

Locating Nodes with EASE: Last Encounter Routing in Ad Hoc Networks through Mobility Diffusion

Matthias Grossglauser
AT&T Labs- Research
180 Park Avenue
Florham Park NJ 07932
USA
mgross@research.att.com

Martin Vetterli
LCAV
EPFL
1015 Lausanne
Switzerland
martin.vetterli@epfl.ch

Abstract—Routing in large-scale mobile ad hoc networks is challenging because all the nodes are potentially moving. Geographic routing can partially alleviate this problem, as nodes can make local routing decisions based solely on the destinations' geographic coordinates. However, geographic routing still requires an efficient *location service*, i.e., a distributed database recording the location of every destination node. Devising efficient, scalable, and robust location services has received considerable attention in recent years.

The main purpose of this paper is to show that *node mobility* can be exploited to disseminate destination location information *without incurring any communication overhead*. We achieve this by letting each node maintain a local database of the time and location of its last encounter with every other node in the network. This database is consulted by packets to obtain estimates of their destination's current location. As a packet travels towards its destination, it is able to successively refine an estimate of the destination's precise location, because node mobility has "diffused" estimates of that location.

We define and analyze a very simple algorithm called EASE (Exponential Age SEarch) and show that in a model where N nodes perform independent random walks on a square lattice, the length of the routes computed by EASE are on the same order as the distance between the source and destination *even for very large N* . Therefore, without exchanging any explicit location information, the length of EASE routes are within a constant factor of routes obtained with perfect information. We discuss refinements of the EASE algorithm and evaluate it through extensive simulations. We discuss general conditions such that the mobility diffusion effect leads to efficient routes without an explicit location service. In practical settings, where these conditions may not always be met, we believe that the mobility diffusion effect can complement existing location services and enhance their robustness and scalability.

I. INTRODUCTION

In large-scale ad hoc networks, some or all the nodes may be moving. Therefore, the network topology changes with time. Routing algorithms have to base routing decisions on at least a partial knowledge of the network topology. The collection and exchange of topology information (e.g., distance vectors or link states) consumes valuable bandwidth and energy. A

variety of routing algorithms have been developed that trade off the quality of routes, their computing and transmission overhead, and the degree of permissible mobility [11].

Position-based (or geographic) routing exploits the fact that nodes usually live in the plane. This enables nodes to make local routing decisions based solely on the destinations' geographic coordinates [8], [2], [3], [7] (see [10] for an excellent review of position based routing). For this purpose, a mobile ad hoc network is regarded as a set of nodes in the plane, with an associated mobility process. Connectivity is achieved through wireless links, and is thus essentially local (see [5] for an example of the model). While the set of nodes and their connectivity defines a graph, this graph is not arbitrary but instead closely related to the geometry of the plane. In general, a node only needs to know its own location and that of its neighbors to make a routing decision towards any destination node with a known location.

However, geographic routing still requires an efficient *location service*, i.e., a distributed database recording the location of every destination node. Devising efficient, scalable, and robust location services has received considerable attention in recent years [10], [9], [6], [14]. Interestingly, location and routing have been mostly considered in isolation so far: a source first looks up the current position of the destination through the location service, and then routes a packet towards that position using a geographic routing algorithm. This requires that the location service has to be able to track all the nodes in the network, and maintain a distributed database recording the locations of these nodes. Every change in topology has to be reflected in this distributed database, which inevitably involves some exchange of location information between nodes, and hence incurs a transmission cost. This transmission cost to maintain location state therefore depends directly on the amount of mobility, or the rate at which the network topology changes.

An elegant way of reducing this cost is by exploiting the *distance effect* [2], which is basically the observation that

the precision with which the position of a destination has to be known to make a good, but sometimes suboptimal, local routing decision at a node, depends on the distance of that node from the destination. If the node is far away from the destination, an imprecise estimate is sufficient, and vice versa. Routing schemes such as DREAM [2] exploit this effect to develop more “lazy” approaches to maintaining location information about all the nodes in the network. This approach essentially amounts to trading off a smaller location maintenance overhead, which is incurred continually with every topology change, for a slightly larger routing cost, as routes are in general suboptimal.

In this paper, we go a step further and try to completely eliminate the cost to update location state. If nodes are not allowed to exchange any explicit location updates, then the only local information available to a node about the network topology is the history of other nodes it has encountered in the past, i.e., that it has been directly connected to. More specifically, we assume that every node remembers the time and location of its last encounter with every other node (i.e., when these two nodes last were directly connected neighbors; cf. Fig. 1). We call a routing algorithm a *last encounter routing* (LER) algorithm if at every node along a packet’s route, the next hop decision depends only on (a) the time and location of that node’s last encounter with the destination, and (b) auxiliary information carried by that packet. The main question we ask in this paper is the following: if all the nodes in the network are moving, is it possible for LER schemes to compute efficient routes, despite the absence of a location service? We show that, depending on the mobility processes, this is indeed possible. This is quite remarkable, given that LE routing invests no network capacity to track nodes, i.e., to maintain distributed location information.

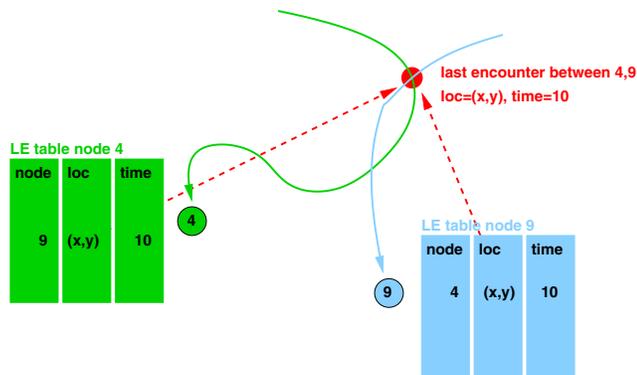


Fig. 1. A *last encounter table* in every node remembers the location and time of the last encounter with every other node in the network. In last encounter routing (LER), this table is queried by a packet to improve, if possible, its estimate of the location of its destination node.

The insight at the root of our investigation is the following. On the one hand, mobility of the nodes creates uncertainty about their location. On the other hand, consider some node d that is the destination of a packet. Some other node i that has encountered d in the past remembers the location of that last

encounter. Three observations explain why LE routing can give rise to efficient routes: (a) the location of the last encounter is still a reasonably good estimate of the destination’s location after some time; (b) the time of that encounter, or equivalently, the “age” of the estimator, is a measure for the precision of that estimate; (c) node i ’s own mobility means that a recent estimate of d ’s position is available at some distance from d ; given that d encounters other nodes all the time due to mobility, this essentially leads to a diffusion effect of noisy position estimates around d . The locality in the mobility processes inherently leads to a distance effect, in that better position estimates for d become available as a packet approaches d ’s current position.

Clearly, the feasibility of LER schemes will depend on the mobility process. If at any point in time, a node can jump uniformly over the entire surface of interest, an estimate based on the previous location is of no help. However, in the more likely scenario where the process has some locality, such as a random walk, then aged location information is useful, and diffuses at the same speed as the node moves itself. If the density of neighbors is sufficient both along the path of the destination node (so as to diffuse sufficiently) and along the path of a packet moving towards the destination (to get enough new estimates), then LE routing can work well.

