Life (and routing) on the Wireless Manifold

Varun Kanade* Santosh Vempala†
College of Computing, Georgia Tech
{varunk,vempala}@cc.gatech.edu

ABSTRACT

We present the *wireless manifold*, a 2-dimensional surface whose geodesic distances accurately capture wireless signal propagation. As a result, the connectivity graph of a wireless network can be viewed as a disk graph *on the manifold*. A compact representation of the manifold can be reconstructed from a sparse set of signal measurements. The manifold distance suggests a simple routing algorithm that avoids obstacles and naturally handles mobile nodes without explicitly maintaining the connectivity graph. It is more efficient compared to using Euclidean distance as measured by success rate, routing load and failure tolerance. Placing sensors to cover the manifold is more effective than covering the underlying physical space.

1. Introduction

The connectivity graph of a wireless network is determined by complex factors such as geographic layout, physical obstacles, noise and electromagnetic interference. Moreover, a principal feature of ad hoc networks is the mobility afforded to the nodes and this implies that the topology of the network could be frequently changing.

A widely studied model for wireless connectivity is the unit disk graph model [9]. Each node of the network is represented as a point on the 2-dimensional plane and two nodes are connected if their distance is at most 1, i.e., the range of a node is a unit circle centered at its location. While this model is attractive and often used for algorithm design or validation, it does not take into account the many factors other than physical (Euclidean) distance that affect connectivity; indeed the disk assumption is typically violated in practice [3, 14, 13, 2].

In this paper, we define the *wireless manifold* using the basic physical principle that signal strength decay follows an inverse square law. The key property of the wireless manifold is that the shortest path (geodesic) distance along the manifold between any two points estimates the signal decay between them. Thus, the connectivity graph is determined by disks *on the manifold*; a disk of radius r on the manifold is the set of all points within geodesic distance r. The contours of these disks on the plane may be very different from circles depending on the structure of the manifold.

*Supported in part by ARC ThinkTank and by the NSF.
†Supported in part by a Raytheon fellowship and by the NSF.

We give an algorithm to construct a compact representation of the manifold from a sparse set of signal strength measurements. Without directly modeling any physical factors, the representation captures and predicts signal decay to high accuracy and is vastly superior to the best possible Euclidean representation, thus improving on the disk model and its known refinements. The manifold representation encodes obstructions implicitly, making the model conceptually much simpler. It is qualitatively different from previous work on assigning virtual coordinates to the nodes in a network [6, 22, 4] or modeling non-Euclidean features of network connectivity by explicitly modeling obstructions [5].

Roughly speaking, the manifolds we consider are distorted 2-dimensional grids. We imagine placing a grid on the region of interest and choosing a length for each grid edge so as to make the resulting shortest path distances as close as possible to a given set of distances. This formulation, which we call *the best manifold problem*, appears quite general and unexplored. Most theoretical work in embeddings focusses on the minimum distortion of mappings from one space to another. Here we are asking a different question: given the mapping, find the best target space from a given class. The formulation is described more precisely in Section 2.

We note that the manifold size can be kept constant, independent of the number of nodes in the network. Each network node will be associated with its nearest manifold grid node. Our representation decomposes the connectivity graph of a wireless network into two parts — the manifold itself which is not expected to change often and the locations of nodes which might change frequently but are easy to update. This can be used in several scenarios; two important ones are (a) sensor placement and (b) routing.

Placing sensors. A typical goal of sensor placement is to provide coverage of a certain area or connectivity for a given set of nodes (in the case of relays). Placing nodes uniformly on the underlying physical space ignores obstacles and interference. On the other hand, the wireless manifold incorporates these features and maximizing coverage of the manifold should be more efficient. Alternatively, the manifold can be used to identify regions where the coverage needs to be boosted.

Routing. Unfortunately, over a decade of research on wireless routing has not led to a viable and scalable system. Methods that work extremely well for wired routing have not been successfully adapted to wireless/mobine networks. The manifold representation

