Robust Routing for Dynamic Wireless Networks Based on Stable Embeddings

Dominique Tschopp, Suhas Diggavi, and Matthias Grossglauser
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland
Email: (first_name.last_name)@epfl.ch

Jörg Widmer
DoCoMo Euro-Labs
Landsberger Str. 312
80687 Munich, Germany
Email: (last_name)@docomolab-euro.com

***Abstract*— Routing packets is a central function of multi-hop wireless networks. Traditionally, there have been two paradigms for routing, either based on the geographical coordinates of the nodes (geographic routing), or based on the connectivity graph (topology-based routing). The former implicitly assumes that geometry determines connectivity, whereas the latter does not exploit this inherent geometry of wireless networks, and assumes a general graph instead.**

In this paper, we explore ideas that attempt to bridge these two paradigms. We do so by investigating routing techniques based on metric embeddings of the connectivity graph. If this graph is closely related to the underlying geometry of the nodes, then it is possible to embed the graph in a low-dimensional normed space. This keeps the overhead of the routing protocol low.

We specifically explore embeddings of dynamic networks induced by channel fading and mobility. This motivates the novel problem of stable embeddings, where the additional goal is to maintain an embedding over time, such that the evolution of the embedding faithfully captures the evolution of the underlying graph itself. This is crucial to limit the control overhead of the routing protocol, and to ensure that our approach is scalable.

I. Introduction

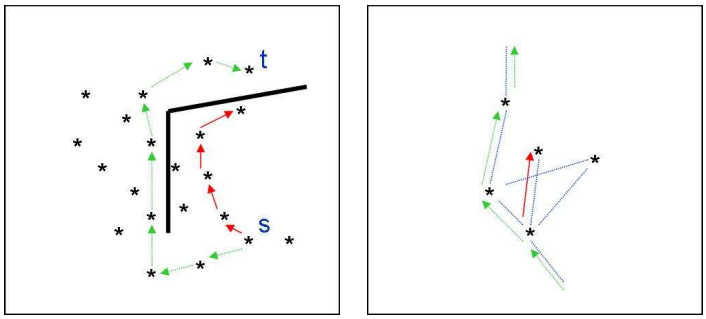
Ad hoc networks comprise potentially mobile wireless nodes, such as sensors, laptops, cell phones, or PDAs. In practice, such networks could be deployed for instance for environmental monitoring, when wired networks are hard and costly to put in place, for disaster recovery if the existing infrastructure was destroyed, for interactive gaming, or to increase the coverage of base stations in cellular networks. Nodes can only communicate directly over short distances because of power limitations and interferences on the wireless channel. To enable long range communications, nodes act both as terminals and relays, *i.e.*, nodes far apart communicate over multiple hops. Note that this communication model is not information-theoretically optimal [1], but captures the current technology (similar to the communication model used in [2]). Our interest in this paper is in dynamic wireless networks, where the connectivity of the nodes is random and may evolve over time.

In such an environment, paths between pairs of nodes change frequently, which makes routing a challenging problem. In topology-based routing protocols such as [3], [4], nodes discover the shortest path to each other through flooding. Once a path is discovered, the path information is either carried directly in the packet header for source routing, or kept on the nodes in a table containing next hops and distances to other nodes. However, end-to-end paths are very fragile, because the disruption of an intermediate link means that the path needs to be reestablished or repaired. The latter problem limits the applicability of such routing algorithms to relatively small and stable ad hoc networks. Indeed, the control overhead involved to maintain such routing tables is not sustainable in large, dynamic networks.

Another way to route in networks is to first assign coordinates to nodes, and then to make purely local routing decisions based on this coordinate system. More precisely, a node forwards a packet to its neighbor positioned closest to the destination. Here we assume that the position of the destination is known. With this strategy, all forwarding decisions are local, and there is no notion of an end-to-end route that breaks if intermediate nodes move.

This immediately raises the question of how nodes can obtain their coordinates. If nodes are equipped with a Global Positioning System (GPS), one can use these "physical" geographical coordinates for routing. Unfortunately, physical coordinates might not capture the topology of the network well. On a large scale, obstacles such as walls, buildings, mountains, etc. could make the communication path very different from a straight line between the source and destination, giving rise to dead-ends in greedy routing. A routing dead-end occurs when a node has no neighbor closer to the destination than itself. In Fig. 1(a) we illustrate such a dead-end due to a large topological void. When such a situation occurs, that node needs to start a recovery procedure. One simple option is to initiate scoped floods in order to identify a suitable next hop. The control overhead to find a next hop with such recovery procedure is usually high. Thus, such dead-ends should be avoided if possible. Furthermore, packets could even get stuck locally due to the random nature of the channel (see Fig. 1(b)).

In this paper we further explore the idea studied in [5] to "embed" the connectivity graph in order to obtain coordinates which reflect the structure of the network well. We can then use the embedded coordinates for routing. This approach therefore bridges geographic and topological routing. The embedding attempts to preserve the graph distances, so that nearby nodes are usually embedded close to each other and



(a) Large scale void (b) Local dead-end

Fig. 1. In 1(a), packets sent from the source node s to the target node t get stuck because of the large wall when a greedy forwarding strategy is chosen (red solid arrows). A better choice would have been to follow the path along the dashed green arrows. As shown in 1(b), on a local scale the randomness of the channel can also lead to dead-ends (red arrow), even though these impact the performance of routing less severely as it is easier to recover from them.

nodes which are many hops apart are embedded far apart. If the embedding captures the network topology well, dead-ends can only be local and do not lead to a severe degradation of the performance. It is obviously desirable that the control overhead per node involved to embed the connectivity graph is low, and ideally independent of the scale of the network.

