatch t_0 as their position estimate, they differ in two aspects: (i) normal updates are centered around $\mathbf{c} = \mathbf{d}(t_0)$, while abnormal updates are not (ii) normal updates have lifetimes of T_i while abnormal updates have lifetimes shorter than T_i .

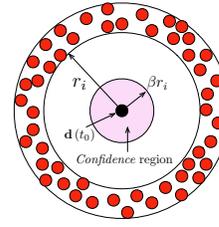


Fig. 3. Update made to ring of index i at time t_0 by the destination node (in black) with lifetime T_i . The red relays receive this update. The region marked in pink is the *confidence* region of this update.

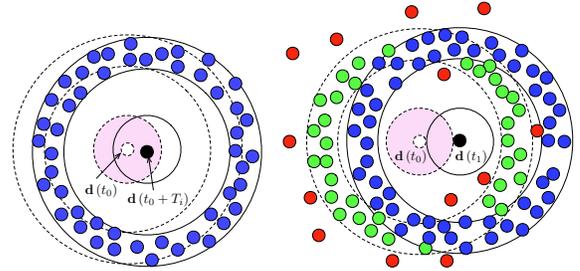


Fig. 4. Left (a): Typical scenario of the destination node staying within the *confidence* region of the previous update, when it expires at $t_0 + T_i$. A new *normal* update of lifetime T_i is made to the ring i which is received by the blue relay nodes. Right (b): An unlikely situation where at time $t_1 < t_0 + T_i$, the destination node leaves the *confidence* region of the previous update, thus requiring an *abnormal* update (received by the green relays) of lifetime $T_i - (t_1 - t_0)$ and a *normal* update of lifetime T_i (received by the blue relays). Relays marked red, *outside* the two update rings possess stale unexpired updates made at time t_0 and these updates can be applied to packets only if these relays re-enter the ring centered at $\mathbf{d}(t_0)$

Types of updates and their triggers: We will explain this via a running example: We focus on a specific ring index i , with the understanding that updates occur in parallel for all ring indices. Suppose that at time t_0 , the destination decides to send a normal update to ring index i as shown in Fig. 3. It updates the relay nodes (marked red) in the ring centered around its current location $\mathbf{d}(t_0)$. The relay nodes in this ring expect that in the lifetime of the update (up to time $t_0 + T_i$), the destination will remain within a circle of radius βr_i around the current location (marked pink), termed *confidence region* for the update. This is ensured by the appropriate choice of T_i . So, typically, the destination will remain in the confidence region of an update and hence at time $t_0 + T_i$, the update will expire and a new normal update will be published to ring i with center $\mathbf{d}(t_0 + T_i)$ as shown in Fig. 4(a). The recipients of the new update are marked blue. However, it is possible that the destination crosses the boundary of the confidence region at time $t_1 < t_0 + T_i$ as shown in Fig. 4(b). In this case, the destination sends an abnormal update to nodes (marked green) in the ring centered at the *prior* update (also referred to as a stale update). The lifetime $T_i - (t_1 - t_0)$ of the abnormal update is chosen to span the remaining lifetime of the stale update. The purpose of the abnormal update is to prevent misrouting due to stale updates. However note that there can be stale inactive (due to spatial validity) updates elsewhere in the network (nodes marked in red). At time t_1 , the destination

also sends a normal update to the ring i (marked blue). This is to ensure that each ring index i always has one normal update, whose confidence region guarantee is met. This enables us to give route stretch bounds. Certain relay nodes may receive both normal and abnormal position updates of index i , which are made at the same time (nodes in Fig. 4(b) which at time t_1 lie in the intersection of the i -th rings centered at $\mathbf{d}(t_0)$ and $\mathbf{d}(t_1)$), in which case the recipients preferentially keep the normal update as its lifetime is longer. 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.

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.

Spatial validity of updates: 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, when a node moves out of this ring, it can not use the information it received about the destination's location. However, if it returns to the update ring, it will use this information. We place such a restriction on the spatial validity of updates to prevent stale updates from influencing packet trajectories. Since updates have spatial validity, we are able to use abnormal updates to "overwrite" misleading information about the destination's location. When abnormal updates are used to compensate for stale updates which have spatial validities, it can be shown that our analysis for route reliability in Section IV-B can be extended to provide bounds on probability of misrouting due to stale updates.

Update propagation: In order to limit the traffic generated by a location update, the destination node sends the update packet in a specified direction \mathbf{u} until it hits the ring it is intended for, at which point it "expands" into a multicast message. Each intermediate node 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 scales 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 III-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 these fields for geographic forwarding, and are allowed to overwrite them if they have a "better" active (spatially and in time) estimate of the destination's location. An estimate is considered "better" only if its ring index is smaller than that of the packet. If the ring indices of the packet and the estimate are the same, this means that the destination node had moved beyond the confidence region of an update and an abnormal update was issued to the same

region to avoid misrouting. So in case of such a tie in the packet index, the estimate at the relay is considered better than the one available to the packet only if it is more recent.