The outline of the paper is as follows. The next section discusses related work, while Section III describes the model we are considering in more detail, in particular the topology, the mobility model and the performance criterion. In Section IV, we examine the asymptotic performance of LE routing when the network size grows large, using standard results from the theory of random walks. We look at a very simple LER algorithm called *exponential age search (EASE)*, and we show that the expected path length obtained with EASE is of the same order as the optimal path length even in very large networks. Section V gives an improved version of EASE called GREASE, which improves the estimate of the destination location more aggressively as the packet moves along. Section VI gives simulation results in fairly large networks (1000 nodes); they confirm the good performance and scalability of EASE and GREASE. Finally, Section VII provides some further insights and discussions.

II. RELATED WORK

Several position-based routing algorithms have been proposed in the literature [8], [2], [3], [7]. The principal goal of these algorithms is to ensure that a short route can be found if one exists between a source and a destination whose locations are known. This is not trivial, because forwarding greedily in the direction of a destination is not guaranteed to work, as there is no guarantee that a node always has a neighbor closer to the destination than itself.

Mobility management is a basic problem in standard mobile networks (see for example [14]). In ad hoc networks, the situation is complicated by the absence of centralized servers

(home location registers). In an ad hoc network endowed with position-based routing, mobility management amounts to tracking the location of every potential destination through a *location service*. This location service has to be itself distributed across the ad hoc network, and this can be achieved with various methods that trade-off complexity, overhead, and robustness [10], [9], [4], [6], [14]. What is common to all these location services is that they incur overhead by explicitly exchanging location information between nodes, either to update location information in the distributed database, to request the location of a destination node, or both.

Approximate location services have been proposed in various forms. For example, in the grid location service (GLS) [9], a quadtree based location service creates a hierarchy of square regions. Updates of a node's position are sent to a decreasing number of nodes acting as location servers as the distance increases. In the geographical region summary service (GRSS) [6], a similar grid location service is proposed, with increased efficiency due to forwarding location aggregation. For a more complete overview of location services and their use in position-based routing, we refer to [10] and references therein.

Our work is closest in spirit to the DREAM algorithm [2]. In DREAM, every node maintains a position table for every other node. DREAM consists of a position update algorithm and a routing algorithm. The position update algorithm ensures that all the nodes in the network have a sufficiently accurate estimate of a reference node's position. This is achieved by the reference node flooding a limited region around itself to install new position estimates in the nodes in this region. The scope of this flooding depends on the distance that the node has traveled since the last such flooding. The routing algorithm ensures that a packet can reach a destination efficiently and with high probability, using restricted directional flooding based on the region where the destination is expected to be located. The approximate location is given by a circle around the last known location of the destination, and the radius of this circle is given by $(t_1 - t_0) \cdot v_{max}$, where t_1 is the current time, t_0 the time when the location was registered, and v_{max} the maximum speed.

III. MODEL

We now describe the model used in the paper for analysis. Although in reality, node positions are continuous processes in continuous time, it is convenient for the analysis to use discrete approximations. We therefore consider node mobility processes on a square grid in slotted time. Our simulations show that our main result on the efficiency of mobility diffusion routing carries over to continuous settings with less regular topologies (cf. Section VI).

Topology. For the sake of analysis, we make a discrete approximation of the continuous geometry of a region in R^2 . That is, nodes live on vertices of the $M \times M$ square lattice. More precisely, the topology is a two-dimensional square grid

of M^2 vertices (x^1, x^2) , $x^{1,2} \in \{1, \dots, M\}$. The distance $|x_2 - x_1| = |x_2^1 - x_1^1| + |x_2^2 - x_1^2|$, denotes the Manhattan distance between two vertices.

There are $N = \lambda M^2$ mobile nodes that move on this grid, where the node density λ remains fixed. Let $X_i(t) = (X_i^1(t), X_i^2(t))$ denote the vertex where node i is located at discrete time t . We assume that each node always knows its current position on the grid and the identity of its neighbors. The neighbors of a node at a given time are the set of other nodes at the same vertex, as well as all the nodes reachable through horizontal and vertical steps, where we are allowed to "step over" empty vertices. In other words, a node is connected to another node through a wireless link if there exists a path in the underlying lattice connecting the respective vertices, such that all the internal vertices of the path are empty.¹

This definition is convenient, as the topology is a connected undirected graph at every time t . Sending a packet from a source to a destination that are at Manhattan distance d of each other requires at most d transmissions (hops) in this topology.

Routing. Nodes wish to send data packets to each other. A node may forward a packet only to its neighbors. Therefore, it is necessary for nodes to serve as relays on behalf of packets of other nodes. In the analysis, we will focus w.l.g. on a single destination node with index 1, with the nodes $i = 2, \dots, N$ serving as potential relays for packets destined for node 1.

With each node $i \in \{2, \dots, N\}$ we associate an *age* variable $T_i(t)$ that records how long ago node i was last a neighbor of the destination node. More formally,

$$T_i(t) = t - \max_{\tau \leq t} \{ \tau : |X_i(\tau) - X_1(\tau)| \leq 1 \}. \quad (1)$$

Let $V(i, t)$ be the neighbors of node i at time t . A *last-encounter routing scheme* is a function that computes the next hop neighbor $j \in V(i, t)$ for a packet k that has arrived at node i , i.e., $j = f(X_i(t), T_i(t), W_k)$, where W_k is some auxiliary information carried by the packet. This auxiliary information can depend only on the set of nodes traversed so far by the packet.

Mobility. The position $X_i(t)$ of node i at discrete time t is an independent random process with uniform stationary distribution over all nodes. More specifically, we assume that $X_i(t)$ is a two-dimensional random walk on the square lattice, i.e.,

$$X_i(t+1) = X_i(t) + \Delta_i(t), \quad (2)$$

where the $\Delta_i(t)$ are i.i.d [12]. Also, both components of $\Delta_i(t)$ are i.i.d., with zero mean and finite variance σ^2 . We assume that the random walks reflect at the boundary of the square lattice.

Time scales. In our analysis, we assume the realistic scenario of nodes moving at "human" speeds, while packets move at

¹Note that λ should be large enough to ensure that empty regions remain small.

light speed. Thus, while topology changes occur at time-scales of minutes or longer, packets can be expected to spend at most tens of milliseconds (due to queueing and propagation delay) in the network. This allows to decouple the time scales, such that for the purpose of routing a packet, the nodes are frozen for the time of the routing to conclude.

Cost metric. Let $C(s, d, t)$ denote the total number of transmissions (or hops) necessary to transmit a packet from source s to destination d at time t . This cost will include both transmissions of the actual packet from a sender to a receiver node to make progress towards its destination, as well as transmissions necessary for a “search” packet to collect information from surrounding nodes to make the next routing decision.

We now define the *Exponential Age Search* (EASE) routing algorithm. EASE computes routes purely based on last encounters, which means that no transmission capacity is sacrificed to explicitly diffuse location information or to maintain a view of the current topology. The goal of EASE is to be so simple as to be amenable to analysis and to provide insight into the conditions under which mobility diffusion provides good routes. In Section V, we will discuss a slight modification of EASE that improves performance, but is less amenable to analysis.

In the description of the algorithm, we assume w.l.g. that $t = 0$.