suggests routing protocols that are fundamentally different. Before we describe the new ideas, it is perhaps useful to discuss a method that might appear related, viz., geographic routing [7, 15, 11, 17, 16]. The latter is an elegant theoretical approach to wireless routing that requires planarization of graphs [23, 8] and assigning coordinates on the plane. It is guaranteed to work in the unit-disk model. Kuhn *et al.*[18] have proposed relaxations of unit-disk graphs to improve robustness of planarization techniques which fail in case the underlying graph violates the standard unit disk assumption. Kim *et al.*[13] propose the cross-link detection protocol, which enables provably correct geographic routing on arbitrary connectivity graphs. All these approaches involve deleting certain links and then assigning coordinates to guarantee success of routing algorithms. The main idea for routing on the wireless manifold is that between two points, there exists a path on the manifold that is monotonically decreasing in distance. The manifold representation makes it easy to maintain shortest path distances. For routing, each packet could have a header field indicating its current manifold distance to the target node. Any node that receives the packet retransmits if and only if its distance to the target is less than the current distance of the packet (in the header) by a certain *width* parameter (which can be tuned to reduce network load) and updates the header. We present experimental results demonstrating the main properties of this protocol. We note that manifold distance can be used in any of the known distance-vector routing protocols.

When a node moves to another location, it has to update its location on the manifold and inform other nodes that wish to send to it. We can implement known location service methods such as [19] for this purpose. The manifold representation is suited for mobility since the manifold itself is not expected to change rapidly (e.g., new buildings). All that needs to be updated are the node locations. Unlike previous routing algorithms manifold routing does not rely directly on the connectivity graph. Even though the connectivity graph could be rapidly changing in response to node movements, it is implicitly known from the wireless manifold (which is more stable) and the node locations. Further, when the manifold does change, our algorithm for finding a representation can also be used to refine it.

In contrast with ad hoc on-demand distance vector (AODV) [21] routing or dynamic source routing (DSR) [10] the manifold routing algorithm does not incur frequent overheads of route discovery or of carrying the entire route in the header. By tuning the width parameter, we demonstrate that it is possible to ensure fairly low routing load on the network, as is the case with these algorithms.

2. The wireless manifold

In this section, we describe precisely a simplified discrete model for manifolds that will suffice for our purpose. Our manifolds are distorted 2-dimensional grids. Let $G = (V, E)$ represent the graph of a $k \times k$ grid where V is the set of grid points with coordinates from the set $\{1, \ldots, k\}$ and E is the set of all grid edges, i.e., pairs of adjacent vertices on the grid. A manifold M is obtained by assigning a nonnegative length, $l(i, j)$, to each grid edge $(i, j) \in E$. These lengths induce a metric on the grid vertices where the distance $M(u, v)$ between each pair of grid points u, v is the length of the shortest path between them. The set of all manifolds, $\mathcal{M}$, is the set of all such metrics induced by length assignments to the grid edges.

We now define the *best manifold* problem. The input is the set of locations of wireless nodes and the measured signal strengths for some pairs of nodes, i.e., a subset W of the grid vertices V and a subset F of pairs of points from W along with nonnegative "distances" $d(u, v)$ for each pair in F. The problem is to find the manifold M, i.e., lengths $l(i, j)$ for grid edges, so that the induced shortest path metric is as close as possible to the given distances for pairs in F in the following measure:

$$\min_{M \in \mathcal{M}} \sum_{(u,v) \in F} \left(d(u, v) - M(u, v)\right)^2 . \qquad \text{(I)}$$

Alternative formulations are possible, e.g., one could restrict the final metric to be non-contracting, i.e., $M(u, v) \geq d(u, v)$, and then minimize the maximum distortion, i.e., minimize the maximum of $M(u, v)/d(u, v)$ over pairs in F. Another possibility is to minimize the average distortion.

Unfortunately, all these formulations are NP-hard, even to approximate. This follows due to a reduction from 3-SAT (we omit the proof in this paper). Thus, we do not hope to find an efficient algorithm to solve the best manifold problem in the worst case (for arbitrary input distances).

On the other hand, the objective function (I) is differentiable, and we use the following method to find a local optimum.

1. Convert signal decay $s(x, y)$ for each pair observed to effective distance $d(x, y) = 1/\sqrt{s(x, y)}$.
2. Complete d to a metric using the induced shortest path distances
3. Set every edge to the same length (we use the value that minimizes (I)).
4. Repeat:
   - Compute the gradient y of (I) at the current solution z.
   - Move z in the direction of the y without making any component of z (grid edge length) negative; if the gradient makes some edge negative, keep that edge length at zero.

One could impose other linear constraints, e.g., insisting that some pairs have lower bounds on the manifold distance.

3. Sensor placement

Once we have identified a manifold, we can use it to guide the placement of network resources such as sensors or relays.

