Age Matters: Ef£cient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages [*]

Henri Dubois-Ferriere Matthias Grossglauser Martin Vetterli
School of Computer and Communication Sciences
EPFL
1015 Lausanne, Switzerland
{henri.dubois-ferriere,matthias.grossglauser,martin.vetterli}@ep¤.ch

ABSTRACT

We propose FResher Encounter SearcH (FRESH), a simple algorithm for efficient route discovery in mobile ad hoc networks. Nodes keep a record of their most recent encounter times with all other nodes. Instead of searching for the destination, the source node searches for any intermediate node that encountered the destination *more recently than did the source node itself.* The intermediate node then searches for a node that encountered the destination yet more recently, and the procedure iterates until the destination is reached. Therefore, FRESH replaces the single network-wide search of current proposals with a succession of smaller searches, resulting in a cheaper route discovery. Routes obtained are loop-free.

The performance of such a scheme will depend on the nodes' mobility processes. Under standard mobility processes our simulations show that route discovery cost can be decreased by an order of magnitude, a significant gain given that route discovery is a major source of routing overhead in ad hoc networks.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]:

General Terms

Algorithms, Design, Performance

Keywords

Routing, wireless networks, ad hoc networks

1. INTRODUCTION

1.1 Background and Motivation

Routing protocols for mobile ad hoc networks generate a large amount of control traffic when node mobility causes link states and the network topology to change frequently. On the other hand, resources such as bandwidth and battery power are usually severely constrained in such networks. Therefore, minimizing the control traffic to set up and maintain routing state is one of the main challenges in the design of scalable routing protocols for mobile ad hoc networks.

One approach to limit control traffic is to establish routes *on demand* rather than proactively. On-demand routing protocols [8, 13, 1, 3, 11] only establish a route to a destination when it is necessary to send packets to that destination, and therefore incur less overhead at the expense of higher route setup latency. *Hybrid* routing protocols [7, 6] combine both on-demand and proactive elements for more flexibility in the latency-overhead trade-off.

On-demand routing overhead can be broken down into two components: route discovery and route maintenance. Their relative costs vary depending on the protocol and scenario, but in general route discovery tends to be costly. For example [4] shows that route discovery represents up to 90% of the total routing overhead of AODV [13]. In this paper, we propose a new approach to reduce the cost of route discovery, which can benefit both pure on-demand and hybrid routing protocols.

When a source node first wishes to establish a route to a destination, it must search the network until it finds either the destination or another node which has a route to the destination. Many of the proposed protocols for ad hoc networks perform a flood-based route discovery, whereby a *route request* (RREQ) packet is flooded across the network, possibly using an expanding ring search to "grow" the flood until the destination is found [8, 13, 1, 3, 11]. This search is omnidirectional: since the source node does not know where the destination lies, the flood cannot favor any one particular direction.

In this paper, we propose an algorithm called FRESH that improves the performance of route discovery over omni-directional approaches. FRESH achieves this performance improvement by exploiting the history of last encounters between nodes (two nodes encounter each other when they are directly connected neighbors). Our work is motivated by a

[*] The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322

MobiHoc'03, June 1–3, 2003, Annapolis, Maryland, USA.
Copyright 2003 ACM 1-58113-684-6/03/0006 ...$5.00.

simple observation: the history of last encounters between nodes contains valuable, but noisy information about the current network topology.

In previous work [5], we focused on the case of geographic routing protocols. In geographic routing, every node is aware of its coordinates in euclidean space, e.g., using GPS. Once the location of the destination is known, packets can be forwarded using the coordinates of neighboring nodes. We showed that if every node maintains an encounter history consisting of the *time and location* of its last encounter with every other node in the network, efficient location lookups can be made based solely on this history.

In this paper, we focus on the case of *blind* routing protocols, i.e., where nodes have no notion of coordinates. We show that if every node maintains an encounter history consisting of only the *time* of its last encounter with every other node, it is possible to significantly improve the performance of flood-based route discovery. We achieve this by relying on encounter ages with the destination to "steer" a flood-based search in the general direction of that destination. This reduces the number of nodes and packet transmissions necessary to find the destination, and therefore scales to larger networks than an omni-directional search. This is illustrated in Fig. 1.

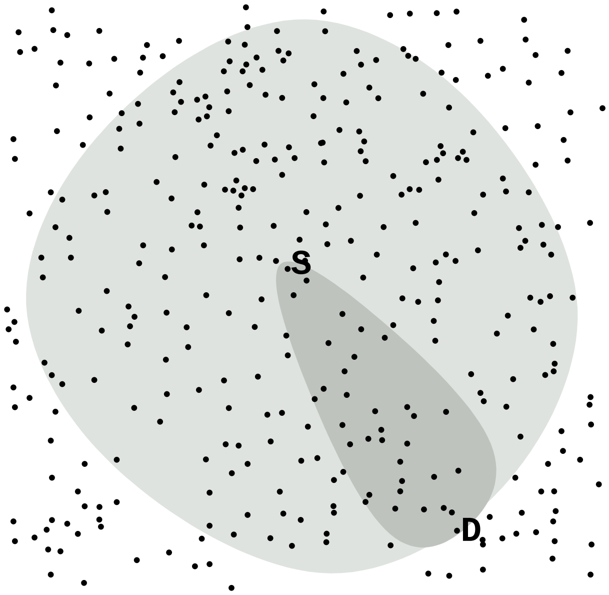

Figure 1: ***Area covered by route discovery floods between source S and destination D. The light grey surface roughly covers the nodes that would be flooded in an omni-directional search. The darker surface shows a search which is 'steered' in the right direction and therefore involves fewer nodes.***

The intuition behind our approach is that encounter ages with a particular destination provide a noisy "age gradient" towards that destination. This gradient results from node mobility. One way of contrasting FRESH with existing approaches is to consider that they route packets through *space*: the primary metric is distance (e.g., hops) to the destination. One the other hand the metric used by FRESH is *time*: packets descend along the age gradient. We now describe the FRESH algorithm, which exploits the age gradient to constrain the search to advance "in the right direction", thereby reducing route discovery overhead.