The paper is organized as follows. In Section II, we formally state the problem and the models used in the paper. We also give the background on the ideas explored in [5]. In Section III, we make some observations on the structure of wireless connectivity graphs. We present the design criterion and a gradient descent algorithm for embedding in Section IV. We combine routing and collection of the embedded coordinates in Section V. We finally conclude with a short discussion in Section VI.

II. Problem Statement and Related Work

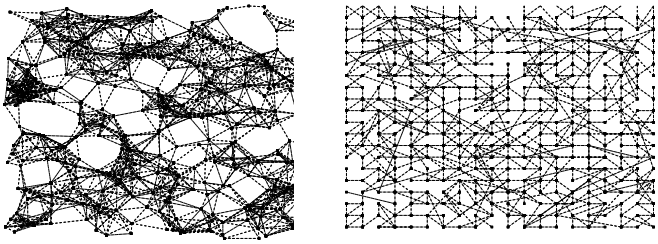
In this section we present models of wireless ad hoc networks, and we relate this paper to existing work.

A. Problem Statement

We model ad hoc networks as unweighted dynamic graphs. Nodes are associated with vertices, and there exists an edge between two vertices if the corresponding nodes can communicate directly with each other over a wireless communication channel. The connectivity depends both on the physical positions of nodes as well as on the channel model. The dynamics result from the mobility of the nodes and from the randomness of the wireless medium. This level of abstraction is sufficient to give us a thorough understanding of the characteristics of ad hoc networks, and in particular, of the applicability of embedding techniques for routing. We consider two particular models.

In the first model, we define a connectivity graph $\mathcal{G}(N, r, p)$, where N is the number of nodes, r is the communication radius, and p is a connection probability. A snapshot of a $\mathcal{G}(N, r, p)$ is shown in Fig. 2(a). First, every node i is placed uniformly at random at a position x_i on the unit square. Then, we add an edge between nodes i and j with probability p if $||x_i - x_j|| \leq r$ for all i, j. In this model the underlying assumption is that nodes can only communicate directly if they are located physically close to each other, and that even in this case, some losses can occur because of the random nature of wireless communications. Note that this model is closely related to the geometric random graph model studied in the literature [2].

In the second model, we define a connectivity graph $\mathcal{H}(N, S, \lambda, r)$, where again N is the number of nodes, S^2 is the number of locations nodes can occupy, λ is the number of neighbors a node can pick, and r is the expected communication radius. A snapshot of a $\mathcal{H}(N, S, \lambda, r)$ is shown in Fig. 2(b). Every node randomly picks a location on a $S \times S$ grid of locations. Every location has a set of channels associated with it, which are drawn a priori. More precisely, for every location we draw λ communication links. The probability of being connected to another location decays exponentially with mean r with the distance (see [6] for a motivation of this model), while the angle is chosen uniformly at random between 0 and 2π. Note that the degree of a location can be different from λ, as the same location might be picked twice and links added when a node is chosen by node or when a node chooses another node. In this model, we focus on long term fading. The assumption is that communication links depend on the environment (*e.g.*, buildings, walls, mountains etc.) and that if two nodes are at the same physical positions they inherit the same channels. However, with a low probability, nodes far apart can be connected.



(a) $\mathcal{G}(400, \frac{4}{\sqrt{1000\pi}}, 0.8)$ (b) $\mathcal{H}(400, 20, 5, 1)$

Fig. 2. In Fig. 2(a), we show an aerial view of a $\mathcal{G}(400, \frac{4}{\sqrt{400\pi}}, 0.8)$, while in Fig. 2(b) we exhibit an aerial view of a $\mathcal{H}(400, 20, 5, 1)$. Note that the connectivity is strictly local in the $G(N, r, p)$, while long links can occur in the $\mathcal{H}(N, S, \lambda, r)$

Nodes move according to a random walk or a random waypoint model. When the second connectivity graph is used, the positions of the nodes are rounded to the closest grid point, and nodes inherit the channels of the corresponding location.

The problem of routing amounts to finding a path between a source node and a destination node in such a connectivity graph. Obviously, we would like this path to be as short as possible in order to minimize the network resources involved in transporting the message. The main cost involved in routing is the communication overhead to find and maintain such paths. In this paper, we are focusing on the characteristics

of such connectivity graphs and explain why they are well embeddable even under mobility. In particular, we highlight the inherently low dimensional structure of these graphs and their relatively slowly evolving nature. We then design an embedding algorithm which exploits these specificities to maintain a stable embedding with low distortion at a low overhead rate per node. We also present local mechanisms which in turn enhance the routability of such low-distortion embeddings.