Algorithm 1: The EASE algorithm

- 1 Set $T_0 := T_s(0)$, $Y_0 := X_s(0)$, $k := 0$.
- 2 Repeat
- 3 Search the nodes around Y_k in order of increasing distance until a node i is found such that $T_i(0) \leq T_k/2$.
- 4 Let $T_{k+1} = T_i(0)$, and $Y_{k+1} := X_1(-T_{k+1})$ be the new anchor point.
- 5 While not at Y_{k+1}
- 6 Route packet: find next hop j towards Y_{k+1} and forward packet to j .
- 7 End while
- 8 $k + +$.
- 9 Until $Y_k = X_1(0)$.

Initially, the packet is at its source at position $X_s(0)$. In steady state, the destination node has visited this position with probability 1 at some time $-T_0$ in the past. The basic idea behind EASE is for a route to follow the trajectory $X_1(\cdot)$ of the destination node between $t = -T_0$ and $t = 0$ in “jumps” of decreasing length, until the packet arrives at the current position $X_1(0)$ of the destination node. We call the end-points of such jumps *anchor points*. We do not prescribe a particular routing algorithm for the packet to get from one anchor point to the next; any position-based routing algorithm could be used for this purpose (cf. Section II). Note that according to our definition, EASE is a last encounter routing (LER)

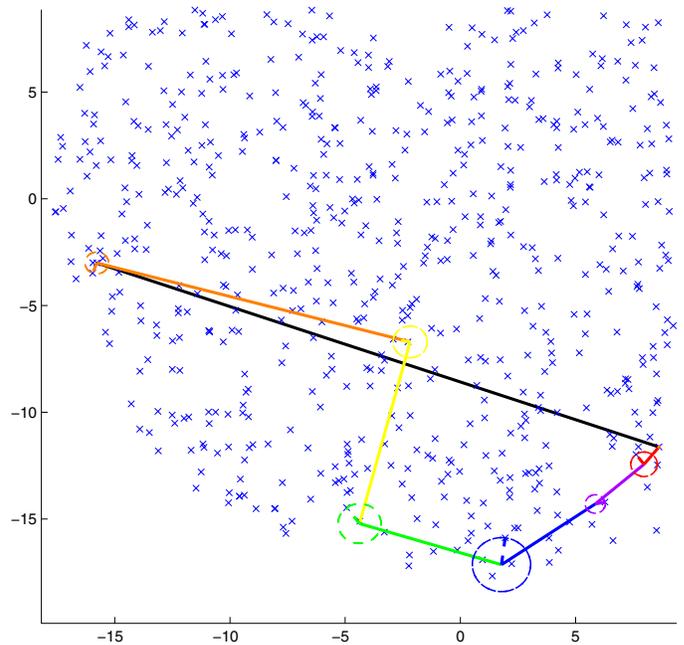


Fig. 2. A sequence of anchor points computed by the EASE algorithm; the search disks S_k are also shown. A node in each search disk around anchor point Y_k provides the next anchor point Y_{k+1} , which corresponds to the point on the trajectory of the destination of at most half the age of the current anchor point.

algorithm, where the auxiliary information carried with the packet consists of the age and location of the last encounter with the destination by a node in the vicinity of the previous anchor point.

IV. ASYMPTOTIC PERFORMANCE OF EASE

In this section, we analyze the asymptotic performance of EASE when the network size N becomes large. Recall that EASE incurs no a-priori overhead to track topology changes, unlike traditional location services. The main question we need to answer is then how large the penalty will be when we route a packet from a source to a destination using only last-encounter information. We show that under the topology and mobility model described in the previous section, the mean route length between a source and a destination is on the same order as shortest routes, even for very large networks. Our main result is as follows.

Claim: For two arbitrary nodes s and d , the route from s to d calculated by the EASE algorithm satisfies

$$E[C(s, d, t)] = O(\sqrt{N}). \tag{3}$$

Note that the expected distance between a randomly selected node pair is also on the order of \sqrt{N} . The result therefore implies that mobility diffusion-based routing is asymptotically efficient, in that routes obtained through EASE are at most a constant factor longer than the distance between source and destination nodes.

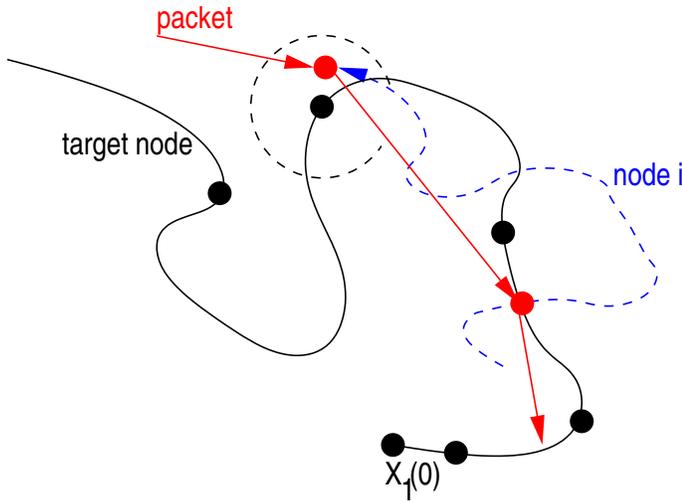


Fig. 3. At every anchor point Y_k of age T_k , the EASE algorithm performs a search until a new anchor point of age at most $T_k/2$ is found.

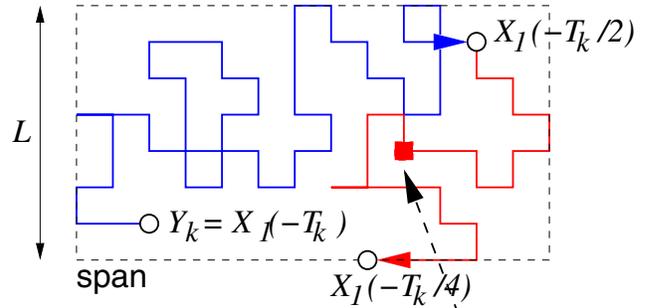
We are unable to provide a full proof here. However, we outline the steps of the proof in the remainder of this section in sufficient detail to give an intuition of why EASE succeeds in computing asymptotically efficient routes, and to crystallize out the salient features of the node mobility processes that permit to this efficiency.

It is useful to make the following definition. A *search box* S of size S centered at node i is a set of S nodes (including i) such that they are contained in the smallest square of grid points centered at $X_i(0)$ that contains S nodes at time 0. Note that the expected side length of a search box of size S is approximately $\sqrt{S/\lambda}$, because there are on average λ nodes per grid position. We denote by S_k the size of the smallest search box centered at anchor point Y_k that contains the node i providing the next anchor point Y_{k+1} . Note that the cost of searching this search box is proportional to S_k^2 .