1.2 FRESH: FResher Encounter SearcH

We require that nodes keep a table of their *most recent encounter times* with all other nodes[1]. An encounter between two nodes happens when those nodes are one-hop neighbors. Since one-hop neighborhood is dependent on the link layer, the exact condition for an encounter to occur will vary depending on the underlying wireless technology used. For example the connectivity range of 802.11b can exceed 250 meters, whereas Bluetooth [14] has a range of several meters at most. The *encounter age* of two nodes n and m is the time elapsed since the most recent encounter of n and m. Encounters can be detected by overhearing any packets (whether regular data packets, or purposely sent "Hello" packets) send by neighboring nodes, or they might be detected at the link layer, as in the case of Bluetooth. In this work we do not assume the use of a specific link layer, since our purpose is to introduce the route discovery algorithm independently from the wireless substrate employed.

We now informally introduce the FRESH algorithm, before giving some insight on how and why it works. Consider a node s that establishes a route to a destination d. We note $T_{LE}(i, d)$ the age of the most recent encounter between nodes i and d, with the convention that $T_{LE}(i, d) = \infty$ if nodes i and d have never encountered, and $T_{LE}(i, d) = 0$ if $i = d$.

Source node s searches[2] for the nearest anchor node a_1 such that $T_{LE}(a_1, d) < T_{LE}(s, d)$ (this is the nearest node which has encountered the destination more recently than s). Node a_1 then searches around itself for the nearest anchor node a_2 such that $T_{LE}(a_2, d) < T_{LE}(a_1, d)$. Anchor node a_2 in turn repeats this fresher encounter search, and the procedure iterates until we reach the destination d (for which $T_{LE}(d, d) = 0$).

Note that this algorithm requires no global knowledge, and lends itself to a distributed implementation, because each search is defined only in terms of the nodes' local encounter tables. Also, the algorithm only makes use of relative times (encounter *ages*), and so clocks need not be synchronized.

Figure 2 shows a route computed with FRESH, where the nodes have been moving according to a random walk. The anchor nodes are represented as crosses. For clarity the path between anchor nodes is approximated by a straight line, though it will in reality follow a multi-hop route through intermediate nodes.

The first performance criterion will be the cost of the n ($n \geq 1$) searches in a route discovery. The baseline to which we will compare FRESH search cost is the search cost of a single-step route discovery as employed by existing protocols. Our simulations in Section 4 show that FRESH allows for a substantial reduction in this cost.

We now give some intuition on what enables fresher encounter search to compute good routes at a lower cost than a single-step route discovery. The basic principle is simple: For most mobility processes, the distance traveled during a

[1] The per-node memory requirements are linear in the number of nodes in the network.

[2] The notion of searching is used informally here; it is exposed in more detail in Section 3.1

time interval of duration t is positively correlated with t. We refer to this as time-distance correlation.

Now consider three nodes i, j, and d. At the present time $t = 0$, node i is separated from node d by a distance D_i, similarly node j is separated from node d by a distance D_j. The intuition behind FRESH is that if $T_{LE}(i, d) < T_{LE}(j, d)$, then *with high probability* $D_i < D_j$. Simply put, "a node that was my neighbor 5 minutes ago is probably closer to me than a node that was my neighbor 5 hours ago". If time-distance correlation holds then successive fresher encounter searches will advance toward the destination. This will result in a directional route discovery as depicted in Fig. 1. We will see that in common mobility processes (random walk and waypoint model) time-distance correlation holds well enough for the algorithm to work very effectively.

The second performance criterion is the quality of routes. Though successive iterations of the fresher encounter search on average bring us closer to the destination, they may not always advance along a straight line, and so we may not obtain the shortest-path route. Since FRESH establishes routes at lower cost than single-step methods, one may consider that we trade off some route quality for a reduction in search cost and so we must be sure that routes remain good enough so that this is worthwhile.

Note that the performance of FRESH is invariant to a rescaling of the node velocities, as this corresponds simply to a rescaling of time, and therefore of all the encounter ages. This does not affect the route or the search cost. On the other hand, performance does depend on the *homogeneity* of the mobility processes. Specifically, if the node velocities are very heterogeneous, then the relationship between encounter age and distance becomes more noisy. For example, the node which saw the target most recently may not be the nearest to the target if it happened to be moving atypically fast.

The outline of this paper is as follows. The next section discusses related work. Section 3 gives a precise formulation of the the FRESH algorithm, and Section 4 gives simulation results for various mobility processes and source-destination distances. Finally Section 5 discusses some interactions between FRESH and the mobility process, and we conclude in Section 6.

2. RELATED WORK

Within the class of protocols for blind nodes, our work applies most directly to those which operate on-demand. This includes AODV [13] and DSR [8] as well as [1], [3], [11]. Such protocols typically use a flooding-based route discovery mechanism, whereby a route request (RREQ) is flooded across the network. These protocols could streamline their route discovery phase by using the FRESH algorithm.

A hybrid approach has been proposed in the ZRP framework [12], which combines an on-demand component with a proactive component. One advantage of a hybrid approach is the flexibility it provides in adjusting the relative weight given to either component. Since ZRP is a framework which accommodates various on-demand routing protocols, it can also benefit from using an on-demand routing protocol which implements FRESH.

The work presented in this paper is not the first attempt to reduce routing overhead by restricting route discovery floods. In [2] a query localization technique is proposed to reduce the scope of network floods on subsequent route discoveries which are made some time after a route was previously known. A recent scalability study [9] has indicated that this technique can reduce routing overhead by up to 50%. We believe that this scheme is complementary to FRESH and that both approaches could be combined to good use, since query localization [2] is useful to re-discover a route which was recently used, whereas FRESH is most useful for the first (and most expensive) route discovery towards a destination.