B. Related Work

This paper relates directly to [5]. In [5] we introduced probabilistic beaconing (PB), a novel embedding algorithm tailored to dynamic connectivity graphs defined by mobile wireless ad hoc networks. The design of the algorithm exploits the fact that connectivity and mobility are local, and that at large scale, geometry plays a role. In the proposed approach, the overhead per node to build the embedding is independent of the number of nodes and remains constant when we increase the size of the network. Further, a sliding window mechanism is introduced, which helps maintaining a stable embedding in the face of mobility and channel uncertainty. The simulation results show that this approach outperforms other embedding schemes developed for such networks both in terms of packet delivery ratio and in terms of overhead to maintain the embedding. It is also shown that embedding large scale distances is easier than to embed small scale distances. This is problematic in the sense that forwarding decisions are local. In order to by-pass this issue, we propose the use of local routing tables to "see further" ahead and consequently avoid this local uncertainty area. Alternatively, we propose a randomized forwarding scheme to achieve the same purpose. We refer the reader to [5] and references therein for a survey of the related work, notably on the specifics of geographic routing. In this paper, we clarify, generalize, and extend the notions presented in [5]. In particular, we make observations on why the proposed embedding algorithm is efficient, and we provide a better understanding of the underlying optimization problem. Several other papers in the literature study the use of virtual coordinates or embeddings for *static* wireless networks (see references in [5] for more details).

III. Observations on Wireless Connectivity Graph

The formal problem of representing a connectivity graph in a metric space is called *graph embedding*, and has received significant recent attention in the computer science literature (see for example [7] and references therein). The goal of the general graph embedding problem is to take as an input a graph with n vertices $\mathcal{V} = \{v_1, \ldots, v_n\}$, and a metric $D(x, y)$ between vertices $x, y \in \mathcal{V}$. The output is then an embedding function $f : \mathcal{V} \rightarrow X^{'}$ where $X^{'}$ is a metric space endowed with metric $D^{'}(\cdot, \cdot)$. The goal of designing f is that the distances of the vertices in the metric space are "close" to the distances on the original graph. That is, we want to ensure that there exist constants c, r such that $x, y \in \mathcal{V}$, $rD(x, y) \leq D^{'}(f(x), f(y)) \leq crD(x, y)$. For this paper, we focus our attention to the case when $X^{'}$ is the normed space R^m, *i.e.,* a m-dimensional Euclidean space. The metric used is $D^{'}(x', y') = ||x' - y'||$ for the norm defined on R^m. The goal of the embedding is to compactly (and as faithfully as possible) represent the graph in an alternate, hopefully low-dimensional space. More concretely, in our problem, we use the shortest path distances for the metric on the connectivity graph. We want to find an embedding onto R^m, for small m, such that for $x, y \in \mathcal{V}$, $rD(x, y) \leq ||f(x) - f(y)||_2 \leq crD(x, y)$. The norm used in our numerical results is the 2-norm.

For arbitrary graphs, it is known that one cannot hope to get good low-distortion and low-dimensional embeddings [7]. However, we believe that connectivity graphs that arise from wireless networks are special, because there is some underlying geometry associated with such graphs. On the one hand, channel fading might destroy some local geometry associated with connectivity. On the other hand, nodes originate from a 2-dimensional world, and connectivity is mostly local. Hence, we suspect that geometry will play a large role in connectivity over larger distances. That is, nodes positioned far apart in the real world are bound to communicate over multiple hops. Therefore, even though one could still construct wireless connectivity graphs which do not have such properties, our experiments and intuition suggest that such configurations occur relatively rarely.

In order to understand some of these issues, we examine the problem using *multi-dimensional scaling* (MDS) which is another technique for extracting coordinates from pairwise distances widely used in statistics literature [8]. Classical scaling, [8], is a technique used in statistics to obtain (embedded) coordinates $\mathbf{X}$ in Euclidean space of n points given only a matrix of pairwise distances. Coordinates obtained with this approach minimize the sum of squared errors between the original distances and the Euclidean distances between points (which is also called the *stress function* [8]), *i.e.,*

$$\min_f \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left[D(v_i, v_j) - D^{'}(f(v_i), f(v_j)) \right]^2 .$$

Therefore, the criterion used to evaluate the embedding in MDS is different from that of the relative distortion criterion introduced earlier. The stretch criterion used in computer science [7] is a worst case distortion, in contrast to the average distortion used in MDS.

A nice property of $\mathbf{X}$ is that it is a principal axes solution, *i.e.,* the variance along axes is maximized [8]. We are interested in this variance as we want to show that the error incurred by representing wireless connectivity graphs in low-dimensional Euclidean space is small. In other words, in an optimal least square embedding in Euclidean space, only a small number of dimensions are necessary to capture most of the variance.

Given a $n \times n$ matrix $\mathbf{D}^{(2)}$ where the (i, j)-th entry in $\mathbf{D}^{(2)}$ is given by $D^2(v_i, v_j)$, for $v_i, v_j \in \mathcal{V}$, we define a matrix $\mathbf{B}$ as,

$$\mathbf{B} = -\frac{1}{2}\mathbf{J}\mathbf{D}^{(2)}\mathbf{J}, \tag{1}$$

where $\mathbf{J} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^{\mathbf{T}}$ and $\mathbf{1} = [1, \ldots, 1]^T$. The m-dimensional coordinate matrix of classical scaling is given by $\mathbf{X} = \mathbf{Q}_+ \Lambda_+^{\frac{1}{2}}$, where Λ_+ is the matrix containing the m largest eigenvalues and $\mathbf{Q}_+$ the corresponding eigenvectors. Note that dimensions are nested so that the $m-1$ first dimensions of a m-dimensional embedding are the same as the $m-1$ dimensions of an $m-1$-dimensional embedding

