Locating Mobile Nodes with EASE: Learning Efficient Routes from Encounter Histories Alone

Matthias Grossglauser, *Member, IEEE*, and Martin Vetterli, *Fellow, IEEE*

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 $\Theta(n)$ nodes perform independent random walks on a square lattice of size n, the length of the routes computed by EASE are of the same order as the distance between the source and destination *even for very large n*. Therefore, without disseminating 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.

Index Terms— **Mobile wireless networks, mobility, location service, routing.**

I. INTRODUCTION

In large wireless ad hoc and sensor 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 [19].

Position-based (or geographic) routing exploits the fact that nodes

Both authors are with the School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland, {matthias.grossglauser,martin.vetterli}@epfl.ch

usually live in the plane. This enables nodes to make local routing decisions based solely on the destinations' geographic coordinates [15], [2], [16], [3], [13] (see [18] for an excellent review of position based routing). These coordinates can be obtained directly by equipping nodes with GPS receivers, or indirectly by inferring a coordinate system based on local measurements of connectivity, such as signal strengths or run-time differences [5], [4]. 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 [11] 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 [18], [17], [12], [24]. 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 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 LER invests no network capacity to track nodes, i.e., to maintain distributed location information.
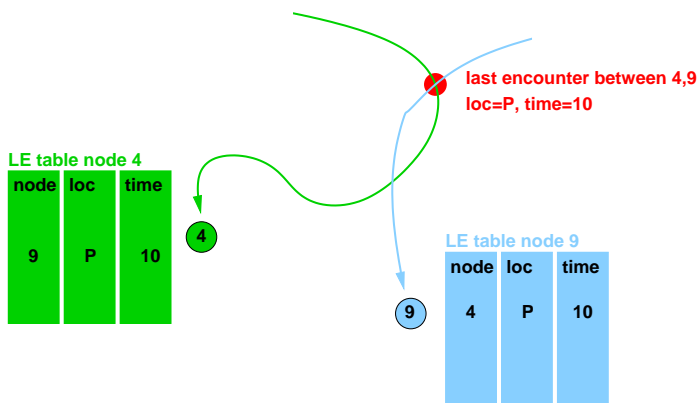

Fig. 1. A *last encounter table* in every node remembers the location and time of the last encounter with every other node in the network. In last encounter routing (LER), this table is queried by a packet to improve, if possible, its estimate of the location of its destination node.

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 LER 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 LER 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 define the *Exponential Age Search (EASE)* routing algorithm, a LER algorithm tailored to random walk mobility. In Section V, we examine the asymptotic performance of EASE when the network size grows large, using standard results from the theory of random walks. We show that the expected route cost obtained with EASE is of the same order as the optimal path length even in very large networks. Section VI gives simulation results in fairly large networks (1000 nodes); they confirm the good performance and scalability of EASE. Finally, Section VII provides some further insights and discussions.

II. RELATED WORK

Several position-based routing algorithms have been proposed in the literature [15], [2], [16], [3], [13]. 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 [24]). 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 [18], [17], [8], [12], [24]. 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) [17], 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) [12], 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 [18] 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 $\mathbb{R}^2$. That is, nodes live on vertices of the $\sqrt{n} \times \sqrt{n}$ square lattice. More precisely, the topology is a two-dimensional square grid of n vertices $(x, y) \in \{1, \ldots, \sqrt{n}\}^2$. We also assume that the border vertices wrap around to form a torus. The distance metric is lattice (Manhattan) distance (i.e., $(1, 1)$ and $(\sqrt{n}, \sqrt{n})$ are at distance two, for example).

There are $m = \lambda n$ mobile nodes that move on this grid, where λ is the node density. Let $X_i(t)$ denote the lattice vertex where node i is located at discrete time t. We assume that each node always knows its current position on the grid as well as the identity and positions of its neighbors. A node j is a neighbor of node i at time t if $||X_i(t) - X_j(t)|| \leq 1$.

Note that this definition does not guarantee a connected node topology. However, the importance of our definition of adjacency lies in the encounters between nodes and the diffusion of this encounter information through mobility. We will simply assume that forwarding a packet from a node i to another node j that are at Manhattan distance d of each other requires at most d transmissions (hops).

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

Routing. We now define the notion of a *last-encounter routing scheme*. In such a scheme, the only information a node maintains

about the network topology is the last encounter table. In this table, each node i maintains an entry (P_{ij}, A_{ij}) for every other node j, where P_{ij} denotes the lattice position of node j when i and j were last neighbors, and A_{ij} measures the age of that encounter, i.e., the time elapsed since the encounter.[1] A packet has to traverse a sequence of nodes from its source node to its destination node. Each node along the way has to make a routing decision for this packet, i.e., it has to decide which neighbor to forward the packet to. In a LER scheme, this decision depends on three things: (i) the identity of the destination node carried in the packet, (ii) potentially some additional information carried in the packet, which can be a function only of information in the nodes that the packet has traversed so far, and (iii) the last encounter table in the current node. In this paper, we describe specific LER schemes where (ii) takes the form of the best, i.e., most recent, last encounter entry the packet has seen for its destination node so far.

Without loss of generality, for the remainder of this paper, we rely on the following simplifying assumptions. First, we will focus on a single destination node with index d, with all other nodes serving as potential relays for packets destined for node d. As we focus on a fixed destination d, we write the last encounter entry of node i for destination d as $(P_{id}, A_{id}) = (P_i, A_i)$. Second, we consider the network at time $t = 0$.

Mobility. The position $X_i(t)$ of node i is a random process with uniform stationary distribution over the lattice. The processes for different nodes are independent. More specifically, we assume that $X_i(t)$ is a two-dimensional unbiased aperiodic random walk on the square lattice, i.e.,

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

where the $\Delta_i(t)$ are i.i.d [22]. Moreover, we assume for simplicity that both components of $\Delta_i(t)$ are i.i.d., with zero mean and finite variance σ^2.

Cost metric. We let the random variable C_n denote the total number of transmissions (or hops) necessary to transmit a packet from a source s to a destination d at time $t = 0$. This cost will include both transmissions of the actual packet between two neighboring nodes 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. Note that because of symmetry, the cost between every pair of nodes has identical distribution; also, the cost obviously does not depend on time, assuming that the system has reached steady state.