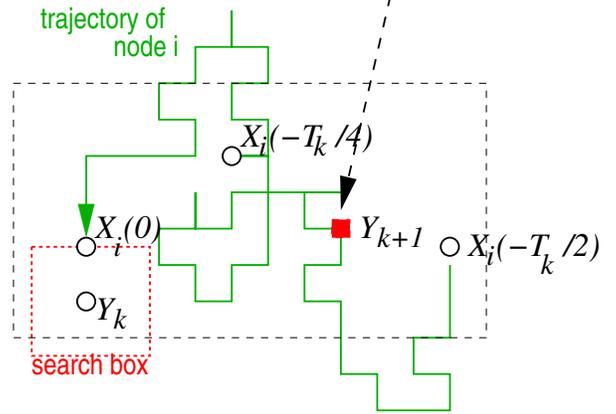
- 1) **Single search step.** Let us consider the k th iteration of EASE, and derive its approximate cost as N grows large. Suppose the packet is at position Y_k , and assume that the age of this estimate is T_k , i.e., $Y_k = X_1(-T_k)$. Now consider the trajectory of the destination node over the interval $[-T_k/2, -T_k/4]$. The goal of the k th iteration of EASE is to find a node i close to Y_k that provides a new anchor point $Y_{k+1} = X_1(-T_{k+1})$, such that $T_{k+1} \leq T_k/2$. We have to determine how many nodes S_k have to be searched on average until a new anchor node is found. We will show that the cost of searching for this new anchor node is small compared to the cost of actually forwarding the packet to Y_{k+1} .
- 2) **Messenger node.** Consider a node i that is a neighbor of the destination node at some time $-t$ between $-T_k/2$

²In practice, this search could be performed, for example, through a TTL-constrained (Time To Live) local flooding, where the TTL is doubled every time the search has been unsuccessful.

(a) span of target random walk



last encounter between nodes 1 and i = new anchor point



(b) search box and messenger node i

Fig. 4. One EASE step: (a) The span encloses the path $X_1(-T_k), \dots, X_1(-T_k/4)$ of the destination node. (b) For a search to be successful, a “messenger” node i encountered by the destination between $-T_k/2$ and $-T_k/4$ must perform a random walk to end up within the search box at time $t = 0$. The worst case occurs if $X_i(-T_k/4)$ and $Y_k = X_1(-T_k)$ are at opposite corners of the span.

and $-T_k/4$. We call such a node a *messenger node*³. The size S_k of the search box will be determined by the locations of the messenger nodes at time 0.

Let us consider the *span* of the trajectory of the destination node, i.e., the smallest box containing $\{X_1(-T_k), \dots, X_1(-T_k/4)\}$ (cf. Fig. 4(a)). This box has a side length on the order of $L = O(\sqrt{T_k})$ [13, Chapter 5].

- 3) **Hitting probability of a single messenger node.** Let us consider the probability that one of the messenger nodes hits a search box of given size S_k . We can lower bound this probability by conservatively assuming that the messenger node starts in one corner of the span at the latest possible time, i.e., at $t = -T_k/4$, and that Y_k is located in the opposite corner of the span (cf. Fig. 4(b)). It is a worst case because one quadrant of the search box is closer to the starting corner than Y_k .

³Note that we are conservative in only considering messenger nodes encountered up to time $-T_k/4$. This assumption ensures that the distance traveled by the destination between $-T_k$ and the time of encounter $t \in [-T_k/2, -T_k/4]$ is comparable to the distance traveled by the messenger node between t and 0.

Let us assume that L and S_k are fixed. We first focus on a single dimension $d = 1, 2$ and determine the probability p that in dimension d , the messenger node i hits the quadrant of the search box overlapping the span. By the central limit theorem, this probability is approximately

$$\begin{aligned}
p &= \Pr \left\{ X_i^d(0) - X_i^d(-T_k/4) \in \left[L - \sqrt{\frac{S_k}{\lambda}}, L \right] \right\} \\
&= \Pr \left\{ \frac{X_i^d(0) - X_i^d(-T_k/4)}{\sigma\sqrt{T_k/4}} \in \left[\frac{L - \sqrt{S_k/\lambda}}{\sigma\sqrt{T_k/4}}, \frac{L}{\sigma\sqrt{T_k/4}} \right] \right\} \\
&= Q \left(\frac{L - \sqrt{S_k/\lambda}}{\sigma\sqrt{T_k/4}} \right) - Q \left(\frac{L}{\sigma\sqrt{T_k/4}} \right) \\
&\approx \frac{16}{\sigma} \sqrt{\frac{S_k}{\lambda T_k}} \phi \left(\frac{2}{\sigma} \right) = c\sqrt{\frac{S_k}{T_k}}, \quad (4)
\end{aligned}$$

where ϕ is the pdf of a standard normal random variable and c is a constant that does not depend on T_k .

Thus, the probability for a single messenger to hit a search box of size S_k is lower bounded by

$$p_{hit} = p^2 = c^2 \frac{S_k}{T_k}. \quad (5)$$

- 4) **Number of messenger nodes.** Now consider the set of all nodes $W = \{i : |X_i(t) - X_1(t)| \leq 1, -T_k/2 \leq t \leq -T_k/4\}$ that are neighbors of the destination node at some point between $-T_k/2$ and $-T_k/4$. The size S_k of the search box required to find the next anchor point is the smallest box centered at Y_k that contains one of the positions $X_i(0), i \in W$. First, we need to compute the size of the set of (distinct) nodes W encountered by the destination. Note that this is different from the number of nodes encountered by the destination, which is of the order of T_k , because it is possible that the destination encounters a node more than once between $-T_k$ and 0. Of course, we should “count” a node i that is encountered multiple times only as one potential messenger node.

We can determine the size of W by noting that the difference between $X_1(t)$ and $X_i(t)$ is also a random walk. The probability that node i and node 1 do not encounter each other again within T_k steps after a first encounter is equal to the probability that the difference random walk $X_1(T_k) - X_i(T_k)$ does not return to the origin within T_k steps. In two dimensions, this probability is $O(1/\log T_k)$ [13, Chapter 4, p. 125]. Therefore, the size of W satisfies

$$\|W\| = \Theta \left(\frac{T_k}{\log T_k} \right). \quad (6)$$

- 5) **Hitting probability for any messenger node.** The key observation to compute the probability that any messenger node hits the search disk of size S_k is that conditional on the destination’s trajectory, the trajectories

of the nodes $i \in W$ after encountering the destination node are independent of each other, and therefore the hit events are independent as well.

The probability that at least one of the nodes in the set W hits the search box satisfies therefore

$$p_{any} = 1 - (1 - p_{hit})^{\|W\|} \approx 1 - \left(1 - \frac{S_k}{T_k} \right)^{T_k/\log T_k} \quad (7)$$

Therefore, S_k is on the order of $\log T_k^4$. As the cost of forwarding the packet at the k th iteration is on the order of $\sqrt{T_k}$, we have

$$\lim_{N \rightarrow \infty} \frac{S_k}{C(Y_k, Y_{k+1}, 0)} = 0, \quad (8)$$

i.e., the cost of searching is insignificant compared to forwarding the packet.

- 6) **Total cost $C(s, d, 0)$.** The total cost of incurred by EASE to route a packet from the source s to the destination d is

$$C(s, d, 0) = \sum_{k=0}^{K-2} C(Y_k, Y_{k+1}, 0) + S_k + C(Y_{K-1}, X_d(0), 0), \quad (9)$$

where K is the number of steps required to reach the destination.

Note that the typical age $T_s(0)$ at the source is $O(N)$, and the first EASE step is therefore of typical length $|Y_0 - Y_1| = O(\sqrt{N})$. As EASE reduces the age T_k of its anchor points by a factor of at least two with every iteration, the sequence of distances between successive anchor points $|Y_k - Y_{k+1}|$ decreases geometrically. Its sum converges and is therefore $O(\sqrt{N})$.