As we discuss in Section 3.1, FRESH makes use of an underlying search primitive, in order to find the nearest node satisfying the fresher encounter requirement. This requires a mechanism to flood a query packet to neighboring nodes within some radius[3].

The topic of flooding in ad hoc networks has been a subject of much recent work, since efficient flooding is beneficial to many ad hoc protocols, including FRESH. Gossiping, a form of probabilistic flooding, has been shown in [10] to reduce message overhead of AODV by up to 35% over plain flooding. For an in-depth comparison study of simple flooding and probabilistic flooding methods, we refer to [15].

3. THE FRESH ALGORITHM

3.1 Search Primitive

In existing routing protocols for blind ad hoc networks, route discovery is embodied in a single network search, using some form of flooding. We refer to this as a *single-step* route discovery. In this context route discovery is equivalent to carrying out a single network search, and therefore the distinction between route discovery algorithm and the search mechanism is not necessary. On the other hand FRESH makes multiple network searches in the course of establishing a route. It can be seen as a higher-level construct and need not be coupled to the specifics of the search mechanism used. For this reason, we have used the general term of searching.

We now give a precise definition of a network search primitive:

Definition 1. A network **search** primitive is a mechanism allowing a source node s to find the nearest[4] node n which satisfies a given boolean condition, if such a node exists in the network. As a side-effect of a successful search, routing state is established in the network which subsequently allows packets to be sent between nodes s and n.

A few comments on this definition are in order. A search primitive will have two important components: a flooding mechanism to propagate the boolean query, and a routing component which sets up the route between s and n. Possible routing choices include a distance-vector approach or a source routing approach. Possible flooding mechanisms include simple flooding or a form of probabilistic flooding.

We will define FRESH in terms of a general search primitive conforming to Definition 1, in order to emphasize that

[3] Of course, as we assume blind nodes, we cannot actually implement a radius-constrained flood; in practice, time-to-live (TTL) constrained flooding would be a good approximation.

[4] Using a suitable distance metric, for example hop count or euclidean distance.

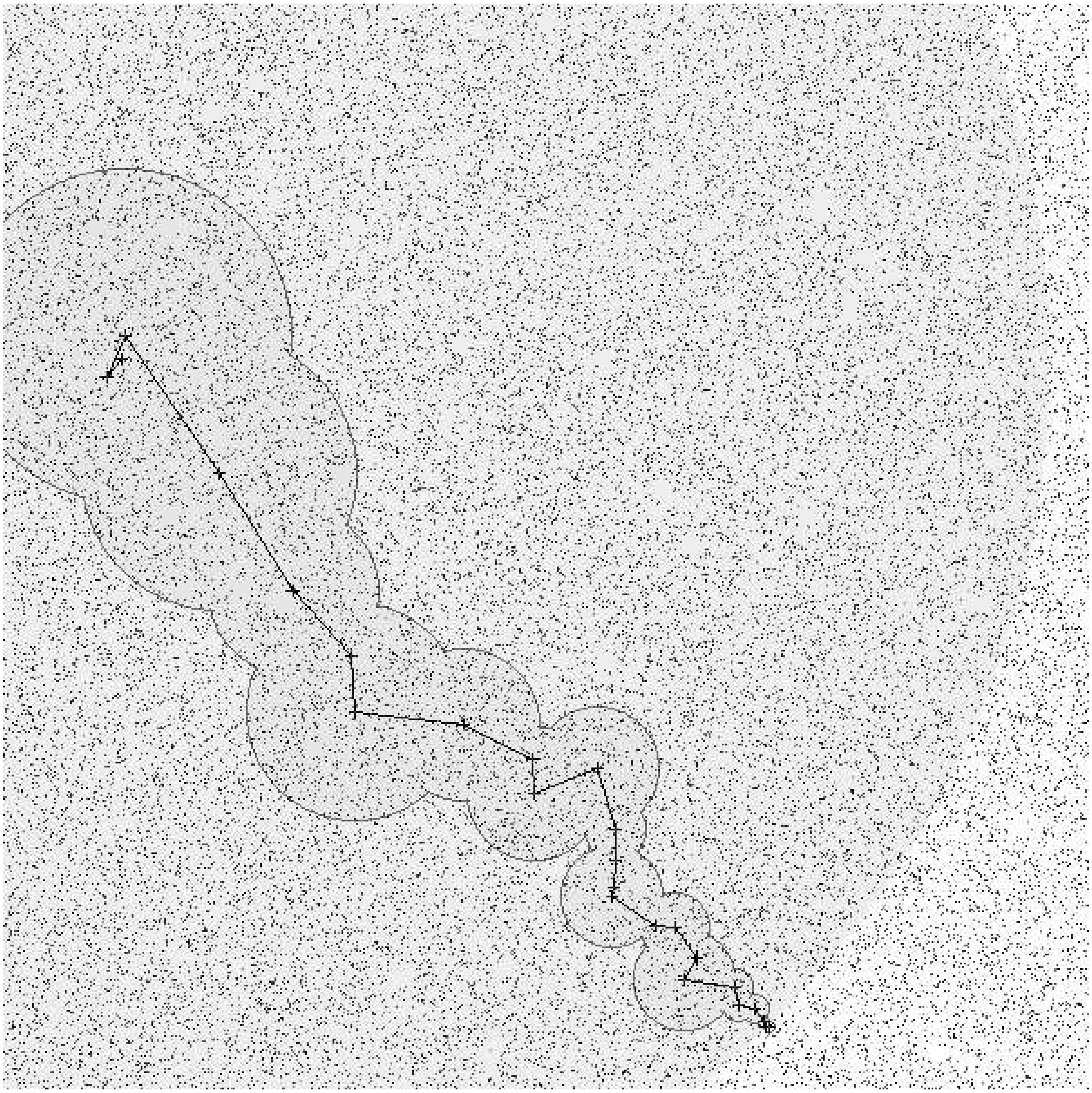

Figure 2: ***An example FRESH route for $N = 32000$ nodes, with a random walk mobility process. The destination is in the lower right side. Anchors are represented as crosses, and the path between anchors is approximated as a straight line for clarity (in reality the route will traverse intermediate nodes between anchors). The darker surface covers the union of the minimal search disks that are necessary at each anchor point to find the nearest node which has encountered the destination more recently. The lighter circle, centered at the source, covers the minimal search disk that would be used in a single-step flood.***

FRESH could be indifferently implemented by a distance-vector or a source routing protocol. Also FRESH is compatible with a variety of flooding approaches.