IV. THE EXPONENTIAL AGE SEARCH (EASE) ALGORITHM

We now define the specific LER scheme examined in this paper, which we refer to as the *Exponential Age Search* (EASE) routing algorithm. EASE is a LER algorithm, i.e., it 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, other than local "hello" packets for neighbor discovery. The goal of EASE is to

---

[1]The encounter can equivalently be recorded as the *time* of the encounter; we work with the age purely for simplicity of exposition.

be so simple as to be amenable to analysis and to provide insight into the conditions under which mobility diffusion provides good routes. We will also discuss a slight modification of EASE that improves performance, but is less amenable to analysis.

We fix three constants α, β, and c such that $0 < \alpha < 1$, $c > 0$, and $\beta > 0$.

Algorithm 1: The EASE algorithm

1   Set $T_0 := \frac{n}{(c\sigma)^2}$, $Y_0 := X_s(0)$, $k := 0$.
2   Repeat
3   Search the nodes around Y_k in order of increasing distance until a node i is found such that either
4   (a) $A_i \leq \alpha T_k$ [age criterion] or
5   (b) $A_i < T_k$ and $|P_i - Y_k| > \beta\sigma\sqrt{T_k}$ [distance criterion].
6   Let $T_{k+1} = A_i$ be the new age, and $Y_{k+1} := P_i)$ be the new anchor.
7   While not at Y_{k+1}
8   Route packet: find next hop j towards Y_{k+1} and forward packet to j.
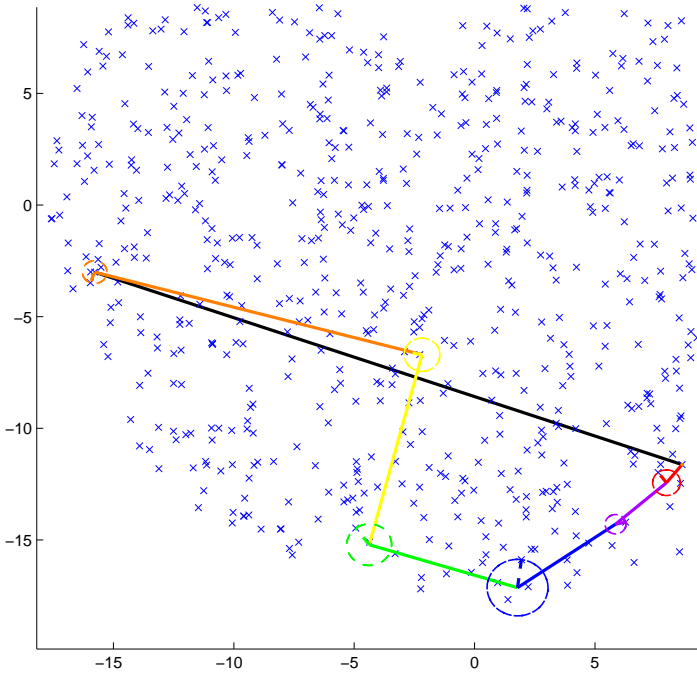9   End while
10      $k + +$.
11   Until $Y_k = X_d(0)$.

Fig. 2. A sequence of anchors computed by the EASE algorithm, with parameter $\alpha = 1/2$; the search disks $\mathcal{S}_k$ are also shown. The source node is to the left, the destination to the right. A node in each search disk around anchor Y_k provides the next anchor Y_{k+1}, which corresponds to the point on the trajectory of the destination of at most half the age ($\alpha = 1/2$) of the current anchor.

Initially, the packet is at its source at position $X_s(0)$. The basic idea behind EASE is for a route to follow the trajectory $X_d(.)$ of the destination node between $t = -T_0$ and $t = 0$ in "jumps" of decreasing length, until the packet arrives at the current position $X_d(0)$ of the destination node (cf. Fig. 3). We call the end-

points of such jumps *anchors*. We do not prescribe a particular routing algorithm for the packet to get from one anchor 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) 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.
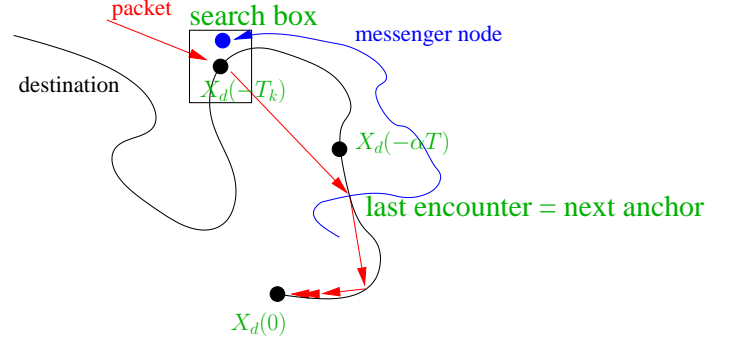

Fig. 3. At every anchor Y_k of age T_k, the EASE algorithm performs a search until a new anchor of age at most αT_k is found.

The EASE algorithm operates in two alternating phases. In the first phase, when a packet has reached an anchor, it performs a local search around that anchor to find the next anchor. In the second phase, an existing position-based routing algorithm is used to route the packet towards the new anchor. 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 the packet is currently headed to, then that estimate is assumed to be the new anchor.

Algorithm 2: The Greedy EASE algorithm

1   Set $T_0 := \frac{n}{(c\sigma)^2}$, $Y_0 := X_s(0)$, $k := 0$.
2   Repeat
3   Search the nodes around Y_k in order of increasing distance until a node i is found such that either
4   (a) $A_i \leq \alpha T_k$ [age criterion] or
5   (b) $A_i < T_k$ and $|P_i - Y_k| > \beta\sigma\sqrt{T_k}$ [distance criterion].
6   Let $T_{k+1} = A_i$, and $Y_{k+1} := P_i$ be the new anchor.
7   While not at Y_{k+1}
8   Route packet: find next hop j towards Y_{k+1} and forward packet to j.
9   If $A_j \leq T_{k+1}$, then $T_{k+1} := A_j$, $Y_{k+1} := P_j$.
10   End while
11      $k + +$.
12   Until $Y_k = X_d(0)$.