In Figure 3 we show how geometry plays a role in the dimensionality of connectivity graphs. We generate[1] several $\mathcal{G}(2000, r, p)$ with varying average node degrees and for $p = 0.6$ and $p = 1$ (see Fig. 3(a) and 3(b) respectively). Note that since the node density is $\frac{N}{1} = N$, the average node degree is $p(N-1)\pi r^2$. We then use (1) to find the m-dimensional coordinate matrix associated with this topology. Assume that the eigenvalues $[\theta_1, \theta_2, ...]$ in Λ are in decreasing order. In particular, we are interested in the spectrum of the re-centered squared distance matrix $\mathbf{B}$ and how variance is distributed in the different dimensions. In Fig. 3, we plot $v(d) = \frac{\sum_{i=1}^{d} \theta_i}{\sum_{i=1}^{D} \theta_i}$, where D is such that θ_D is the smallest positive eigenvalue. This random experiment is repeated, and the cumulative results are presented in Figure 3. It can be observed that there is a considerable gap between the first and second dimensions, while there are only small increments for subsequent dimensions. This indicates that most of the variance is captured by the two first dimensions. Further, this gap diminishes as we increase the communication radius. These experiments seem to suggest that wireless connectivity graphs are well representable using a small number of dimensions. Not surprisingly, when we increase the average degree and consequently the communication radius, geometry progressively plays a smaller role and the fraction of the total variance in the two first dimensions is reduced. For a very large average degree, the variance appears to be equally distributed in all dimensions. Interestingly, when we make connectivity random by setting $p = 0.6$, we reduce the variance in the two first dimensions. Intuitively, adding complexity to the channel also decreases the embeddability of the connectivity graph in a low dimensional space.

In Fig. 4, we show that remarkably this low dimensionality is always present, independently of the size of the network. This suggests that a smart design for a distributed embedding algorithm should exploit this property and be highly scalable. In other words, it seems that the communication overhead necessary to capture this low dimensional structure should be of $O(1)$ per node, as adding new nodes does not change the properties of the connectivity graph.

A. Stability of Embeddings

One of the main concerns in this paper is to handle *dynamic* wireless networks, where the connectivity changes over time either due to node mobility or channel fading. In this case, an important concern is how to find out the path to (or the location of) a destination. The traditional method (as explained in Section V) is to use a separate service to handle these updates

[1]See Section II-A for notation.

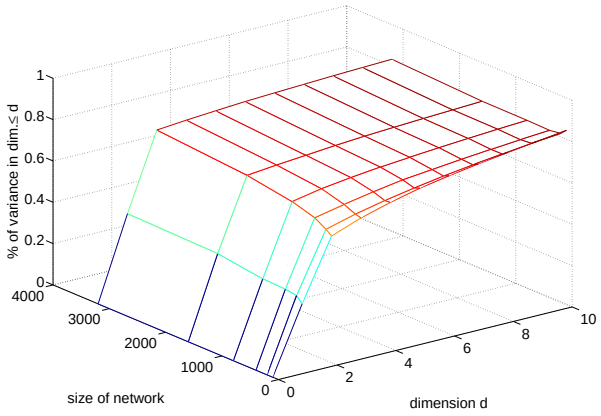


Fig. 4. Cumulative distribution of variance in dimensions for $G(N, r, 1)$. The average degree (deg) is 16 ($r = \sqrt{\frac{deg}{p(N-1)\pi}}$) and probability $p = 1$ of link connectivity. We vary the networks size N. One can observe that the low dimensional structure of wireless connectivity graphs is independent of N

and distribute the information. If one is counting control traffic overhead, this would also need to be taken into account. In the presence of significant mobility, such updates might be delayed, and hence we may only have access to "outdated" information about a destination node. In order to understand how much such a delay would affect our techniques, we define the notion of *stability* of the embeddings.

Suppose the connectivity graph is time-varying, with the graph at time t given by $\mathcal{G}_t$, and the distances by $d_t(v_i, v_j) = D_{\mathcal{G}_t}(v_i, v_j), v_i, v_j \in \mathcal{V}$, where $= D_{\mathcal{G}_t}(\cdot, \cdot)$ is the shortest-path distance between v_i, v_j in the graph $\mathcal{G}_t$. We define an embedding for each time t through the mapping $f_t : \mathcal{V} \to X^{'}$, which tries to capture the distances in the graph $\mathcal{G}_t$. Let us define $d_t^{'}(v_i, v_j) = D^{'}(f_t(v_i), f_t(v_j)), v_i, v_j \in \mathcal{V}$ as the distance between the vertices in the embedded space at time t. If we have access to outdated information about the embedded coordinates of vertex v_j from time $t - \Delta$, we would only be able to compute $D^{'}(f_t(v_i), f_{t-\Delta}(v_j))$.