Initialization: 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).

IV. SCALABILITY, RELIABILITY AND STRETCH

We now analyze the proposed protocol to provide design guidelines on choosing protocol parameters so as to achieve scalability, reliable routing, and guarantees on route stretch. Four parameters, α , β , γ and μ , are involved. 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: $T_i = T_0 \alpha^{\gamma i}$. The confidence region for an update to a ring of index i is given by βr_i .

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 K scales with n , note that we need $\pi r_K^2 = \pi r_0^2 \alpha^{2K}$ roughly equal to the network area n/Λ , which yields $K = \Theta(\log n / (\Lambda r_0^2)) = \Theta(\log n)$.

A. Publish Overhead

For 2D Brownian motion, it can be shown (details omitted) that 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 $\gamma < 2$. This implies that the rate of abnormal updates tends to zero, and that the update rate U_i for ring i is approximately $1/T_i$ for i large.

As described in Section III-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 i th ring ($i \geq 1$) is $A_i = \pi((r_i + d_i)^2 - r_i^2) = \pi d_i^2 + 2\pi r_i d_i = \Theta(r_i d_i)$ (since radius r_i scales faster than 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, the number of transmissions to multicast in this area is $C(A_i) = \Theta(|A_i|/r^2)$. Proceeding along the straight line takes $\Theta(r_i/r)$ transmissions, which can be ignored in comparison to the preceding. Thus, the number of transmissions for an update to the i th ring is $\nu_i = \Theta(r_i d_i / r^2)$.

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 converges if and only if $\alpha^{1+\mu-\gamma} < 1$ (since $\alpha > 1$ in order to exponentially expand the rings, $1 + \mu - \gamma < 0$ is a necessary condition). If this condition is satisfied, we can show that

$$t_U = \Theta(\sigma^2/r^2) = \Theta(\sigma^2/\log n),$$

which matches the throughput available per node t_A for a fixed bandwidth. Notice that $t_U/t_A = \Theta(\Lambda\sigma^2/W)$.

B. Routing Reliability

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 i -th ring. The lifetime T_i of normal updates must be short enough that the probability of a miss tends to zero. 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 has a smaller probability of meeting a node with an active update. Using this, and accounting for the "dissipation" of updates due to node motion, we can show that the miss probability for the i th ring satisfies

$$\log P_{miss}(i) \leq -\frac{d_i^2}{\sigma r \sqrt{2\pi T_i}}, \quad (1)$$

under the following asymptotics: $r_i/\sqrt{T_i} \rightarrow \infty$, $d_i/r_i \rightarrow 0$ and $d_i/\sqrt{T_i} \rightarrow 0$. We want the right-hand side of (1) to go to $-\infty$ to drive the probability of miss to zero, which requires that $d_i^2/\sqrt{T_i} \rightarrow \infty$, corresponding to the condition $\gamma < 4\mu$.

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 in terms of α and β (formula omitted).

D. Protocol parameter choices

The protocol parameters are $\alpha > 1$, $0 < \beta < 1$, $\gamma > 0$ and $0 < \mu < 1$. We have 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 $\alpha\beta < 1 - \beta$.

V. 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. However, we have developed the protocol in sufficient detail (although we skip most of the detailed description here) that the approach can be verified by computer 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.

We have had to omit many technical details due to space considerations, and refer to a technical report posted online for further details [11].

REFERENCES

- [1] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *MobiCom '00*.
- [2] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, 2000.
- [3] —, "Critical power for asymptotic connectivity," in *IEEE CDC '98*.
- [4] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, 2002.
- [5] M. Grossglauser and M. Vetterli, "Locating mobile nodes with EASE: learning efficient routes from encounter histories alone," *IEEE/ACM Trans. Netw.*, 2006.
- [6] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age matters: efficient route discovery in mobile ad hoc networks using encounter ages," in *MobiHoc '03*.
- [7] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *MobiCom '98*.
- [8] R. Flury and R. Wattenhofer, "MLS: an efficient location service for mobile ad hoc networks," in *MobiHoc '06*.
- [9] 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*.
- [10] 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 *MobiHoc '06*.
- [11] D. Ramasamy and U. Madhoo, "Geographic Routing in Large-Scale MANETs," University of California, Santa Barbara, Tech. Rep., January 2012. [Online]. Available: <http://www.ece.ucsb.edu/wcsl/Publications/scalableGeoRoutingTR.pdf>