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). In this case, a search is performed only once around the source. An example of this case is provided in Figure 4.
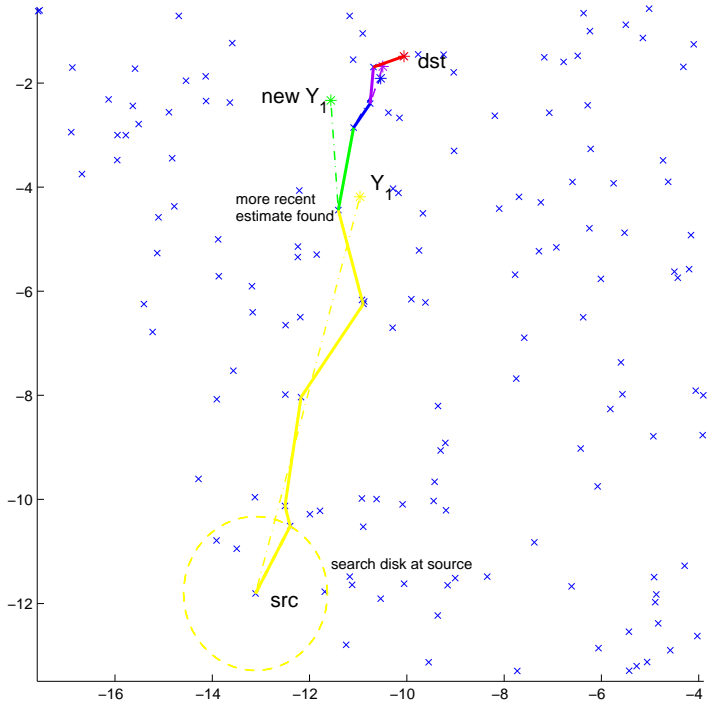

Fig. 4. A sample GREASE route. In this case, the greedy local improvement of the packet's anchor carries the packet all the way to the destination, without any further searches after the initial search around the source.

V. 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 of the same order as shortest routes, even for very large networks. Our main result is as follows.

Theorem 5.1: For two arbitrary nodes s and d, the route from s to d calculated by the EASE algorithm satisfies

$$\mathsf{E}\left[C_n\right] = O\left(\sqrt{n}\right). \tag{2}$$

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

We first provide an outline of the proof. The main idea is to show that the *forwarding cost*, i.e., the length of the route found by EASE, is of the same order as the shortest path, i.e., the distance between the source and the destination. Then, it is shown that the search cost is asymptotically negligible with respect to the forwarding cost.

First, consider a route that has been found using only the age criterion (a) in EASE. The successive ages T_k found by EASE decrease exponentially at a rate of at least α. For a random walk, this implies that the length of the segments between anchor decreases exponentially as well, which ensures that the sum converges and is of the same order as the shortest path.

We then need to show that the search cost is at most of the same order as the forwarding cost. In fact, we show that the search cost for a single iteration - or step - of EASE (i.e., going from an anchor of age T_k to an anchor of age at most αT_k) becomes asymptotically very small relative to the expected one-step forwarding cost. This is because the density of nodes around an anchor Y_k that satisfy either the age criterion (a) or the distance criterion (b) is quite high. More precisely, we show that roughly one in $\log T_k$ nodes in the vicinity of Y_k satisfy one of the criteria, requiring a search of expected cost $\log T_k$.

To show this, we focus in Theorem 5.1 only on nodes that the destination encountered between $-T_k$ and $-\alpha' T_k$, where $0 < \alpha' < \alpha$ is some constant. This ensures that these encountered nodes have enough time to travel, on average, as far as the destination. More precisely, assume the destination encounters a node at some time t, with $-T_k \leq t \leq \alpha' T_k$. Then the time the destination took to travel from Y_k to $X_d(t)$ is at most $(1-\alpha')T_k$, and of the same order as the length of time that the encountered node travels between time t and time 0, which is at least $\alpha' T_k$.

A complication arises because it is possible that the destination travels atypically far in a step. This is why we also need the distance criterion (b) in EASE. If the destination node has traveled atypically far between time $-T_k$ and time $-\alpha' T_k$, then the probability of finding a messenger node i that satisfies (a) in an iteration of the EASE algorithm can be very small. However, in this case, it is easy to find a node i satisfying (b), i.e., a node providing a new anchor whose age is more than αT_k, but still makes typical progress towards the destination in terms of distance. Note that the distance criterion alone would not be sufficient, as it would fail in the typical case where the destination has not traveled far.

It is useful to make the following definition. We call a box $B(P, s)$ centered at position P of size s the smallest square region with center at P that contains s vertices. Note that the expected number of nodes in a box $B(P, s)$ is $\lambda \|B(P, s)\|$.

Furthermore, we denote by S_k the size of the smallest box $B(Y_k, .)$ centered at anchor Y_k that contains the node i providing the next anchor Y_{k+1}. Note that the cost of searching this search box is proportional to S_k. In 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.

We now give the proof of Theorem 5.1, where we only sketch certain aspects of the argument for the sake of readability. We also provide some intuition of why EASE succeeds in computing asymptotically efficient routes, and crystallize out the salient features of the node mobility processes that permit this efficiency.

Proof: Let us consider the kth iteration of EASE, and derive its 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., $Y_k = X_d(-T_k)$.

Now consider the trajectory of the destination node over the interval $[-\alpha T_k, -\alpha' T_k]$. The goal of the kth iteration of EASE is to find a node i close to Y_k at time $t = 0$ that provides a new anchor $Y_{k+1} := P_i = X_d(-T_{k+1})$, such that $T_{k+1} \leq \alpha T_k$. We have to determine how many nodes S_k have to be searched on average until a new anchor Y_{k+1} is found. We will show that the cost of searching for this new anchor is small compared to the cost of actually forwarding the packet to Y_{k+1}.

Typical excursion of destination node. We first condition on the maximum excursion of the destination between time $t = -T_k$ and time $t_2 = -\alpha' T_k$. Assume a box $B = B(Y_k, c\sigma\sqrt{T_k})$, where $c > 0$ is some constant (cf. Fig. 5). The probability that the random walk starting at Y_k never leaves the box converges to a nonzero constant. This can be seen by rescaling the random walk as $X_i(tT_k)/\sqrt{T_k}$, which converges to a Brownian motion for $T_k \to \infty$. The probability of leaving the box is therefore asymptotically equal to the probability of the Brownian motion running for one time unit to hit a box of constant sidelength. From now on, we condition on the event F that the destination does not leave B.

Find a messenger node around current anchor Y_k. Consider a node i that is a neighbor of the destination node at some time t between $t_1 = -\alpha T_k$ and $t_2 = -\alpha' T_k$. We call such a node a *messenger node*. The size S_k of the search box will be determined by the locations of the messenger nodes at time 0 (cf. Fig. 5).