A notion of stability can be defined through the *stretch* between $d_t^{'}(v_i, v_j)$ and $D^{'}(f_t(v_i), f_{t-\Delta}(v_j))$. Let

$$g(v) \stackrel{def}{=} \max\{v, \frac{1}{v}\}$$

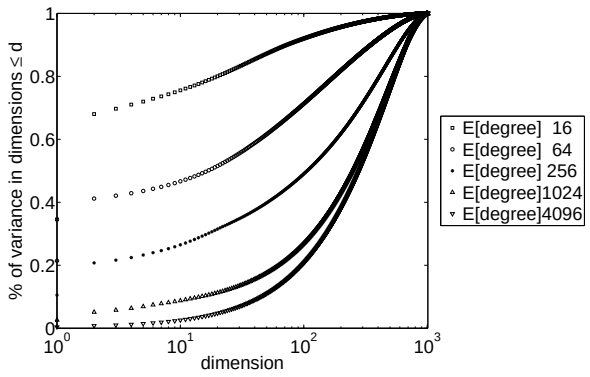
Then we define stretch $\mathcal{S}(\Delta, i, j)$ (distortion) when we examine an outdated destination position as

$$\mathcal{S}(\Delta, i, j) = g\left(\frac{D'(f_t(v_i), f_t(v_j))}{D'(f_t(v_i), f_{t-\Delta}(v_j))}\right)$$

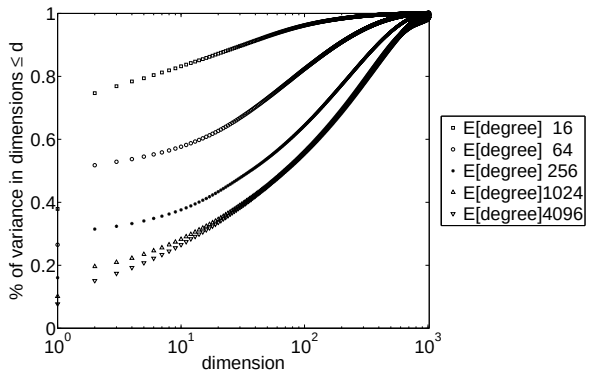
In order to isolate the effect of topology changes and the embedding errors we examine just the changes in the embedded distances without using outdated information through $\tilde{\mathcal{S}}(i, j)$

$$\tilde{\mathcal{S}}(i, j) = g\left(\frac{d_t(v_i, v_j)}{D'(f_t(v_i), f_t(v_j))}\right)$$

for the updated embedding. We have suppressed the dependence on time t since we expect this to be stationary, *i.e.*, not to depend on the time-instant chosen.



(a) Spectrum with $p = 0.6$



(b) Spectrum with $p = 1$

Fig. 3. Cumulative distribution of variance in dimensions for $G(2000, r, p)$ for various average degree and probability $p = 0.6$ and $p = 1$ of link connectivity.

In Fig. 5, we examine[2] the stability of an $\mathcal{H}(1500, 30, 10, 1.5)$. We use probabilistic beaconing with a square window of length 10, and embed in 20 dimensions. Nodes move according to the random walk model with speed 1. It is interesting that the mean distortion to an outdated

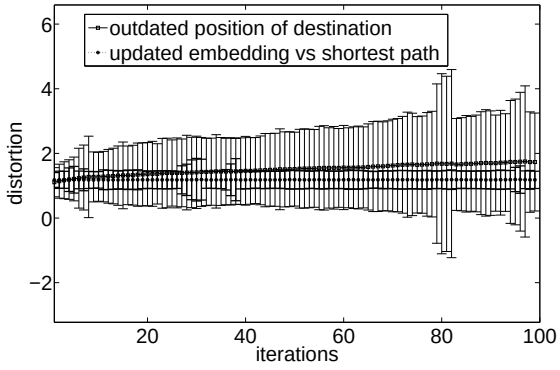


Fig. 5. Stability of the PB embedding of a $\mathcal{H}(1500, 30, 10, 1.5)$ subject to random walk mobility of nodes.

embedded position grows very slowly and sublinearly with time while the standard deviation grows linearly with time. Hence, even when we update the embedding in a lazy way, the embedding remains a good approximation of the underlying communication graph for several iterations.

There can be several alternate notions of stability. For example, we might want that when there are small changes in the embedded distance between times $t - \Delta$ and t, the difference between $D^{'}(f_t(v_i), f_{t-\Delta}(v_j))$ and the correct embedded distance $d_t^{'}(v_i, v_j)$ to be small as well. That is, we would like $|d_t^{'}(v_i, v_j) - D^{'}(f_t(v_i), f_{t-\Delta}(v_j))| \leq \gamma |d_t^{'}(v_i, v_j) - d_{t-\Delta}^{'}(v_i, v_j)|$, for some constant γ. In order to understand the overall impact of outdated information, we might also look for another characterization. For small changes in the graph distance between times $t - \Delta$ and t, we would like the difference between $D^{'}(f_t(v_i), f_{t-\Delta}(v_j))$ and the correct embedded distance $d_t^{'}(v_i, v_j)$ to be small as well. That is, we would like $|d_t^{'}(v_i, v_j) - D^{'}(f_t(v_i), f_{t-\Delta}(v_j))| \leq \delta |d_t(v_i, v_j) - d_{t-\Delta}(v_i, v_j)|$, for some constant δ. This captures both the embedding error as well as the changes in topology. These forms might be useful in other contexts of embedding dynamic graphs.

[2]See Section II-A for notation.