This is not to say that the performance of such a protocol would be indifferent to the search primitive used. For example, a flood mechanism tailored to FRESH could gain from exploiting the directionality of the route discovery and by avoiding flooding the same nodes in two consecutive searches. However, our objective here is to show that FRESH is highly effective even with a simple flooding technique that is not optimized for the task at hand. The performance results that we present in Section 4 represent a conservative measure in this sense, that can be further improved by using an optimized search.

Despite the generality of Definition 1 we do make two assumptions on the search primitive:

Assumption 1. *The search primitive is omni-directional, that is to say it does not favor any specific direction for finding the required node.*

Assumption 2. *The search proceeds in concentric rings of expanding radius until a node is found which satisfies the boolean condition.*

The first assumption is consistent with the route request phase of existing protocols [13], [8] where the flood advances evenly in all directions. It will be used to motivate a simple search cost metric (Section 4) and leads to a conservative evaluation of our algorithm. Note that having an omni-directional search primitive does not preclude the overall route discovery from being directional; this is indeed the case of FRESH.

We believe that the second assumption is valid for large-scale networks, where the overhead of doing a global flood for each route discovery would be prohibitive. It is also consistent with many existing protocols [13], [8] which either mandate an expanding ring search or offer it as an option. Of course in small networks, unconstrained flooding may be quite acceptable and in such cases there would be no need to use FRESH.

3.2 FRESH Algorithm

We now give a simple formulation of the FRESH algorithm. The following facilities are assumed:

Nodes keep a table of their most recent encounter times with all the nodes they have encountered. This table is queried by calling `prevEncounterAge(NID)`, where `NID` is a unique node identifier, for example the node's IP address. `prevEncounterAge(NID)` returns a scalar representing the time elapsed since `NID` was last a one-hop neighbor, or ∞ if `NID` has never been encountered.

The pseudo-code below invokes the search primitive through an abstract interface which allows a querying node `N` to find the nearest anchor node `A` having seen the destination node `D` more recently than a time `T`. This search is invoked by calling `findNextAnchor(D, T)`, which triggers a network search and returns `A`. In accordance with Definition 1 the search process creates routing state in the network which will allow `N` to subsequently send packets to `A`. This state will be used by the `notifyNextAnchor` call to instruct `A` to pursue the route discovery. More precisely, `notifyNextAnchor(A, D)` will send a packet to `A`, which triggers invocation of the call `FRESH(D)` on node `A`. We note that the packet sent by the `notifyNextAnchor(A, D)` call does not need to carry the time `T` representing the current node's encounter age with `D` since node `A` only needs its own encounter age with `D` in order to iterate the search.

The algorithm, which is run at every node in the network, is as follows:

```
proc FRESH (D) = {
  if (thisnode.ID = D) then {
    replyToSource()
  } else {
      T := prevEncounterAge(D);
      A := findNextAnchor (D, T);
      if (A != D) then
        notifyNextAnchor(A, D);
  }
}
```

3.3 Properties

Having given a precise formulation of FRESH, we can now show two properties of the algorithm.

The first property concerns operation of the algorithm under a special-case mobility process, that is when nodes are static. Since FRESH is dependent upon node mobility so that nodes make encounters and populate their encounter table, this situation can be seen as a degenerate case for the algorithm. Indeed, when nodes are static, encounter tables are empty and FRESH will make a single fresher encounter search, which terminates at the destination. This is a regular single-step route discovery (where s directly searches for d), as done by current on-demand protocols [8] [13].

Property 1. *In the case of static nodes, FRESH reduces to a single-step route discovery.*

Proof. If nodes are static, and thus make no encounters, we have that $\forall i \neq k$, $T_{LE}(i,k) = \infty$, and $\forall i = k$, $T_{LE}(i,k) = 0$. In this case d is the only node satisfying $T_{LE}(d,d) < \infty$ and FRESH terminates after the first fresher encounter search, which is satisfied by the destination. □

The second property shows that fresher encounter searches converge at the destination and therefore provide a loop-free route. Note that guaranteeing convergence does not guarantee that the algorithm will be more efficient than a single-step route discovery.

Property 2. *FRESH is guaranteed to terminate with a loop-free route to the destination as long as the source and destination are part of a connected subset of nodes.*

Proof. Consider a route discovery between a node s and a destination d. Consider the ith iteration, where the algorithm is at node n_i (possibly $i = 1$ in which case $n_1 = s$). By construction, the algorithm will advance to a node n_{i+1} s.t. $T_{LE}(n_{i+1}, d) < T_{LE}(n_i, d)$. Therefore the current encounter age is monotonically decreasing at each step of the algorithm. Since $\forall i \neq k$, $T_{LE}(i,k) > 0$ and $T_{LE}(d,d) = 0$, the encounter age decreases until the destination is reached in a finite number of steps. □

4. PERFORMANCE AND SIMULATION RESULTS

We have performed simulations to verify the scaling performance of FRESH at large network sizes, under different mobility processes. The network sizes ranged from 1000 to 64000 nodes, with either a random walk or a waypoint mobility process.