Note that we are conservative in only considering messenger nodes encountered up to time $t_2 = -\alpha' T_k$. This assumption ensures that the distance traveled by the destination between $-T_k$ and the time of encounter $t \in [t_1, t_2]$ is comparable to the distance traveled by the messenger node between t and 0.

Consider the set of messenger nodes $W = \{i : ||X_i(t) - X_d(t)|| \leq 1, t_1 \leq t \leq t_2\}$, i.e., the set of nodes that are neighbors of the destination node at some point between t_1 and t_2. Let H denote the event that one of these messenger nodes hits a search box $B(Y_k, s)$ of given size s, and let $H_i = \{X_i(0) \in B(Y_k, s)\}$ denote the event that messenger node $i \in W$ hits this search box. The $\{H_i\}$ are dependent, but conditionally independent given a destination trajectory $X_d(-T_k, -\alpha' T_k)$, because the $X_i(.)$ are independent.

We find a stochastic upper bound for S_k by making two conservative assumptions about the messenger nodes in W. First, we assume that each messenger node i has its encounter with the destination (i) at the latest possible moment, i.e., at t_2, and (ii) as far away from Y_k as possible, i.e., at a corner of B (recall that we condition on the event F that the destination does not leave B after $t = -T_k$). In other words, for each $i \in W$, we replace the process $X_i(.)$ with another process $X_i'(.)$ that starts in a corner
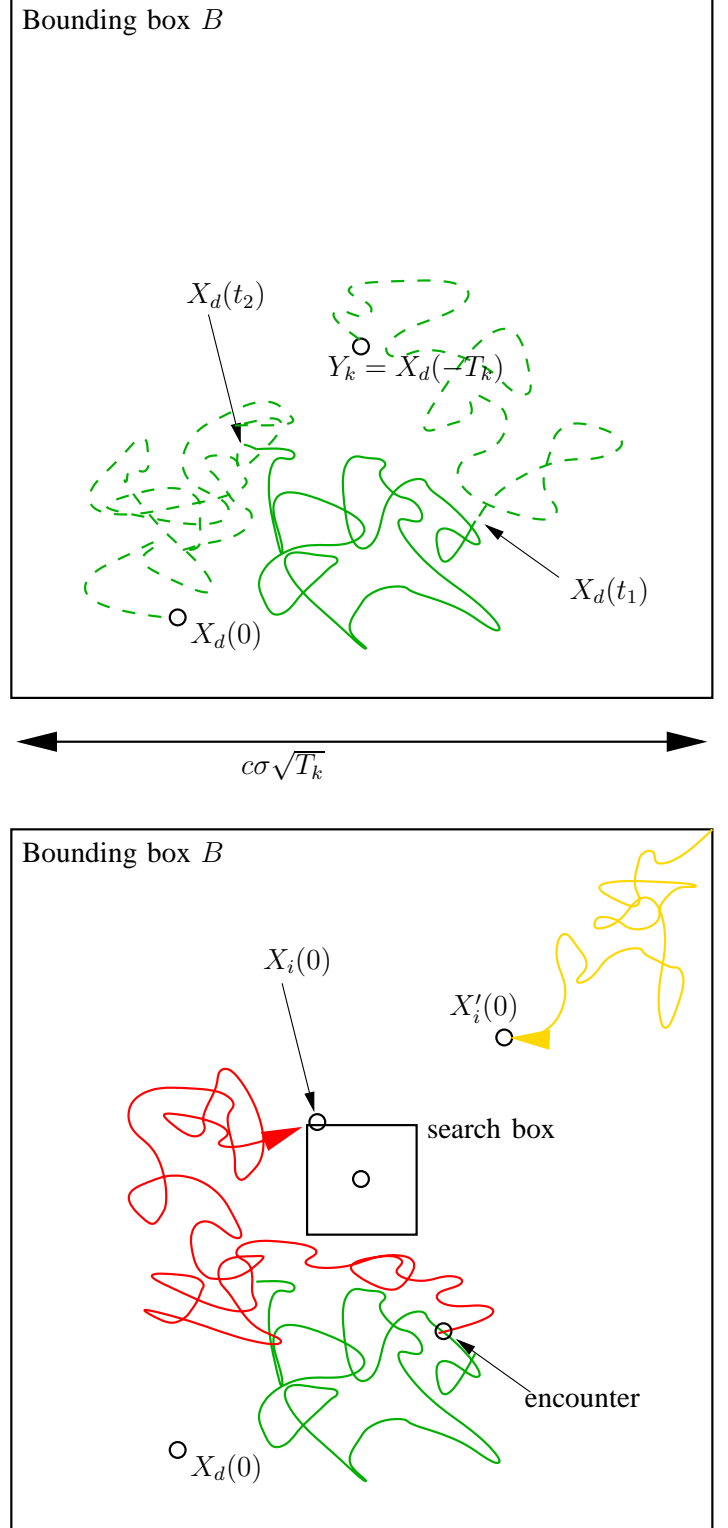

Fig. 5. One EASE step. (a) Only the nodes that the destination encounters between $t_1 = -\alpha T_k$ and $t_2 = -\alpha' T_k$ are considered messenger nodes in the proof. (b) For a search to be successful, a messenger node i encountered by the destination between $-\alpha T_k$ and $-\alpha' T_k$ must perform a random walk to end up within the search box at time $t = 0$.

of B and runs only for $\alpha' T_k$ time steps (cf. Fig. 5).