As we have pointed out, this argument is not technically rigorous, but it provides the intuition to explain why the total cost of a route, including both search cost and forwarding cost, does not grow faster on average than the shortest path route cost when the network grows large. One aspect of the above argument that is imprecise is the assumption that the side length L of the span is fixed and of typical length $\sigma\sqrt{T_k}$. In reality, of course, L is a random variable, and we would have to take the expectation of search cost over L . A slight complication then arises for the case when L happens to be atypically large, as the *conditional* probability for a messenger node to hit the search box by traveling far enough is then very small. In this case, the search box is atypically large⁵.

One way to avoid this problem is to add a *cutoff distance criterion* to the simple half-age based criterion in the EASE algorithm. Specifically, if $L \gg \sigma\sqrt{T_k}$, i.e., the destination node has traveled atypically far between time $-T_k$ and time $-T_k/4$, then the probability of finding a messenger node i in an iteration of line 3 of the EASE algorithm is very small. We can then modify the condition to accept a new anchor point as

⁴Because $\lim_{x \rightarrow \infty} (1 - c/x)^x = e^{-c}$.

⁵Unless special precautions are taken to avoid these atypical cases, it can be shown that the expectation of search box size actually does not exist.

follows: find a node i such that either (a) $T_i(0) \leq T_k/2$ (the existing criterion), or (b) $T_i(0) < T_k$ and $|X_1(-T_i(0)) - Y_k| > c\sigma\sqrt{T_k}$, where $c > 0$ is some constant. Condition (b) ensures that in the unlikely event of an atypically large L , the EASE algorithm accepts a new anchor point Y_{k+1} that reduces the age by less than a factor of two, but still makes typical progress towards the destination in terms of distance.

We now discuss some qualitative properties of the node mobility processes that make LER succeed. From the above argument, we can identify two general conditions that have to be satisfied. The first condition concerns the distance traveled by the messenger nodes. For messenger nodes to have a reasonable chance of hitting a given search box, the typical distance traveled by a messenger node between the time of encounter with the destination and time 0 has to be comparable to the distance traveled by the destination between $-T_k$ and the time of encounter. This requires a certain homogeneity in the mobility processes of the nodes. In Section VI, we will examine a case where mobility processes are highly inhomogeneous; we find that it is very difficult to route towards destinations that move much more quickly than most other nodes.

The second condition concerns the density of messenger nodes within the span. In order for the probability to be reasonably high that at least one messenger node hits the search box, there should be a sufficient number of such nodes, i.e., the set W has to be large enough. This is the case if the nodes' mobility processes are such that the total number of grid vertices visited over a time interval t is asymptotically much larger than the distance between the start and end location for that interval. This is because the size of the set W grows essentially with the number of grid vertices visited, while the probability of hitting a node of the search box decreases roughly as the square of the distance (cf. (7)). For random walks, this condition is satisfied, as the size of W grows almost linearly with t , while the distance grows only as \sqrt{t} . In Section VI, we will examine a random waypoint mobility model where this scaling between number of encounters and distance is much less favorable than for a random walk; we find that the cost of routing is considerably higher in this case.

V. GREASE: THE GREEDY EASE ALGORITHM

The EASE algorithm described in the previous section operates in two alternating phases. In the first phase, when a packet has reached an anchor point, it performs a local search around that anchor point to find the next anchor point. In the second phase, an existing position-based routing algorithm is used to route the packet towards the new anchor point. We have made no assumptions about the specific routing algorithm used for this purpose.

The two-phased approach is useful to analyze the performance of EASE and to develop an understanding of why it achieves low cost routes. However, it is clear that EASE ignores a lot of potentially useful information, as it does not consult the local

LE databases of the nodes it traverses in the second phase. Therefore, we propose a modified algorithm called GREASE (GREedy EASE) that checks the age of the last encounter with the destination at each hop. If it encounters a node that has a more recent estimate of the destination's location than the anchor point the packet is currently headed to, then that estimate is assumed to be the new anchor point.

Algorithm 2: The GREASE algorithm

```

1  Set  $T_0 := T_s(0)$ ,  $Y_0 := X_s(0)$ ,  $k := 0$ .
2  Repeat
3    Search the nodes around  $Y_k$  in order of increasing
    distance until a node  $i$  is found such that  $T_i(0) \leq$ 
     $T_k/2$ .
4    Let  $T_{k+1} = T_i(0)$ , and  $Y_{k+1} := X_1(-T_{k+1})$  be the
    new anchor point.
5    While not at  $Y_{k+1}$ 
6      Route packet: find next hop  $j$  towards  $Y_{k+1}$  and
      forward packet to  $j$ .
7      If  $T_j(0) \leq T_{k+1}$ , then  $T_{k+1} := T_j(0)$ ,  $Y_{k+1} :=$ 
       $X_1(-T_{k+1})$ .
8    End while
9     $k++$ .
10 Until  $Y_k = X_1(0)$ .
```

Note that it is entirely possible, and actually a frequent occurrence, that GREASE finds the destination without leaving the inner loop, if the packet always finds a more recent location estimate for the destination before it reaches the current estimate (anchor point). In this case, a search is performed only once around the source. An example of this case is provided in Figure 5.

VI. SIMULATION RESULTS

We have performed a range of simulations in order to evaluate the quality of routes computed by EASE and GREASE. In these simulations, we are interested in the relative cost of routes followed by EASE and GREASE as compared to the shortest path between the source and destination. We know from Section IV that for random walks, they are of the same order, but we do not know the constants involved, and we do not know what performance to expect for other classes of mobility processes. Thus, we investigate various scenarios with different mobility processes (small versus large variances, homogeneous versus heterogeneous traffic, various single step distributions including heavy tailed ones, random waypoint mobility).

Nodes are constrained to move in a disk of surface area N , so that the average node density is 1. The location $X_i(t)$ of node i at discrete time t is an independent constrained random walk, where the disk boundary is reflecting⁶. We look at Gaussian

⁶Note that we go back to the Euclidean domain.

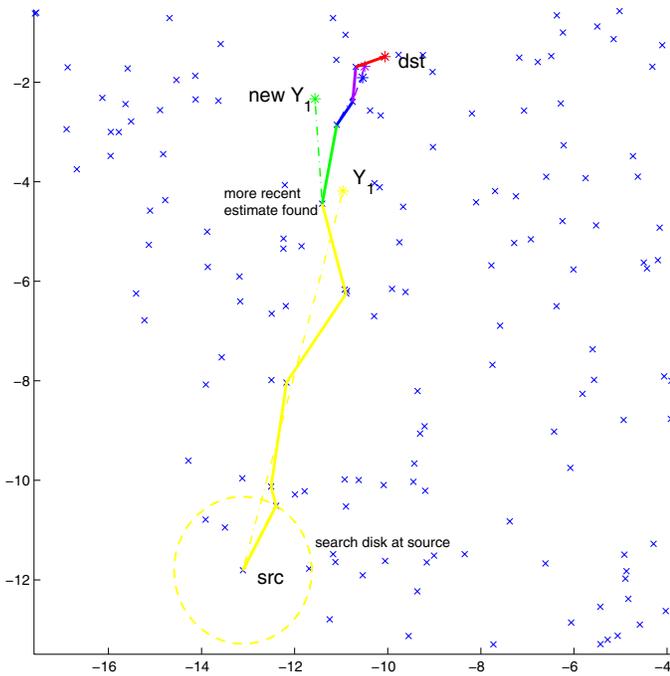


Fig. 5. A sample GREASE route. In this case, the greedy local improvement of the packet's anchor point carries the packet all the way to the destination, without any further searches after the initial search around the source.

and heavy tailed single-step distributions. We also consider a random waypoint model, which has frequently been used in simulation studies of ad hoc networks, but which is not a random walk.