The simulations used two metrics to evaluate the performance of the protocol: search cost and route quality. In this section we report the results and further discuss two other important aspects of routing performance: proactive overhead and latency.

The *search cost* of a route discovery is the overhead necessary to build the route from a source to a destination. In the case of the on-demand protocols we are considering here, this will be the cost of the search(es) associated with the route discovery. *Route quality* measures the difference between the route obtained by the algorithm and the shortest-hop path.

4.1 Simulation Environment

Our purpose is to evaluate the performance of FRESH in relation only to the mobility process and the size of the network. This way we are sure to that our measures are made in isolation from any outside (positive or negative) interactions such as MAC layer and cross-traffic effects.

The simulations use an idealized model of the MAC layer. Nodes are one-hop neighbors when they come within unit distance of each other, and interferences and collisions are not modeled. We note that this simplification is neutral to the evaluation since we have no cross-traffic (we compute routes sequentially) and since FRESH may run over a variety of MAC implementations.

The topology is a continuous square surface with size chosen to have a node density of 1. In the waypoint mobility model, nodes choose a random target which is uniformly distributed in the surface and advance toward it at constant velocity. When they reach the target, a new target is generated and the node moves again. In the random walk model nodes move at each step in one of the four cardinal directions, and reflect at the boundary.

The simulations run in two phases: warm-up and route computation. In the warm-up phase, nodes move according to the chosen mobility process, populating their tables with the most recent encounter times of each peer node that they encounter. The warm-up phase runs until an encounter ratio of 40% is attained, where the encounter ratio is the proportion of node pairs that have encountered at least once since the beginning of the warm-up.

Once the warm-up is complete, we apply FRESH to sequentially compute a number of routes between randomly chosen source-destination pairs and record the statistics of interest to us. We note that a single route discovery happens on a timescale of tens or hundreds of milliseconds whereas node mobility occurs on a timescale of several seconds or minutes. This allows us to use the approximation that nodes' positions are static for the duration of a route discovery. Of course, a full protocol must be designed to address a possible 'race' condition where a node moves during route discovery and invalidates a prior portion of the route, but this situation is not specific to FRESH and can be corrected by a route maintenance mechanism similar to existing protocols [13], [8].

4.2 Search Cost

We define the cost of a single search which originates at node s and terminates at node f as $C_S(s,f) = (\alpha(|X_s - X_f|))^2$, for some $1 < \alpha < 2$. The cost is quadratic in the distance because (under a uniform node distribution), the number of nodes involved (equivalently, the number of packet transmissions) is proportional to the area of the surface searched. Under the assumption of omni-directional searches (assumption 1), we approximate the surface by a disk centered at the originating node.

The constant α models the fact that the radius of the search disk will on average be larger than the distance between the source and the node which is found. Indeed, since we are considering expanding search disks (assumption 2), it is likely that the search of sufficient radius to locate the requested node will be bigger than the minimal ring size. In the case of a search disk that doubles at each iteration, the worst case would be for $\alpha = 2$, whereas the best case is trivially for $\alpha = 1$. Figure 2 represents the minimal search disks ($\alpha = 1$), both for the FRESH search disks and for the single-step search disk.

A single-step route discovery makes only one search originating at the source and terminating at the destination, and its search cost is

$$C_S^{single-step} = (\alpha|X_s - X_d|)^2 \tag{1}$$

where X_s, X_d are the positions of the source and destination respectively.

A FRESH route discovery, composed of N successive searches has the following search cost cost:

$$C_S = \sum_{i=1}^{i=N} (\alpha|X_i - X_{i+1}|)^2 \tag{2}$$

where X_i is the position of the ith anchor, under the convention that the first anchor is the source and the last anchor is the destination.

Finally the normalized search cost of FRESH is:

$$C_{NS} = \frac{C_S}{C_S^{single-step}} \tag{3}$$

Note that the constant α cancels out in the ratio, which allows us to make a fair comparison between FRESH and a single-step route discovery without making assumptions on the value of α.

Fig. 3 shows the search cost C_S for establishing routes at increasing source-destination distances, using a factor $\alpha = 1.5$ as defined above. Each half-bar has length equal to one standard deviation. The absolute value of the cost does not have a meaningful interpretation, but we can nonetheless observe that the rate of increase of this cost is quite stable for increasing source-destination distances. The random walk mobility process appears to be slightly less favorable to FRESH than the waypoint model.

Fig. 4 shows the normalized search cost C_{NS} of FRESH as defined in (3). We see that the search cost of FRESH converges to approximately 10% of the cost of a direct search. (As stated in Section 3.1 above this is a conservative measure given the sub-optimal search primitive that is assumed in our cost metric.) At smaller source-destination distances, the gain afforded by FRESH is not quite as strong. In the limiting case, when source and destination are one-hop

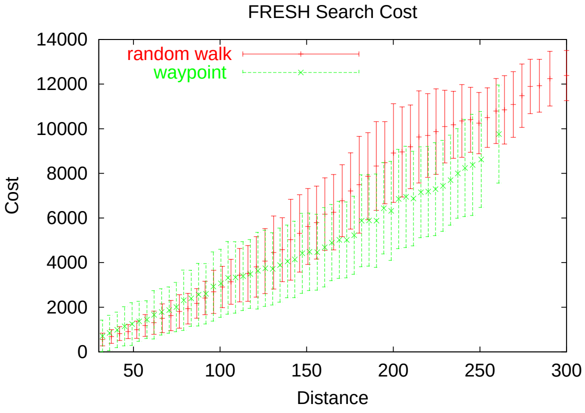

Figure 3: ***Search cost C_S for $N = 64000$ nodes. This is the sum of the search costs of all the fresher encounter searches necessary for a route discovery. Bars represent one standard deviation length on either side.***