We now compute an upper bound of $\mathsf{P}\{\bar{H}\}$.

$$\mathsf{P}\{\bar{H}\} = \mathsf{P}\left\{\bigcap_{i \in W} \bar{H}_i\right\}$$

6

$$
\begin{aligned}
&= \sum_{X_d(t_1,t_2)} \mathsf{P}\left\{ \bigcap_{i \in W} \{X_i(0) \notin B(Y_k, s)\} \,\middle|\, X_d(t_1,t_2) \right\} \times \\
&\qquad \mathsf{P}\{X_d(t_1,t_2)\} \\
&\overset{(a)}{=} \sum_{X_d(t_1,t_2)} \prod_{i \in W} \mathsf{P}\{X_i(0) \notin B(Y_k, s) \,|\, X_d(t_1,t_2)\} \times \\
&\qquad \mathsf{P}\{X_d(t_1,t_2)\} \\
&\overset{(b)}{\leq} \prod_{i \in W} \mathsf{P}\{X_i'(0) \notin B(Y_k, s)\} \\
&\overset{(c)}{\leq} \left(1 - \frac{c_1 s}{T_k}\right)^{\|W\|},
\end{aligned}
\tag{3}
$$

where $c_1 > 0$ is a constant that depends on the statistics of the random walk and α, α', and where the sum is over all trajectories that remain inside B. In (3), (a) is because the $X_i(0)$ are conditionally independent, (b) is because X_i' is independent of X_d, and by assumption starts at least as far away from Y_k as X_i, and travels for at most as long, and (c) is because for every position $P \in B$, the probability $\mathsf{P}\{X_i'(0) = P\} \geq c_1/T_k$. This follows because we scale both the surface area of the box B and the runtime of the walk $X_i'(t_2, 0)$ as $\Theta(T_k)$.

Number of messenger nodes $\|W\|$. The size S_k of the search box required to find the next anchor is the smallest box centered at Y_k that contains one of the positions $X_i(0), i \in W$.

Note that the absolute number E of encounters of the destination with other nodes in an interval of length T satisfies

$$
\mathsf{E}[E] = 5\frac{m-1}{n}T \to 5\lambda T,
\tag{4}
$$

which is a simple consequence of the law of large numbers. Note that this is different from the number of encountered nodes $\|W\|$, because it is possible that the destination encounters a node more than once.

We can determine $\|W\|$ by noting that the difference between $X_d(t)$ and $X_i(t), i \neq d$ is also a random walk. First condition on the destination trajectory $X_d(0, T)$. Let E_i denote the number of encounters between node i and the destination between 0 and T, and W the set of nodes that encounter the destination at least once in $[0, T]$.

Focus on a node i that encounters the destination, and assume w.l.g. that this happens at the origin $(0,0)$. Note that $\{E_i, i \in W\}$ are conditionally independent, conditional on the destination trajectory $X_d(0, T)$, because E_i is the number of returns to the origin of the distance random walk $X_d(t) - X_i(t), i \in W$. Now

$$
\begin{aligned}
&\mathsf{P}\{E_i > k \,|\, X_d(0,T) = x_d(0,T)\} \\
&\leq \mathsf{P}\{E_i > k \,|\, X_d(0) = \ldots = X_d(T) = (0,0)\},
\end{aligned}
\tag{5}
$$

for $k \geq 1$, which follows from the fact that the return time to origin stochastically lower-bounds the time to reach any other state [22].

The returns to the origin form a renewal process. It is known that the tail of the probability of not returning to the origin over a long time-interval T [23, Chapter 4, p. 125] behaves as $\propto 1/\log T$. We can upper-bound E_i by conservatively assuming that every return to origin is an independent Bernoulli trial with success probability $c_2/\log T$. This means we can stochastically upper-bound E_i with

a geometric random variable E_i' with parameter $p = c_2/\log T$, which has mean $\mu = O(\log T)^2$.

As $\sum_{i \in W} E_i = E$, and using the bounding variables E_i', the Renewal Theorem [20] asserts that for $E \to \infty$, the number of nodes W encountered by node d satisfies $\frac{\|W\|}{E} \to \frac{1}{\mu}$, and therefore

$$
\|W\| > c_3 \frac{T}{\log T}
\tag{6}
$$

with high probability, for some $c_3 > 0$.

Bound on the size of the search box S_k. We now upper-bound the size of the search box S_k. For this, we conservatively assume that we try a sequence of search boxes with sizes that are a multiple of $s = \log T_k$.

Consider first a search box of size s. The probability that at least one of the nodes in the set W hits the search box $B(Y_k, s)$ satisfies

$$
\mathsf{P}\{H\} \geq 1 - \left(1 - \frac{c_1 s}{T_k}\right)^{c_3 T_k / \log T_k} \to c_4,
\tag{7}
$$

combining (3) and (6)[3].

We also need the following elementary property. Consider two search boxes at Y_k of size s and $s' > s$. Condition on the event that none of the messenger nodes hits the search box of size s. It should then be intuitively clear that the conditional probability of hitting the larger search box of size s' is slightly higher than the unconditional probability, because the conditional density of nodes outside the smaller box is higher than the unconditional density.

Consider a sequence of search boxes of sizes js, $j = 1, \ldots$, and consider the random variable J, which is the first box for which a messenger node hits. By virtue of the above argument and of (7), J is stochastically upper bounded by a geometric random variable with mean $1/c_4$. This gives

$$
\mathsf{E}[S_k] \leq s/c_4 = O(\log T_k).
\tag{8}
$$

Atypically large excursion of destination. We have so far conditioned on the event F that the destination stays inside B between $t = -T_k$ and time $t = 0$. If F does not occur, then it is not conservative any longer to let messenger nodes start from a corner of B. We briefly sketch how we can avoid this difficulty.

If the random walk $X_d(.)$ leaves B at some point after $-T_k$, then an appropriate choice of β ensures that there are sufficient nodes encountered outside B, but still close enough to Y_k, that satisfy the distance criterion. Therefore, even though the number of messenger nodes is too low (possibly zero), we can show with a similar argument as above that it is not too costly to find a node around Y_k that satisfies the distance criterion.

If we move to an anchor found through the distance criterion rather than the age criterion, then we reduce the age by a factor

---

[2]Strictly speaking, this only holds for a random walk on the infinite lattice, not a finite torus. However, the additional probability of returning through a "wrap-around" in the torus is negligibly small.

[3]Note that we have omitted an additional error term due to the residual probability of W being atypically small in (6); this can be shown to be negligible, by considering the speed of convergence in (6) for large T. We omit the details here.

less than α. It remains to show that this does not change the order of the forwarding cost. This follows from the fact that the increments of the random walk over non-overlapping time intervals are independent. Therefore, we can conclude that the total cost to reduce the encounter age from T_k to at most αT_k is of order $O(\sqrt{T_k})$.

Total cost C_n. The total cost incurred by EASE to route a packet from the source s to the destination d is

$$C_n = \sum_{k=0}^{K-2} ||Y_k - Y_{k+1}|| + S_k$$

(9)

where $K = O(\log n)$ is the number of steps required to reach the destination.