We place nodes so as to cover the wireless manifold rather than the Euclidean plane on which the nodes lie. In a random placement, we choose each grid point with probability proportional to the sum of the lengths of adjacent edges. In Section 5, we compare picking grid nodes uniformly at random (which corresponds to sampling the Euclidean plane) to picking them according to the manifold, to see which method achieves connectivity and coverage faster.

Besides connectivity, placing relays using the manifold can also be advantageous for routing. We discuss and evaluate that aspect in the context of routing.

4. Routing

The basic idea for routing is very simple. Imagine each node has a table of the distances to every other node in the network (we will shortly see that we do not need an explicit table and distances can be computed from a small representation). A packet P has two pieces of information in its header: (1) the target node t and (2) the manifold distance d to the target t from the location of its most recent retransmission. Let $R(v)$ denote the radius of influence of node v. Here is a candidate rule for forwarding packets:

When node v receives packet P,
– *if $M(v,t) < d - \alpha R(v)$,*
– *set $d := M(v,t)$ and retransmit P.*

We also allow nodes to broadcast. This is done by setting the destination ID to a special character. Broadcast is useful when a node has to announce its new location.

The *width* parameter α can be tuned for efficiency. At $\alpha = 0$, the algorithm is guaranteed to find a route if one exists. It might use a larger number of nodes than necessary for retransmission. As we increase α, the set of nodes participating reduces. We propose that α be adjusted dynamically. Figure 7 in the evaluation illustrates contours of network load for different values of α on a randomly generated instance. Moreover, each node could have its own width setting, based on the local structure of the manifold.

We view the size of the grid to represent the manifold as a constant independent of the size of the network. Each node keeps a copy of the grid. The distance between two network nodes is estimated by the distance between their nearest grid points.

5. Evaluation

In this section, we present our preliminary evaluation for (a) the quality and learnability of the manifold (b) sensor placement and (c) routing efficiency.

We report experiments on the CRAWDAD data set Rutgers-noise [12] as well as randomly generated instances. The Rutgers-noise data sets contain RSSI (Received Signal Strength Indication) measurements for some pairs of nodes. The data set was collected by injecting white Gaussian noise at multiple locations into the ORBIT[1] indoor testbed, which consists of an 8×8 grid, with nodes placed at some locations. There are 29 nodes in the data sets we examined. Signal measurements were collected at 3 noise levels: 0 dB(dbm0), -5 dB(dbm-5) and -20dB(dbm-20).

We also generated random manifolds for evaluation. For this we used Gaussian distributions with randomly chosen centers and random covariances. Each Gaussian affects the lengths of manifold edges that are at least some distance away from its center. The length of an edge is proportional to the maximum density any Gaussian places at the midpoint of the edge (see Fig. 4 for an example). We chose this model as it appears to capture the effects of having steep barriers caused by obstructions.

5.1 Learning the manifold

Manifold distance is a shortest path metric and so will always satisfy the triangle inequality. As a preliminary analysis, we checked whether the effective distance (inverse square root of signal decay) was approximately a metric. We found an average violation of 1.07, a maximum violation of 1.83 and less than 25% showing any violation at all (violation is the ratio of the direct effective distance between two points u, v to the shortest alternative path with effective distances).

Next we turn to the central question of how well our manifold can approximate effective distance (and therefore signal strength) and how well it generalizes. We applied the algorithm described in Section 2 to each of the Rutgers-noise data sets. We compared the manifold distance to the original effective distance. To test generalization, we randomly omitted a subset of the observed measurements, computed the manifold on the rest and then compared the manifold prediction with the omitted measurements.

Figure 1 shows the manifold obtained by the algorithm for the data set dbm0. The red dots are positions of network nodes. The shortest path between two nodes is marked and is quite different from the shortest path on the plane. The manifold also gives an idea of barriers across which communication is harder and thus suggests points to place relays.

To evaluate the accuracy of the manifolds obtained we used several measures. Table 1 compares the manifold with the best plane embedding in terms of (1) average error: the ratio of the objective function value (I) to the sum of squares of all the effective distances, (2) average expansion: the average expansion factor for pairs whose distance went up compared to the original, (3) average contraction: the average shrinking factor for pairs whose distance went down and (4) maximum distortion: the product of maximum contraction (max factor by which some edge length was reduced) and maximum expansion (factor by which some edge length was