IV. Probabilistic Beaconing

As seen in Section III, if the distances between vertices on a graph are known, then one can use many embedding techniques in order to find $f(\cdot)$ and hence the coordinates for the vertices $\mathcal{V}$ in $X^{'}$. However, one of the reasons to attempt embedding is to reduce the amount of control traffic being sent over the network. Therefore, we would like only a constant *overhead rate per-node* in the network, which means that the control traffic rate can only grow linearly in the network size. Therefore, any technique that attempts to discover the distances between every pair of nodes in the network would entail $\Omega(n^2)$ rate and hence a growing overhead. In order to get a constant overhead, we develop a method where only a constant number of nodes act as beacons at any time, and thereby at every time we only know the exact distances of all the nodes in the network to this small set of beacons. Hence at every time step k, a new beacon node B_k is randomly selected. This beacon floods the network with a control message, through which each node learns its shortest-path distance in the connectivity graph to this beacon. We need to infer or estimate the other distances only from these measurements to a constant number of such beacons.

We do this inference by utilizing the coordinates of the neighborhood of each node in the wireless network. The intuition behind this is that the embedded coordinates of each

node should be close to that of its neighbors. Therefore, we do a local averaging step which moves the hop-distance of a node $v_i \in \mathcal{V}$ to a beacon B_u towards the average of its neighborhood (see Algorithm 1). We then iteratively solve a problem which tries to minimize an appropriately defined least-squares criterion to reconcile the embedded coordinates of the vertex and those of the beacons. An alternate method could have been to choose to do local averaging on the coordinates $x_i \in X^{'}$ of a node $v_i \in \mathcal{V}$ by moving towards the center of its neighborhood $\mathcal{N}_i$.

Working with average hop counts ensures that nodes in the same neighborhood do not get embedded to distant locations. Additionally, it also tends to place a node between its neighbors. This property is useful for greedy routing as being placed in the center of its neighborhood augments the chances of finding next hops in every directions, and consequently reduces the risk of dead-ends. Another reason to work with average hop counts is that the distance measurements can be noisy. This is particularly true in mobile environments or in environments where channels are fluctuating rapidly. In this perspective, a node v_i can view its outdated distance measurement $P_i(k)$ and the outdated distance measurements $P_j(k)$, for $v_j \in \mathcal{N}_i$, to a beacon B_u as a series of noisy distance measurements. Given these noisy measurements, a node can now try to estimate its hop distance to a beacon by the average of the distance measurements of its neighborhood as,

$$\hat{h}_u(k) = \frac{1}{|\mathcal{N}_j + 1|}\left[\left(\sum_{v_j \in \mathcal{N}_i} P_j(k)\right) + P_i(k)\right] \qquad (2)$$

In particular, given this average hop-count $\hat{h}_u(k)$ of its neighborhood, node v_i attempts to solve the following least-squares problem:

$$\min_{\vec{x}} \sum_{u=0}^{B-1} [\hat{h}_u(k) - ||\vec{x} - \vec{b}_u||]^2 \stackrel{def}{=} \min_{\vec{x}} h(\vec{x}), \qquad (3)$$

where $b_u = x_{B_{k-u}}^{(k-u)}$ is the coordinate of the window of beacons under consideration. Therefore, the gradient of $h(\vec{x})$ with respect to $\vec{x}$ is $2\sum_{u=1}^{B} \frac{\vec{x}-b_u}{||\vec{x}-b_u||}[||\vec{x} - b_u|| - \hat{h}_u]$. Hence, the iterative method proposed in Algorithm 1 is therefore just a gradient-descent method. This iterative method goes to a local minimum and therefore converges. This iterative gradient-descent technique is illustrated in Figure 6. Instead of stopping after a fixed number of steps W, we can also choose a stopping criterion the such that the gradient is small enough.

There are flavors of the gradient descent algorithm which take a variable step-size instead of the constant step size chosen in Algorithm 1. Such a variable step size could depend on the gradient itself, taking larger steps when the gradient is large and smaller ones when the gradient is small to ensure a smoother descent. One can incorporate the many ideas used in gradient-descent techniques into this problem.

Algorithm 1 Probabilistic beaconing

1 At time k calculate distance $P_j(k)$ of vertex v_j to beacon B_k
2. Adjust distances to beacons,
averaged over one-hop neighborhood
For $u = 0$ to $b - 1$
Set $\hat{h}_u(k) = \frac{1}{|\mathcal{N}_i|+1}\left(\sum_{j\in\mathcal{N}_i} P_j(k-u) + P_i(k-u)\right)$
end
3. Local optimization
Starting point is center of mass of node i + neighbors
$x := \frac{1}{|\mathcal{N}_i|+1}\left(\sum_{j\in\mathcal{N}_i} x_j^{(k)} + x_i^{(k)}\right)$
Repeat W times
$x := \frac{1}{b}\sum_{u=0}^{b-1} x + \left(\hat{h}_u(k) - ||x - x_{B_{k-u}}^{(k-u)}||\right)\frac{x - x_{B_{k-u}}^{(k-u)}}{||x - x_{B_{k-u}}^{(k-u)}||}$
end
4. Update position
Set $x_i^{(k+1)} := x$