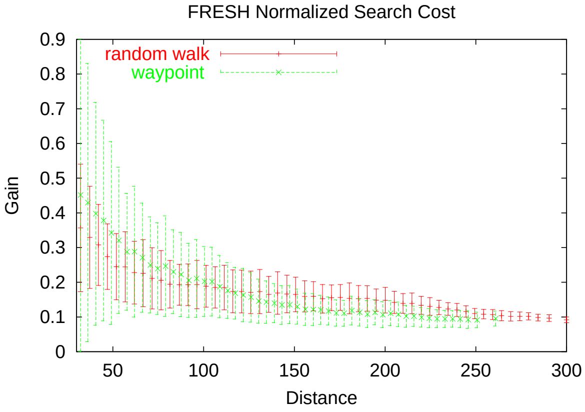

Figure 4: ***Normalized search cost C_{NS} for $N = 64000$ nodes. This is the search cost of FRESH normalized by the cost of a single-stop search. Bars represent one standard deviation length on either side.***

neighbors, route discovery with FRESH will have the same cost as a direct route discovery.

4.3 Route Quality

There is a cost associated with using suboptimal routes, and this must be explicitly recognized when evaluating a routing protocol. In the case of FRESH, it is not guaranteed that the route obtained will be the shortest possible. One may consider that FRESH trades off some amount of route quality in exchange for large reductions in search cost, and so we must be sure that routes remain good enough so that this is worthwhile.

Of course, the relative importance of route quality and search cost depends on traffic statistics. For example, if a flow between two nodes is short-lived and only carries a small number of packets, then search cost dominates the total cost incurred by this flow. On the other hand, if a flow is long-lived and carries a large number of packets, then search cost is insignificant with respect to the route length. (As a side-note we believe that FRESH can be extended to exploit this trade-off, building a cheap but sub-optimal route for low-volume flows, or an expensive but near-optimal route for longer-lived flows).

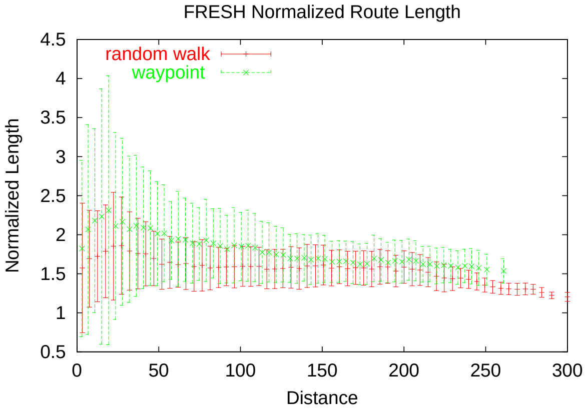

Figure 5: ***Normalized route length L_N, 64000 nodes. Bars represent one standard deviation length on either side.***

Since we do not make any assumptions on the traffic matrix, the purpose of our measurements is not to find the optimal trade-off point between route quality and search cost, but rather to make sure that the routes obtained are at most a small factor longer than the optimal routes, and that this factor does not increase as the source-destination distance increases. We do this by measuring the normalized route length, which is the ratio of the obtained route length and the optimal route length:

$$L_N = \frac{\sum_{i=1}^{i=n-1} |X_i - X_{i+1}|}{|X_1 - X_n|} \qquad (4)$$

Though this ratio measures euclidean route lengths, it should also be indicative of hop lengths, given that the number of hops will be proportional to the distance (assuming a uniform node distribution), and furthermore that the constants involved will cancel out in the ratio.

Fig. 5 shows the route quality L_N as defined in (4). As expected, the routes obtained are longer than the optimal route. However the degradation is reasonable: a factor of about 2.3 in the worst case (waypoint model, very short source-destination distance), decreasing below 1.5 for source-destination pairs which are increasingly separated.

4.4 Proactive Overhead

FRESH requires that nodes keep track of their one-hop neighborhood in order to maintain their encounter tables with up-to-date information. One solution is for nodes to broadcast periodic hello messages in order to inform one-hop neighbors of their presence. These messages are not always necessary. For example, when a node is transmitting data or routing packets, hello messages become redundant since nearby nodes can overhear those packets and infer that the transmitting node is in range. Or, hello messages may be redundant to some MAC layers (as for example Bluetooth [14]) which already do neighbor detection.

More importantly, the per-node proactive overhead remains constant as the network size increases. Also, it can be modulated depending on the expected rate of change of a node's neighborhood and the needs of each node. In the extreme case, a node that is not expecting to receive any traffic need not broadcast any hello messages, since it is not necessary for this node to be present in other nodes' encounter tables. Proactive overhead therefore does not increase with the network size, and can to a large extent be controlled by the protocol designer (unlike the the search cost, which depends on the mobility process). For this reason, and since proactive overhead is not always necessary, we do not present quantitative measures of this overhead.

4.5 Latency

Latency is an important aspect of routing protocol performance. In this paragraph we explain why latency of FRESH is similar to the latency of single-step methods [8] [13]. We consider two types of latency: route establishment latency and round-trip time (RTT) latency.

RTT latency is proportional to the route length (the constants will depend on the MAC layer and on congestions); we have shown above that length of routes obtained by FRESH is quite close to optimal.

Route establishment latency is the time elapsed between the moment when a source requests a route to the destination and the moment when it has a route and may start sending packets. Of course route establishment latency includes RTT latency, since a packet must travel from the source to the destination and back in order to establish the route. However the main source of latency in route establishment will be the time spent doing expanding ring searches. At each expanding ring search, the timeout is usually linear (for example see [13]) in the radius of the search. Considering that the minimal radius of a single-step route discovery search is the distance between source and destination, and considering that the sum of radii of expanding ring searches in a FRESH route discovery is the route length obtained, we see that the route establishment latency of FRESH is also close to the route establishment latency of a single-step approach.