Note that the initial age T_0 at the source is $O(n)$, and chosen to make the box $B = B(Y_0, c\sigma\sqrt{T_0})$ equal to the entire torus. 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 by a factor of at least α with every iteration, the sequence of distances between successive anchor $||Y_k - Y_{k+1}||$ decreases geometrically. Its sum converges and is therefore $O(\sqrt{n})$. ∎

We have thus shown that last encounter routing is asymptotically competitive when all the nodes perform random walks, because mobility diffuses estimates of the destination's position sufficiently quickly and densely. Note that the exponential search criterion is important to achieve this performance. This is because there is a tradeoff between the forwarding and the search cost. If the search criterion were more aggressive, the search cost could become very large; an extreme example of this is if the source searched for encounter age $T_0 = 0$, which would result in a single flood of size $\Theta(n)$. On the other hand, if the search criterion were less aggressive than a reduction by at least a factor α, then the forwarding cost could become very large. To see this, note that the total length of the trajectory of a random walk observed over an interval of length t is of order t; however, the distance between the start and end positions of the walk, and therefore the length of the shortest path, is only of order $\sqrt{t}$. If the age criterion were chosen too loosely, then the algorithm would be allowed to compute routes whose length is close to the total length traversed, and therefore much longer than the shortest path.

Note that the distance criterion is a device to avoid complications in the proof for atypical excursions of the destination. In practice, it appears that the distance criterion is in fact not necessary, which can be explained by the various conservative assumptions we make in the proof. Chief among them is the fact that we disregard messenger nodes of age less than $\alpha' T_k$, and the conservative assumption that each messenger node starts in a corner of B, maximally far away from Y_k. In our simulations of EASE below, we in fact do not use the distance criterion.

It is quite remarkable that efficient routes can be computed in a network where the only control traffic is for neighbor discovery. However, the above result does not imply that the same property holds for other mobility models. In Section VI, we resort to extensive simulations to explore the robustness of the EASE algorithm under different assumptions. Here, we wish to provide insight on this question by discussing some of the 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 locations of a node 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.

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 V 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 reflecting[4]. We look at Gaussian 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 [14].

The node positions are initially uniformly distributed over the disk. We then run the random walks for a sufficient warm-up

---

[4]Note that we go back to the Euclidean domain.

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 warm up 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, 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

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

where A is a random set of source-destination pairs, and where $\hat{C}_n$ 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. In all the simulations, we have chosen $\alpha = 1/2$, $\alpha' = 1/4$, and $\beta = \infty$, i.e., EASE operates without the distance criterion. Also, instead of using initial age $T_0 := \frac{n}{(c\sigma)^2}$ as given in the definition of EASE, we in fact use $T_0 := T_s(0)$, i.e., the encounter age of the source (which is possibly infinite if the source has never met the destination). This has the advantage that the algorithm can operate without knowledge of σ^2 (but is more difficult to analyze analytically).

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, approximately 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., $|X_s(0) - X_d(0)| \leq x$. 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.
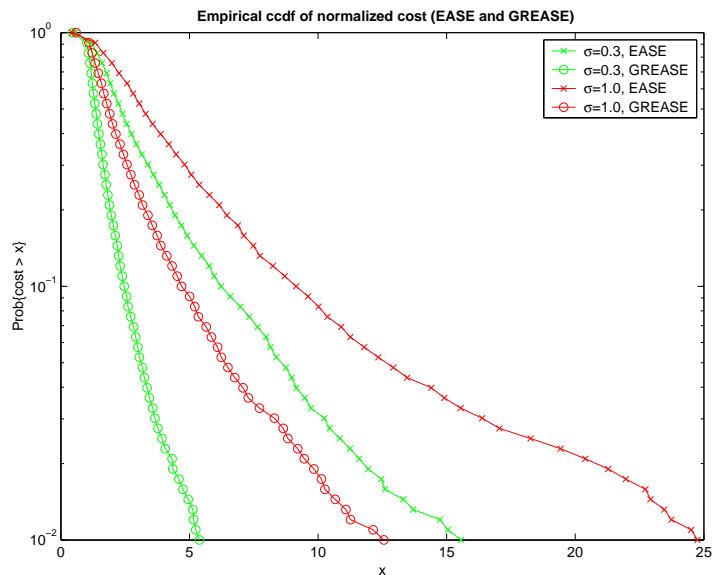

Fig. 6. The empirical complementary CDF (CCDF) of relative cost for both EASE and GREASE and various values for the single-step standard deviation σ.

Gaussian increments, heterogeneous mobility. For the sake of discussion, assume a static population and a single fast moving destination 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.
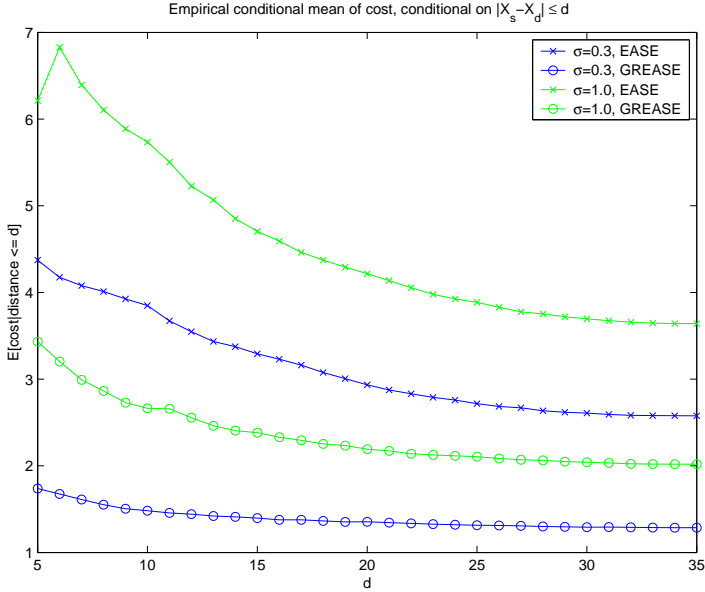
Fig. 7. The empirical conditional mean of the normalized cost, conditional on the initial source-destination distance $|X_s(0) - X_d(0)| \leq x$, plotted as a function of d.
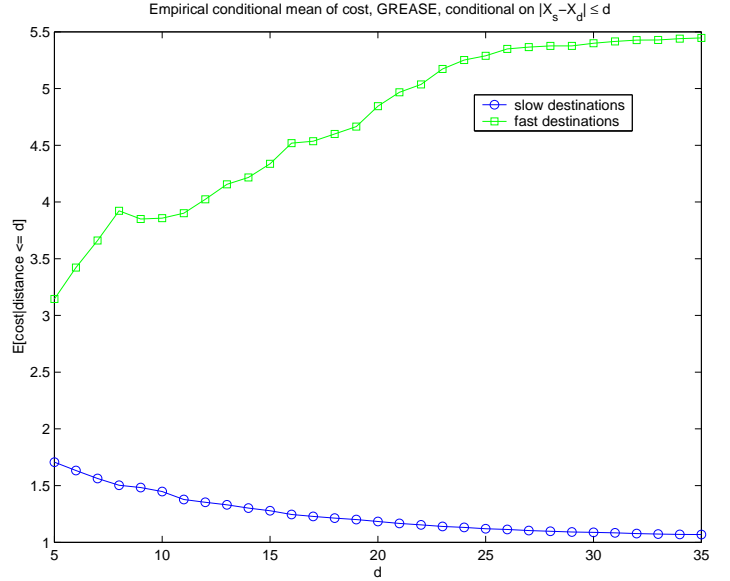