The node positions are initially uniformly distributed over the disk. We then run the random walks for a sufficient warm-up period so that a fair proportion of node pairs have met at least once. Note that the spread of a single random walk is $O(\sigma \cdot \sqrt{T_{wu}})$ after a warm up time T_{wu} . Recalling that the size of the domain is $O(\sqrt{N})$, the warm up time must be of order $O(N/\sigma^2)$. In most simulations, we used a warmup period of 10'000 iterations, except for the heterogeneous case, where we used 40'000 iterations. This ensures that in all the simulations, the fraction of node pairs that have met is at least 30%. Note that the performance of EASE and GREASE is obviously monotonically improving with the warmup time, as the fraction of node pairs that have had encounters increases.

At every time t , we assume that connectivity is given by the Delaunay graph generated by the set of points $\{X_i(t)\}$. This is equivalent to generating the Voronoi tessellation of the set of points $\{X_i(t)\}$, such that every node $X_i(t)$ is the center of a Voronoi cell, and is connected to the center nodes of its adjacent cells. Each node updates the entries in its LE database for its directly connected neighbors.

The advantage of this topology over other topologies we could have adopted (e.g., k nearest neighbors) is that we are guaranteed that a node always has a neighbor that is closer to a destination (except when that destination is already in the first node's Voronoi cell). Therefore, a packet can always

make progress towards its anchor point, and we do not have to deal with backtracking, avoiding routing loops, etc. This allows us to focus on the main issue at hand, i.e., the quality of computed routes based on diffused information about last encounters.

The main metric we evaluate is the relative cost of EASE/GREASE routes compared with the cost of the shortest path route. The relative cost metric therefore captures the relative penalty incurred for not having the exact position of the destination available. More specifically, we generate a set A of random source-destination pairs (s, d) . The empirical relative cost $\hat{\beta}$ is then given by

$$\hat{\beta} := \frac{\pi}{4|A|} \sum_{(s,d) \in A} \frac{\hat{C}(s,d,0)}{|X_s(0) - X_d(0)|}, \quad (10)$$

where A is a random set of source-destination pairs, and where $\hat{C}(s, d, 0)$ is the empirical cost, i.e., total number of transmissions for both forwarding and searching, to get a packet from source s to destination d at time 0. The factor $\pi/4$ stems from the fact that the expected length of the shortest path between two nodes s and d in the Poisson-Delaunay graph is equal to $4/\pi|X_s(0) - X_d(0)|$ [1].

In the simulation results below, we actually give the relative cost conditional on the distance between the source and destination. This provides an indication on whether the relative quality of EASE/GREASE routes increases or decreases as routes get longer.

Gaussian increments, homogeneous mobility. First, consider a homogeneous population with i.i.d. Gaussian position increments of variance σ^2 . As σ increases, we expect the entire process to become noisier, resulting in less efficient routes. We therefore expect the constants involved to be dependent on σ . This is verified empirically in Figure 6, which shows the empirical distribution of relative cost for EASE and GREASE for $\sigma = 0.3$ and $\sigma = 1.0$. As can be seen, approx. 90% of the routes are less than 3 to 8 times longer than optimal, depending on σ and the chosen algorithm. Note that GREASE outperforms EASE by a factor of 2 or more in all cases of interest.

Figure 7 shows the relative cost, conditioned on the source-destination distance being smaller than d , i.e., $|X_s(0) - X_d(0)| \leq d$. That is, we look at increasingly large attempted source-destination distances, and see how the ratio between found and shortest routes evolves. Very interestingly, the ratio decreases monotonically and stabilizes at some rather small value, especially for GREASE. While these simulations do not go beyond 1000 nodes, they seem to indicate good scaling properties, since restricting the distance to a certain d approximates a network with d^2 nodes. At very small d , it seems some small scale discretization effect hurts the behavior (going in the wrong direction in a very small network quickly decreases the performance). Between $d = 10$ and $d = 32$ (corresponding to a network of a 100 to a 1000 nodes, respectively) the performance of GREASE at low σ seems to

have stabilized, with routes about 1.7 times as long as the minimal length.

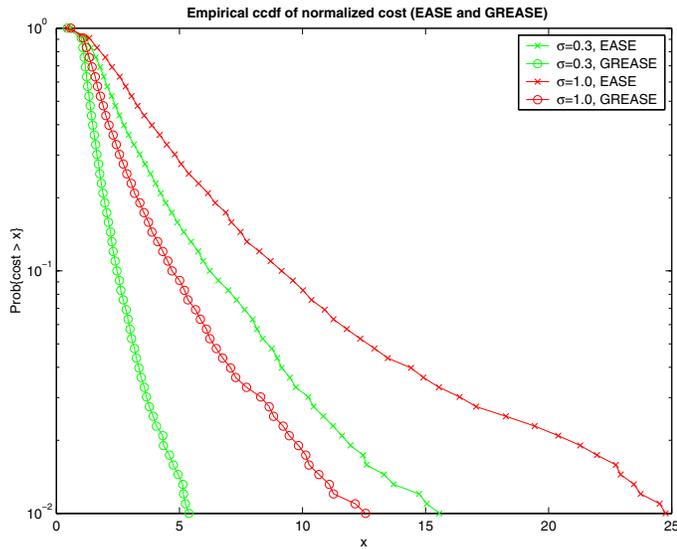


Fig. 6. The empirical complementary CDF (CCDF) of relative cost for both EASE and GREASE and various values for the single-step standard deviation σ .

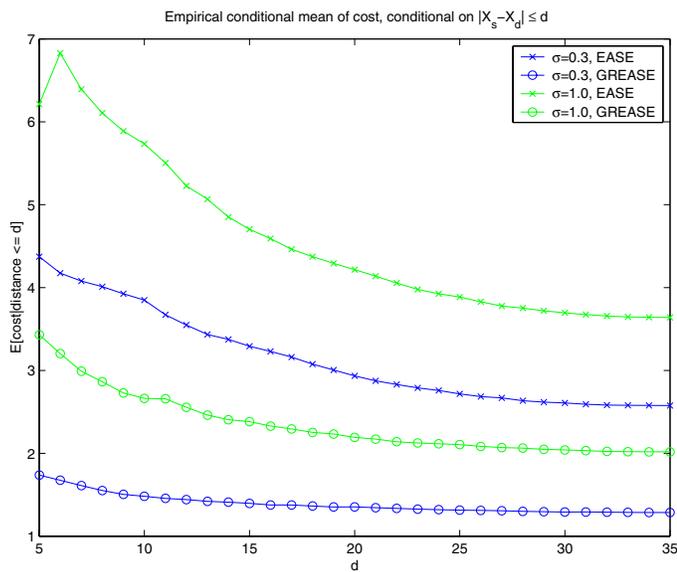


Fig. 7. The empirical conditional mean of the normalized cost, conditional on the initial source-destination distance $|X_s(0) - X_d(0)| \leq d$, plotted as a function of d .