5. FRESH AND MOBILITY

The performance of FRESH is closely related to the mobility process at hand. In this respect it is quite different from existing protocols, which simply consider mobility as a source of 'noise', and whose performance decays as mobility increases. In this section we give some insights on the interactions between FRESH and the mobility process.

Locality and route quality

We say that a mobility process has **locality** over a time-scale t_l when for $t_\triangle \in [0, t_l]$, the position of a node at time $t + t_\triangle$ is correlated with the position at time t. Locality of the mobility process is an important factor in the quality of the routes computed by FRESH. If the mobility process has no locality at all, then the distance between nodes is uncorrelated to their encounter ages, and finding a node with a fresher encounter does not bring us closer to the destination. Some degree of locality is therefore necessary for FRESH to compute good routes.

We illustrate locality with three examples, from strongest to weakest. First, consider the deterministic case where two nodes which have encountered are moving in a straight line, at constant speed. Here, the distance between the nodes is directly proportional to the time elapsed since their encounter. Second, consider two nodes that make independent random walks in the plane. Here, the distance between two nodes is proportional *in expectation* to the square root of their encounter age. Thirdly, consider nodes that make random jumps from one point to another with equal probability (independently of their current position). Then the distance between two nodes is independent of their encounter age and therefore there is no time-distance correlation.

Search cost and route quality

Before considering how mobility affects search cost, it is instructive to observe that route quality and search cost do not necessarily go hand in hand. In other words, the same route can be obtained at low search cost, if the encounter tables of the nodes are such that route discovery is done over many small searches, or it could be obtained at high search cost, if route discovery makes few large searches. Figs. 6 and 7 show two cases where a near-optimal route has been obtained. In the first case FRESH made two large searches, and in the second case FRESH made many smaller searches, resulting in a lower overall search cost. Similarly Figs. 8, 9 show a suboptimal route which is obtained in one case through many small searches, in the second case through few large searches. It is important to realize that which scenario happens is not directly controllable by FRESH; it will depend only on the nodes' encounter tables and hence on the mobility process.

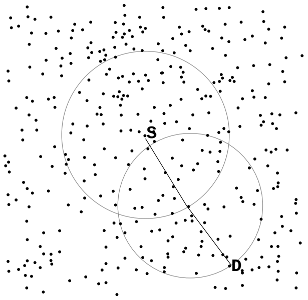

Figure 6: *Expensive search, near-optimal route.*

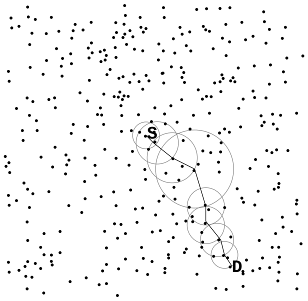

Figure 7: *Cheap search, near-optimal route.*

Locality and search cost

We have seen that *locality* is necessary for FRESH to advance toward the destination at successive iterations, and that the quality of routes obtained is related to this property. We have also illustrated how the search cost is lower when FRESH makes many smaller searches rather than few

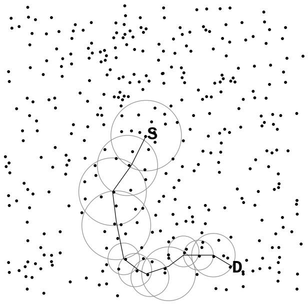

Figure 8: ***Cheap search, sub-optimal route.***

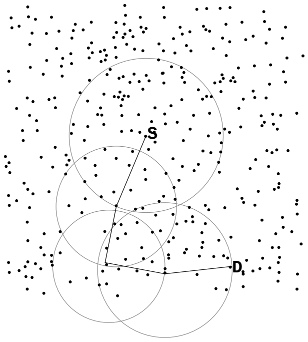

Figure 9: ***Expensive search, sub-optimal route.***

big searches, and that this can happen independently of how direct the route obtained is. We now discuss what properties of the mobility process may result FRESH achieving low search costs. Low search cost is achieved when inter-anchor distances are small, so that FRESH makes many small searches rather than few big ones. Therefore, *diffusion* of node trajectories throughout the surface is necessary so that nodes make a sufficient number of encounters with all other nodes. It is interesting to note that diffusion and locality are opposing properties. For example, in an extreme case of locality where each node is restricted to a small area, node encounters would not diffuse outside of each such area. Or, at the opposite extreme is the random jump process which has no locality but high diffusion. Many common mobility processes (like the random walk or the waypoint model), exhibit intermediate degrees of both properties; as we have seen FRESH works effectively in these situations.

Age gradients

In order to see how distance is related to encounter age, we have plotted the empirical conditional mean of the distance between node pairs, conditional upon their encounter age. Figs. 10 and 11 show this empirical mean for the two mobility processes we have considered, the random walk and the waypoint model, over a square surface of side 32, for $N = 1000$ nodes. Each point in these graphs was computed by considering all the node pairs whose last encounter time is within a certain age interval, and averaging over the distance between these node pairs.

In both cases, we observe that as the encounter age increases, the expected distance converges to a constant which is on the order of a half side of the square surface. In this regime the encounter age carries no information about the two nodes' relative positions. Using the terminology introduced above we see that the waypoint process in this case has locality over a time-scale 20, and the random walk has locality over a time-scale of roughly 250.