Fig. 8. The empirical conditional mean of the normalized cost, conditional on the initial source-destination distance $|X_s(0) - X_d(0)| \leq x$, plotted as a function of d, for (1) slow destinations; (2) fast destinations.

Thus, consider the following scenario: a small number of nodes moves much faster than the other nodes. More specifically, out of the $m = 1000$ 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 8 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 9, 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 10, routing to a slow destination does not lead to any local searches at all, and a much better route.

Infinite-variance increments. In this scenario, we consider heavy-tailed increment distributions, which allows nodes to make occasional large jumps. Specifically, we assume a Pareto law for the single-step distance, for which the complementary CDF is

$$\mathsf{P}\{R > r\} = \left(\frac{r + \theta}{\theta}\right)^{-\alpha}, \qquad (11)$$

with $\theta = 0.2$ and $\alpha = 2$. For this choice of parameters, $\mathsf{E}[R] = \theta/(\alpha - 1) = 0.2$. Note that $\mathrm{Var}[R] = \infty$ 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 the steps. However, Figure 11 still shows a decrease in the conditional relative cost with distance.

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
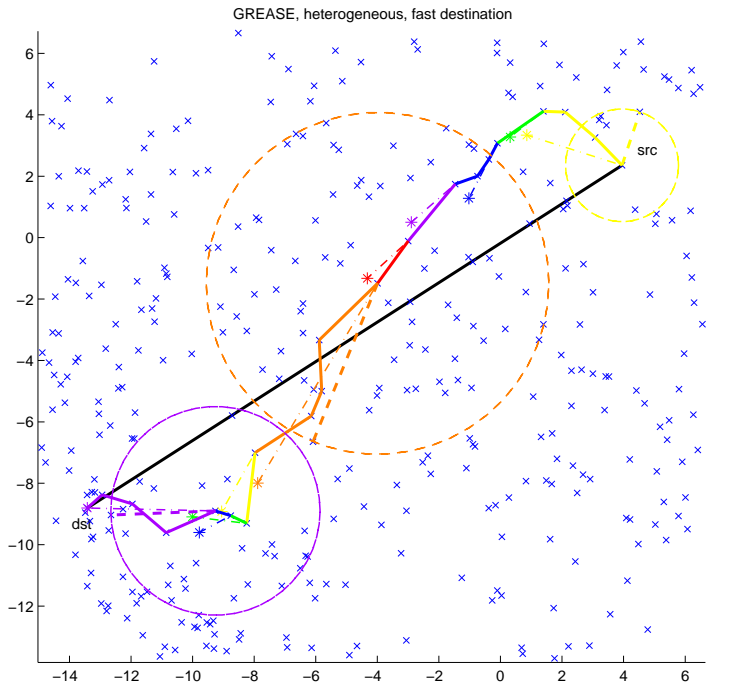

Fig. 9. A sample route for a fast destination with $\sigma_{fast} = 0.5$. Note that GREASE invokes searches around its current anchor several times, and that the route is relatively costly.

is advantageous, since for a distance $O(\sqrt{T})$, 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
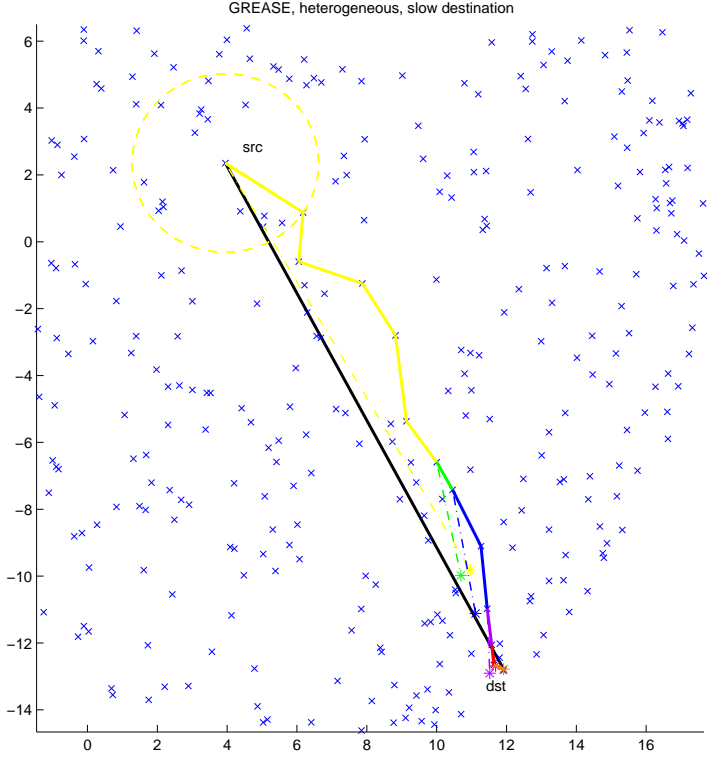
Fig. 10. A sample route for a slow destination with $\sigma_{slow} = 0.05$. Note that GREASE invokes no local searches beyond the initial search around the source, and the route is very efficient.
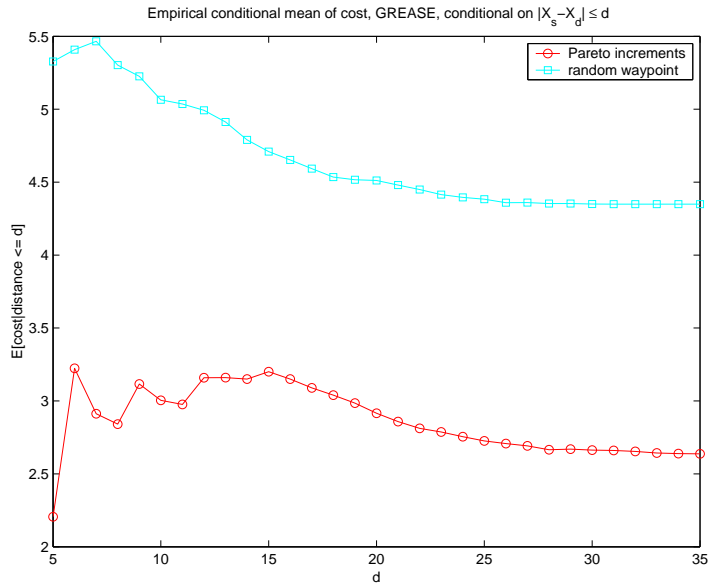