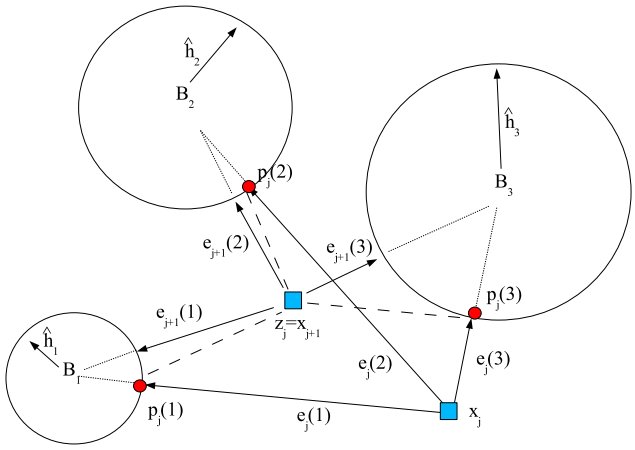


Fig. 6. Illustration of an iteration (from step j to $j+1$) of the gradient-descent algorithm in 2 dimensions with three beacons. The input position x_j is projected on 3 hyperspheres of radii $\hat{h_1}, \hat{h_2}$ and $\hat{h_3}$ around beacons B_1, B_2 and B_3 respectively, to obtain $p_j(1), p_j(2)$ and $p_j(3)$. The output of this iteration is $x_{j+1} = \frac{1}{3}[p_j(1) + p_j(2) + p_j(3)]$

As mentioned in Section III, one of the important issues that we are concerned with is the stability of the embeddings. Recall that we have access to only a partial distance matrix at any time (through measurements of a constant number of beacons). Therefore, when the topology changes, we need to ensure the stability of the embedding (as defined in Section III). In our algorithm 1, this is accomplished by retaining the distances to a fixed window size of the past beacons and attempting to solve the least-squares problem given in (3). This means that we are using a "rectangular windowing" of size b on the past beacon data. This naturally leads to the question of whether we can modify the algorithm and retain weighted information from a larger set of beacons. For example, we can modify the criterion in (3) to incorporate an exponentially

decaying window on the previous beacons as,

$$\min_{\vec{x}} \sum_{u=0}^{\infty} w_u[\hat{h}_u(k) - ||\vec{x} - \vec{b}_u||]^2 \quad (4)$$

where as before $b_u = x_{B_{k-u}}^{(k-u)}$ is the coordinate of the beacon B_{k-u} used at time $k-u$ and w_u is a forgetting factor. For example, for a rectangular window, $w_u = 1, 0 \leq u \leq B-1$ and is zero otherwise. We can also use an exponential forgetting factor with $w_u = a^u, 0 < a < 1$. These windowing schemes attempt to ensure stability of the embeddings to variations in the topology.

V. Combination with LER

The main application of our embedding algorithm, as pointed out earlier, is as a coordinate system for geographic routing in wireless networks. For a geo-routing algorithm to be able to forward a message to a destination node (or a data item identified by some key), it must first be able to determine the coordinates of that destination. This is achieved through a *location service*, essentially a distributed database that keeps track of the locations of all the nodes in the network. Every time the location of a node changes, it generates an update to the location service; a node sending a message first looks up the location of the destination through a query, and then sends the message towards that location using geo-routing.

A problematic aspect of location services is that the overhead due to control traffic (updates and queries) may become prohibitive in large and highly mobile networks. In a recent approach called Last Encounter Routing (LER), this problem is addressed by merging the location service with the geo-routing protocol [9]. The idea is that a message starts out with only an imprecise estimate of the destination's location. While the message advances through the network towards the destination's estimated location, the message can improve this estimate progressively. This process continues until the message arrives at the destination. The route taken by the message is not necessarily the shortest path, given that the message is not always moving in the optimal direction. However, if the precision of the location estimate increases quickly enough as the message moves through the network, then the route cost remains small, and ideally comparable to the shortest path route cost.

The key advantage of LER is that nodes only need to collect local information about changes in the network topology due to mobility. Specifically, each node maintains a last encounter table, in which it remembers the time and location when it has last been a directly connected neighbor of every other node. Collecting this last encounter history does not incur overhead, in that each node can derive it simply from local discovery messages (HELLOs) that are necessary to establish connectivity anyway. Therefore, there is no control traffic to update the location service every time a node moves, which makes LER inherently scalable.

Of course, the main question concerns the penalty in route cost due to the uncertainty of the destination's location. However, it was shown that for some mobility processes, the routes are asymptotically competitive, i.e., the average stretch is a small constant, independent of the size of the network [9].

One drawback of LER (and of all geo-routing approaches) is that nodes must be embedded in a low-dimensional normed space, which can be either the physical coordinate system (obtained through a positioning system such as GPS), or a virtual coordinate system such as one obtained through our embedding algorithm. Here, we are interested in the performance of LER based on an embedded rather than an absolute coordinate system. While this is conceptually straightforward, it is not a priori clear to what extent the performance of LER will decrease.