Gaussian increments, heterogeneous mobility. For the sake of discussion, assume a static population and a single fast moving node. Clearly, this is an unfavorable situation. The source node needs to find the trail of the destination node, and then the packet simply follows the trail. If source and destination have met at time $-T_0$, the expected distance between source and destination is $O(\sqrt{T_0})$, but the path length is $O(T_0)$. In this case, the incurred cost is $O(\sqrt{T_0})$ larger than optimal.

Thus, consider the following scenario: a small number of nodes moves much faster than the other nodes. More specifically, out of the $N = 1000$ nodes, 10 nodes have $\sigma_{fast} = 0.5$, while the remaining 990 nodes have $\sigma_{slow} = 0.05$. We are interested in this experiment to evaluate the difficulty for a packet to find a fast destination node, compared to a packet with a slow destination. Figure 10 shows the result of this experiment with a heterogeneous population. As expected, the performance of tracking fast nodes based (mostly) on slow mobility diffusion is substantially worse than tracking slow destinations.

It is instructive to watch the actual working of the algorithm for fast and slow destinations, respectively. In Figure 8, a sample path with GREASE shows that the algorithm needs to search around its current location at several points in order to route towards one of the fast destinations, leading to a costly route. In Figure 9, routing to a slow destination does not lead to any local searches at all, and a much better route.

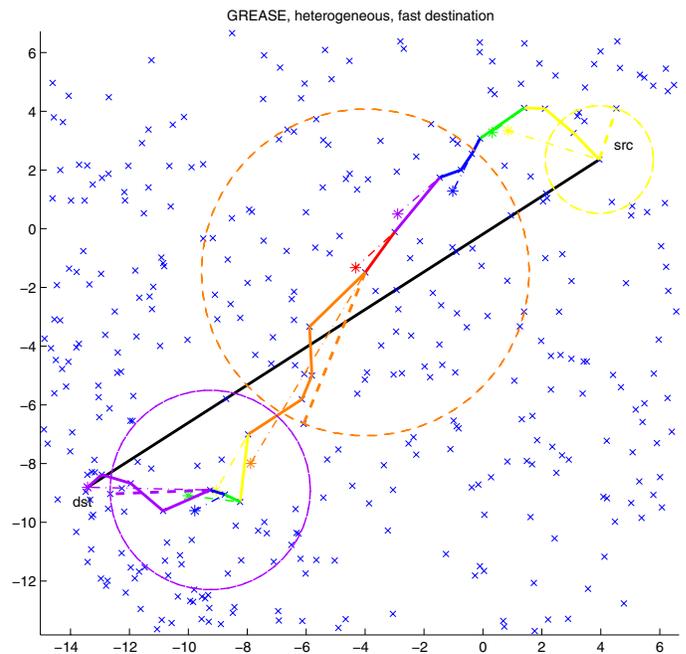


Fig. 8. A sample route for a fast destination with $\sigma_{fast} = 0.5$. Note that GREASE invokes searches around its current anchor point several times, and that the route is relatively long.

Infinite-variance increments. In this scenario, we consider heavy-tailed increment distributions, which allows nodes to make occasional large jumps. Specifically, the complementary CDF $F_R(r)$ of the single-step distance R is given by

$$F_R(r) = \Pr\{R > r\} = \left(\frac{r + \theta}{\theta}\right)^{-\alpha}, \quad (11)$$

with $\theta = 0.2$ and $\alpha = 2$. For this choice of parameters, $E[R] = \theta/(\alpha - 1) = 0.2$. Note that $\text{Var}[R] = \infty$ for $\alpha \leq 2$. The angle Φ of the single-step increment is uniform over $[0, 2\pi)$.

As to be expected, the performance of both EASE and GREASE will degrade due to the unbounded variance of

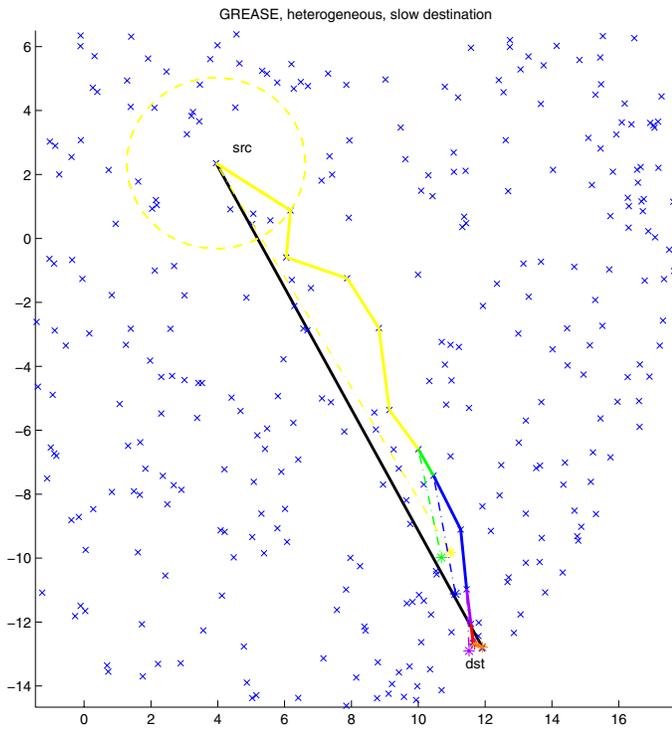


Fig. 9. A sample route for a slow destination with $\sigma_{slow} = 0.05$. Note that GREASE invokes no local searches beyond the initial search around the source, and the route is very efficient.

the steps. However, Figure 11 still shows a decrease in the conditional relative cost with distance.

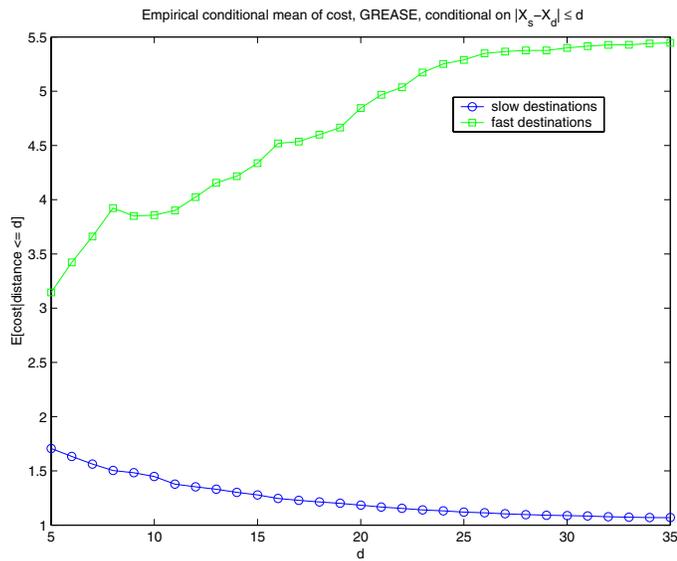


Fig. 10. The empirical conditional mean of the normalized cost, conditional on the initial source-destination distance $|X_s(0) - X_d(0)| \leq d$, plotted as a function of d , for (1) slow destinations; (2) fast destinations.