Fig. 11. The empirical conditional mean of the normalized cost, conditional on the initial source-destination distance $|X_s(0) - X_d(0)| \leq x$, plotted as a function of d, for (1) heavy tailed single step distributions; (2) random waypoints.

with slow mobility), LER still appears to be feasible in this scenario.

We give an intuitive explanation for the performance in the random waypoint case. Note that the length of a segment between two waypoints is $O(\sqrt{n})$. In this time, the destination node encounters $O(\sqrt{n})$ other nodes. Within the time scale of a single segment, these $O(\sqrt{n})$ diffuse roughly throughout the network

domain. In other words, after the time it takes the destination to traverse $O(1)$ segments, roughly one node in $\sqrt{n}$ nodes has encountered the destination; furthermore, these nodes are roughly distributed uniformly in the network. Therefore, it takes a search of approximately $\sqrt{n}$ nodes *anywhere in the network* to find an anchor on one of the last segments. Note that this is considerably more costly than in the random walk case. On the other hand, the random waypoint model, as many other mobility processes, is predictable, i.e., the current location of the destination is not a sufficient statistic - contrary to the random walk - for a future location of the destination. In [21], we give an example how this observation can be used to improve the performance of LER under random waypoint mobility.

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], [17]). 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 [17], 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. In fact, LER can be viewed as exploiting the additional transport capacity available in a network due to node mobility to disseminate control information [9]; however, while [9] took the extreme view of ignoring delay, in LER there is a tight interplay between the age of information and where it gets used. This is the reason why the performance of LER depends on much finer details of the mobility processes than does the mobility capacity

as defined in [9].

More generally, we recognize in this work that the collection of the histories of the local connectivity at nodes contains valuable, though noisy information about the current network topology for certain types of mobility models. We expect this observation to have consequences beyond the one analyzed in this paper, i.e., routing in position-aware ad hoc networks. For example, in recent work we have shown that in position-unaware networks, encounter ages alone are valuable to improve the performance of flooding-based node discovery, essentially by walking down a noisy age gradient in a sequence of constrained floods until the destination is found [6], [7].

In some contexts, it is more important for a network to be able to locate information items rather than particular nodes. The concept of LER can readily be applied to such a case, where the goal is to locate, and learn a route to, a data item with some identifier i. In this case, neighboring nodes have to exchange the set of data items that they handle, and update a LE table that maintains encounter entries for data items instead of for nodes. The cost of this will depend, of course, on the number of data items that each node has to maintain.

We plan to investigate several ways to further improve the performance of LER. 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 location[5]. 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.

---

[5]Of 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 LER routes would be used only for this first round.

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] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less Low-Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications*, October 2000.

[5] S. Capkun, Maher Hamdi, and J. P. Hubaux. Gps-free positioning in mobile ad-hoc networks. *Cluster Computing*, 5(2), April 2002.

[6] H. Dubois-Ferrière, M. Grossglauser, and M. Vetterli. Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages. In *Proc. ACM MOBIHOC 03*, Maryland, June 2003.

[7] H. Dubois-Ferrière, M. Grossglauser, and M. Vetterli. Space-Time Routing in Ad Hoc Networks. In *Proc. Ad Hoc Now 03*, Montréal, Canada, October 2003.

[8] S. Giordano and M. Hamdi. Mobility Management: The Virtual Home Region. Technical Report No. SSC/1999/037, EPFL, Lausanne, Switzerland, October 1999.

[9] M. Grossglauser and D. N. C. Tse. Mobility Increases the Capacity of Ad-hoc Wireless Networks. *IEEE/ACM Transactions on Networking*, 10(4), August 2002.

[10] M. Grossglauser and M. Vetterli. Locating Nodes with EASE: Last Encounter Routing for Ad Hoc Networks through Mobility Diffusion. In *Proc. IEEE INFOCOM 03*, San Francisco, March 2003.

[11] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[12] P. H. Hsiao. Geographical Region Summary Service for Geographical Routing. *Mobile Computing and Communications Review*, 5(4):25–39, January 2002.

[13] 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.

[14] David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing (Tomasz Imielinski and Hank Korth, eds.)*, pages 153–181, 1996.

[15] B. Karp and H. T. Kung. Greedy Perimeter Stateless Routing for Wireless Networks. In *ACM MOBICOM 2000*, Boston, Mass., August 2000.

[16] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. In *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Annapolis, Maryland, USA, June 2003.

[17] 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.

[18] M. Mauve, J. Widmer, and H. Hartenstein. A Survey on Position-Based Routing in Mobile Ad Hoc Networks. *IEEE Network Magazine*, 15, November 2001.

[19] Charles E. Perkins. *Ad Hoc Networking*. Addison-Wesley, Boston, 2001.

[20] Sheldon Ross. *Stochastic Processes (Second Edition)*. John Wiley & Sons, 1996.

[21] N. Sarafijanovic-Djukic and M. Grossglauser. Last Encounter Routing under Random Waypoint Mobility. In *Proc. NETWORKING 2004*, Athens, Greece, May 2004.

[22] Frank Spitzer. *Principles of Random Walk (2nd ed)*. Springer Verlag, 1976.

[23] G. H. Weiss. *Aspects and Applications of the Random Walk*. (H. E. Stanley and E. Guyon, ed.). North-Holland, 1994.

[24] 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.