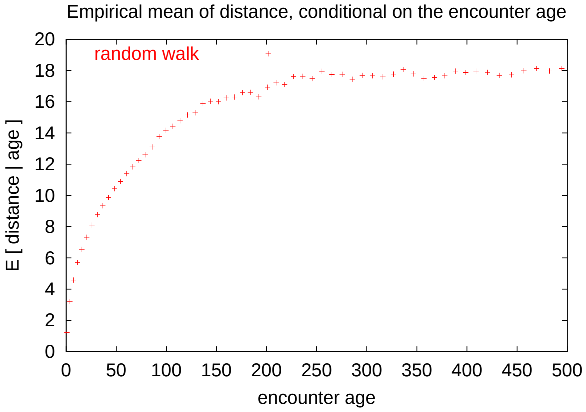

Figure 10: ***Age gradient, random walk. (Empirical conditional mean of distance, conditional on the encounter age).***

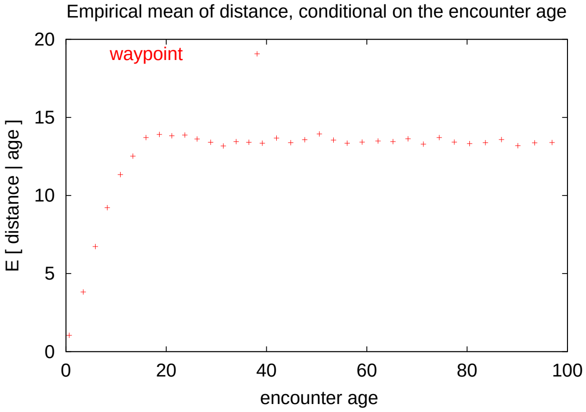

Figure 11: ***Age gradient, waypoint model. (Empirical conditional mean of distance, conditional on the encounter age).***

Another observation is that convergence occurs earlier for the waypoint model than for random walk, indicating that the waypoint process reaches its stationary distribution more rapidly. This happens because the waypoint mobility process chooses a new target at random each time it attains the previous one. Therefore once a node moves toward its second target, its position is already independent of its starting point. And given a constant unit speed, and a square of side 32 , the first target is reached an average after 15 to 20 time units, corresponding to the point (at age approximately 20) in Fig. 11 where the function flattens out. In the case of the random walk, a node takes on the order of t^2 iterations to traverse a distance t, and the stationary distribution is attained after a node has reflected one or more times off the border, which means traversing a distance on the order of a half-side. This corresponds to an age on the order of 225 (15^2), which is confirmed by Fig. 11.

In both cases, we see that once the stationary regime is reached, the empirical mean of the distance between two nodes is constant, and therefore does not vary with the encounter age. This indicates that in the first iterations of FRESH, if two successive iterations are in this range, then the route will not be progressing toward the destination. This is in fact what happens in the route shown in Fig. 2, where we see that the first two hops appear to be going in independent, random directions. After these first two hops we reach a node whose encounter age lies within to the descending area of the age gradient, and we see that the route makes good progress from there onward.

6. CONCLUSIONS AND FUTURE WORK

We have introduced an algorithm for efficient route discovery in mobile ad hoc networks that uses iterated fresher encounter searches. A novel aspect of this algorithm is that it *takes advantage* of the fact that nodes are moving. Compared to geographic algorithms, an advantage of our proposal is that it does not assume any hardware add-ons such as GPS receivers. In formulating the algorithm we have introduced a general definition of a *search primitive* to show that FRESH can be used over a variety of flooding and routing techniques.

Though this paper has focused on the application of routing between peer nodes, we believe that FRESH will have other applications in ad hoc networks. For example, assuming an ad hoc network which has one or more gateways to the wired internet, FRESH could be used by a mobile node to establish a route to the nearest gateway.

Under a conservative search cost metric, where we assume a naive search strategy, our simulations indicate that the algorithm reduces the flood overhead by an order of magnitude in large networks. This is significant since route discovery is a major source of overhead in ad hoc routing protocols. We believe that this route discovery algorithm may therefore be a useful component in designing routing protocols that scale to larger numbers of nodes. The search cost will be further reduced with an enhanced search strategy which could for example exploit the directionality of sequential searches. This will be a topic for further investigation.

As part of our future work we intend to develop a full routing protocol incorporating the ideas described in this paper. One topic that will deserve further attention is the possibility to trade off better routes in exchange for a higher search cost, (alternatively to trade off a sub-optimal route for a lower search cost) by recursively applying FRESH to interior portions of the route. This trade-off deserves to be adjustable dynamically, since the optimal point will vary widely depending on the duration of a connection.

7. REFERENCES

[1] George Aggelou and Rahim Tafazolli. RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks. In *Proceedings of the Second ACM International Workshop on Wireless Mobile Multimedia*, Seattle, WA, August 1999.

[2] R. Castaneda and S. Das. Query localization techniques for on-demand routing protocols in ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 1999.

[3] C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks. In *Proceedings of the IEEE Singapore International Conference on Networks*, Singapore, April 1997.

[4] Samir Ranjan Das, Charles E. Perkins, and Elizabeth E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, March 2000.

[5] Matthias Grossglauser and Martin Vetterli. Locating nodes with EASE: Last Encounter Routing for Ad Hoc Networks through Mobility Diffusion. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, April 2003.

[6] Zygmunt J. Haas, Marc R. Pearlman, and Prince Samar. The interzone routing protocol (ierp) for ad hoc networks. IETF Internet-Draft, draft-ietf-manet-zone-ierp-02.txt, July 2002.

[7] Zygmunt J. Haas, Marc R. Pearlman, and Prince Samar. The intrazone routing protocol (iarp) for ad hoc networks. IETF Internet-Draft, draft-ietf-manet-zone-iarp-02.txt, 2002.

[8] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[9] Sung-Ju Lee, Elizabeth M. Belding-Royer, and Charles E. Perkins. Scalability study of the ad hoc on-demand distance vector routing protocol. In *International Journal on Network Management (IJNM)*, to appear, 2003.

[10] L. Li, J. Halpern, and Z. Haas. Gossip-based ad hoc routing. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, New York, NY, June 2002.

[11] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Kobe, Japan, April 1997.

[12] Z. J. Haas M. R. Pearlman and T. Samar. Zone routing protocol (zrp). IETF, Internet-Draft, 2002.

[13] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999.

[14] Bluetooth SIG. Bluetooth v1.1 specification. http://www.bluetooth.org, 2000.

[15] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2002.