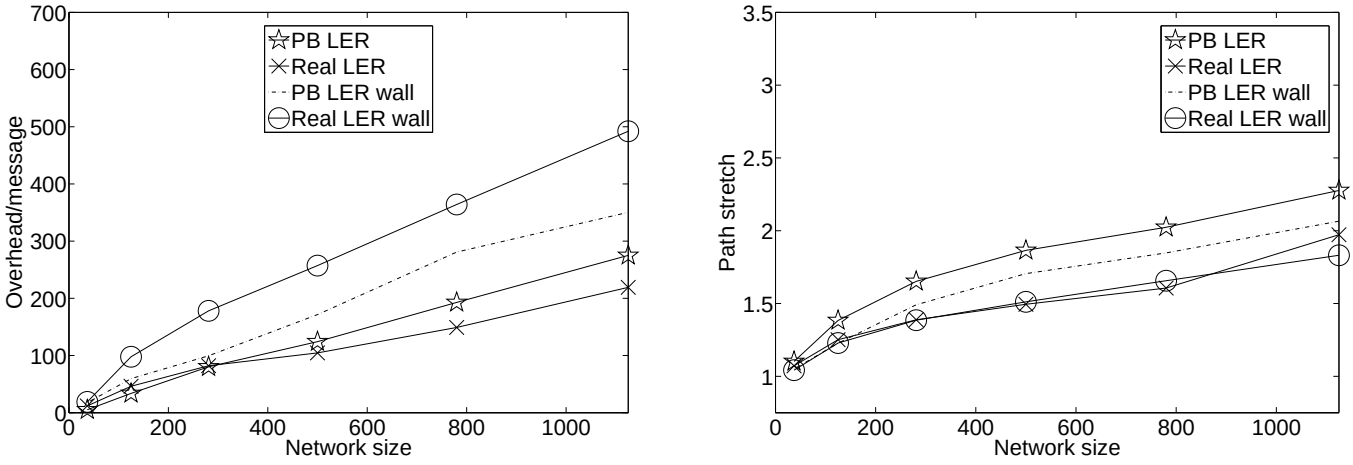
As we had pointed out earlier, an embedded coordinate system will not be perfectly stable. Given that the message is routed based on past locations of the destination, an unstable embedding adds noise to the position estimate carried by the message, which in turn may lead to a more circuitous route to the destination.

In Fig. 7(a) and 7(b), we show how LER routing performs when used in conjunction with PB. In our simulations, N nodes move on an $\mathcal{H}(N, 20, 10, 1.5)$. The connectivity graph is embedded using PB with 20 beacons, while the embedding dimension is 20. We send 1000 packets between random source destination pairs. With LER, all simulations resulted in a delivery ratio of 100% and we therefore analyze the resulting overhead. Figure 7(a) shows the total overhead for LER with real coordinates and for LER with PB (which also includes the overhead for building the embedding) in a setting without obstacles. The performance of LER with real and inferred coordinates is very similar. In a second experiment, we add 10 randomly placed "walls" of length 4. Walls are modeled by straight lines through which communication is impossible. Interestingly, once we add obstacles to the scenario in this way, PB has a significantly lower total overhead than real coordinates, despite the beacon messages required to build the embedding and despite the additional information that LER with real coordinates possesses.

This can be explained by the fact that adding obstacles makes the topology inhomogeneous and consequently real coordinates do not capture the structure of the network well anymore, leading to a large number of dead-ends. Recovering from such dead-ends is in turn more costly in terms of overhead than to build and maintain a virtual coordinate system which captures the topology more faithfully.

VI. Discussion

In this paper, we further explored the ideas presented in [5] for routing on dynamic wireless networks. The notion of stability in embeddings was introduced, which becomes important when combined with a distributed location service such as LER. We explored the structure of wireless connectivity graphs both in terms of inherent dimensionality and stability through numerical simulations. These explorations lead us to conjecture that wireless connectivity graphs could be represented well in a low-dimensional Euclidean space. Moreover,



(a) Overhead

(b) Path stretch

Fig. 7. Last Encounter Routing with PB and real coordinates in a mobile network.

our numerical results indicate that we have representations which provide stable embeddings. Future work includes the following. The theoretical questions include a formal proof of the above conjectures as well as development and analysis of algorithms with provable properties. The practical questions include performance evaluation on real wireless networks. The benefits of routing using embedded coordinates is also an interesting question to be addressed. In summary, we believe that this is a rich topic for future research.

REFERENCES

[1] A. Ozgur, O. Leveque, and D. Tse, "How does the information capacity of ad hoc networks scale?" in *Allerton conference on communications, control and computing*, 2006.

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

[3] D. B. Johnson, D. A. Maltz, and J. Broch, *Ad Hoc Networking*. Addison-Wesley, 2001, ch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks, pp. 139–172.

[4] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999.

[5] D. Tschopp, S. Diggavi, M. Grossglauser, and J. Widmer, "Robust geo-routing based on embeddings of dynamic wireless networks," in *Infocom*, 2007.

[6] R. Gowaikar, B. Hochwald, and B. Hassibi, "Throughput scaling in random wireless networks," in *Information Theory and Application*, San Diego, 2006.

[7] P. Indyk and J. Matousek, "Low-distortion embeddings of finite metric spaces," in *CRC Handbook of Discrete and Computational Geometry*, 2004, chapter 8.

[8] I. Borg and P. J. Groenen, *Modern Multidimensional Scaling - Theory and Applications*, 2nd ed., ser. Springer Series in Statistics, 2005.

[9] M. Grossglauser and M. Vetterli, "Locating Mobile Nodes with EASE: Learning Efficient Routes from Encounter Histories Alone," *IEEE/ACM Trans. on Networking*, vol. 14, no. 3, June 2006.