Random waypoints. A principle at work in our analysis is that a sufficient number of encounters need to be made as the destination travels a given distance. In the random walk case, this number is advantageous, since for a distance $O(\sqrt{T})$,

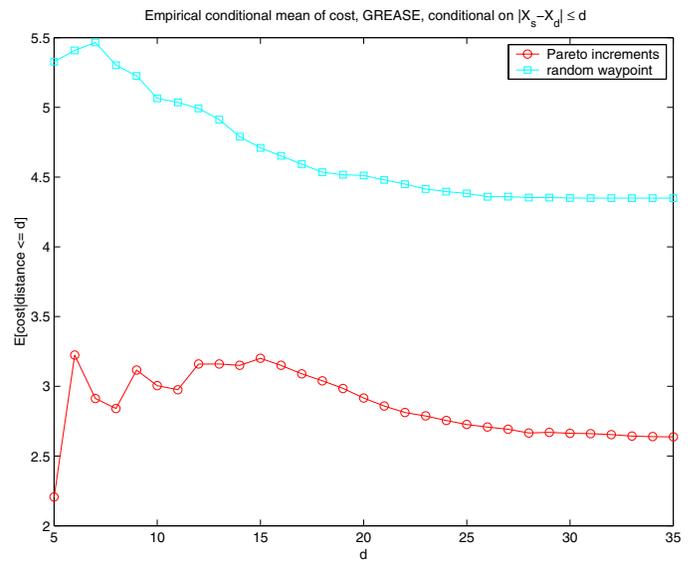


Fig. 11. The empirical conditional mean of the normalized cost, conditional on the initial source-destination distance $|X_s(0) - X_d(0)| \leq d$, plotted as a function of d , for (1) heavy tailed single step distributions; (2) random waypoints.

a number $O(T)$ encounters are made that then spread the information.

Let us consider a case where the number of encounters is of the same order as the distance traveled. In this scenario, nodes do not perform random walks. Rather, each node has a randomly chosen waypoint that it moves towards with constant speed of 0.3/step. Once a node reaches its waypoint, a new waypoint is chosen uniformly on the disk, and the node immediately starts moving towards the new waypoint. As can be seen, while performance is degraded quite a bit with respect to the best case (slow destinations with slow mobility), LE routing still appears to be feasible in this scenario.

VII. DISCUSSION AND CONCLUSION

This paper defines last-encounter routing, a scheme that solely relies on information carried by a packet (in the case of EASE and GREASE, the age and location of the most recent encounter by any node on the packet's path so far) and on the current node's last encounter with the destination. As such, LER uses no network capacity to explicitly update location information. We have shown that LE benefits from *mobility diffusion*, as nodes spread out estimates of the destination's position. As a packet travels towards its destination, it is able to successively refine its estimate of the destination's precise location.

Intuitively, mobility diffusion exploits three salient features of the node mobility processes: *locality*, *mixing*, and *homogeneity*. Locality is a necessary ingredient to ensure that aged information about the last encounter with a destination node is still useful to a packet that tries to find that destination. Mixing of node trajectories (or at least the absence of complete

synchronization of movement processes) ensures that position information about a destination node diffuses around this destination node, because a node continually encounters new neighbors. Homogeneity in the mobility processes ensure that the “speed of diffusion” is of the same order as the movement of a destination, so that location information spreads at least as fast as the destination moves.

The benefit of locality has been recognized and exploited before (e.g., [2], [9]). Specifically, the DREAM algorithm proposes to flood position information about a destination node in a limited area, depending on how far this node has moved [2]. Also, in the GLS system [9], the authors recommend that nearby location servers be updated more frequently than faraway ones, for the same reason. However, the crucial novel observation in the present paper is that at least for certain classes of mobility processes, this limited diffusion of position information *can be obtained for free*: the movements of other nodes that have recently encountered the destination implicitly lead to the same effect, without investing any costly transmission resources. This is certainly of interest in ad hoc networks, where communications costs represent a major bottleneck.

We plan to investigate several ways to further improve the performance of LE routing. First, besides the mobility diffusion based on last encounters, we can use packet-based diffusion. Recall that EASE/GREASE packets carry along the most recent location information for the destination. If a packet passes through a node that does not have a better (more recent) estimate, the node can update its own database for the destination location⁷. For heavy traffic, this clearly can make a difference. This type of diffusion depends on the traffic process, rather than node mobility.

Second, the problem of destination location estimation can be posed as a general estimation problem, with two components: (1) estimation based on a search around the current position of the packet (but taking all the data into account) and (2) estimation based on the whole path of the packet from the source to the current position. In cases where mobility has more temporal structure than a random walk, such estimators may be able to improve performance.

Third, we have focused on the extreme scenario where last encounter routing alone is used and no overhead is incurred for a location service, and we have identified conditions on the node mobility processes under which LER provides efficient routes. In practical settings, mobility processes may possess features that are not captured in the stochastic models studied here. It is an open question how LER performs with more realistic mobility patterns. However, at the very least, we expect LER to have the potential to improve the performance of existing location services. This is an interesting topic for

future research.

ACKNOWLEDGMENTS

The authors gratefully acknowledge fruitful discussions with Henri Dubois-Ferrière and Aman Shaikh.

REFERENCES

- [1] F. Baccelli, K. Tchoumatchenko, and S.Zuyev. Markov Paths on the Poisson-Delaunay Graph with applications to routing in mobile networks. *Advances in Applied Probability*, 32(1), 2000.
- [2] S. Basagni, I. Chlamtac, and V. R. Syrotiuk. A Distance Routing Effect Algorithm for Mobility (DREAM). In *ACM MOBICOM 98*, Dallas, Texas, October 1998.
- [3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Seattle, August 1999.
- [4] S. Giordano and M. Hamdi. Mobility Management: The Virtual Home Region. Technical Report No. SSC/1999/037, EPFL, Lausanne, Switzerland, October 1999.
- [5] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.
- [6] P. H. Hsiao. Geographical Region Summary Service for Geographical Routing. *Mobile Computing and Communications Review*, 5(4):25–39, January 2002.
- [7] R. Jain, A. Puri, and R. Sengupta. Geographical Routing Using Partial Information for Wireless Ad Hoc Networks. *IEEE Personal Communications*, 8(1):48–57, February 2001.
- [8] B. Karp and H. T. Kung. Greedy Perimeter Stateless Routing for Wireless Networks. In *ACM MOBICOM 2000*, Boston, Mass., August 2000.
- [9] 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 *ACM MOBICOM 00*, Boston, Mass., August 2000.
- [10] M. Mauve, J. Widmer, and H. Hartenstein. A Survey on Position-Based Routing in Mobile Ad Hoc Networks. *IEEE Network Magazine*, 15, November 2001.
- [11] Charles E. Perkins. *Ad Hoc Networking*. Addison-Wesley, Boston, 2001.
- [12] Frank Spitzer. *Principles of Random Walk (2nd ed)*. Springer Verlag, 1976.
- [13] G. H. Weiss. *Aspects and Applications of the Random Walk*. (H. E. Stanley and E. Guyon, ed.) North-Holland, 1994.
- [14] V. W. S. Wong and V. C. M. Leung. An adaptive distance-based location update algorithm for next-generation PCS networks. *IEEE Journal on Selected Areas of Communications*, 19(10), October 2001.

⁷Of course, the node can also trivially obtain the *exact* position of the source from a packet. Note that when two nodes establish a two-way session to exchange multiple packets, both nodes would know their exact position after one round of packets. Therefore, suboptimal LE routes would be used only for this first round.