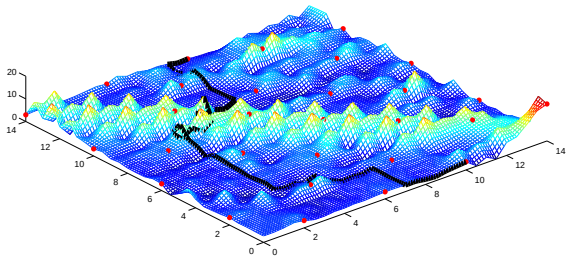

Figure 1: : Manifold for dbm0

Data set	Measure	Euclidean	Manifold	Manifold prediction
dbm0	Avg Error	0.185	0.017	0.07
	Avg Exp	2.33	1.13	1.25
	Avg Contr	0.35	0.9	0.93
	Max Dist	25.85	2.86	1.71
dbm-5	Avg Error	0.137	0.022	0.03
	Avg Exp	1.83	1.15	1.36
	Avg Contr	0.44	0.89	0.87
	Max Dist	14.81	2.58	1.87
dbm-20	Avg Error	0.145	0.021	0.02
	Avg Exp	1.72	1.11	1.12
	Avg Contr	0.29	0.88	0.89
	Max Dist	14	2.13	1.79

Table 1: Embedding data sets into manifolds

increased). From the first and second columns of the table, we see that the manifold embedding typically has a 5 to 10-fold advantage over Euclidean and further the actual error values are quite small. Moreover, the average error of the rank-5 approximations for the three data sets dbm0, dbm-5, dbm-20 were 0.093, 0.086 and 0.066 respectively. The matrix of specified distances does not have a good low-rank approximation, implying that even using more coordinates does not help. The curvature induced by the manifold is essential to accurately recover signal propagation.

To test how well the learned manifold generalizes, we dropped at random 8% of the measured effective distances (edges) from the data sets, computed the manifold on the rest of the observations and made predictions on the 8% not used in computing the manifold. The last column in Table 1 shows that the manifold prediction error is low on all the measures and is comparable to that on the full set of values. From this we conclude that the manifold captures and generalizes wireless connectivity accurately.

We also collected signal strength data ourselves by placing wireless nodes at various positions on the first floor of the Klaus Advanced Computing Building at Georgia Tech. Figure 2 shows the floor plan of the building with the placement of the nodes. Figure 3 shows the manifold obtained for this region. The average error, expansion and contraction were again significantly smaller for the manifold compared to the best Euclidean scaling.

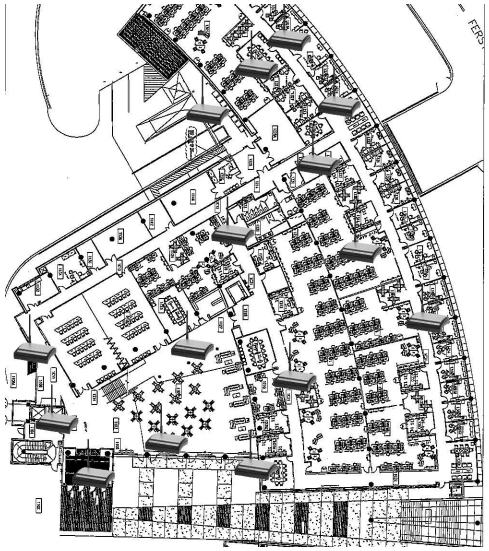

Figure 2: Placement of nodes in Klaus building at Georgia Tech

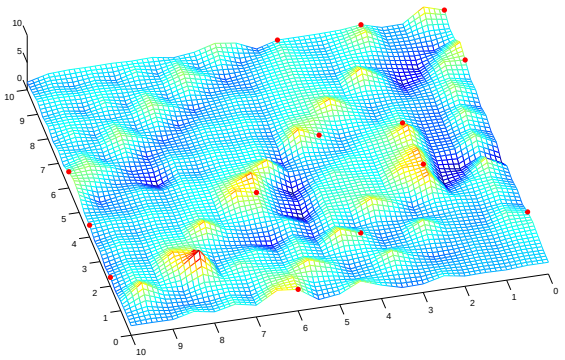

Figure 3: Manifold for the Klaus building at Georgia Tech

5.2 Sensor placement

We used randomly generated manifolds for evaluating sensor placement. Figure 4 gives an example. For a fixed radius r (two grid points on the manifold can communicate directly if their manifold distance is at most r) we chose node locations on the in two ways: (i) a uniformly random grid points (ii) random grid points each with probability proportional to the sum of the incident manifold edge lengths. We then report the number of nodes required to be chosen to ensure that the network is connected. Figure 5 show the plot of the number

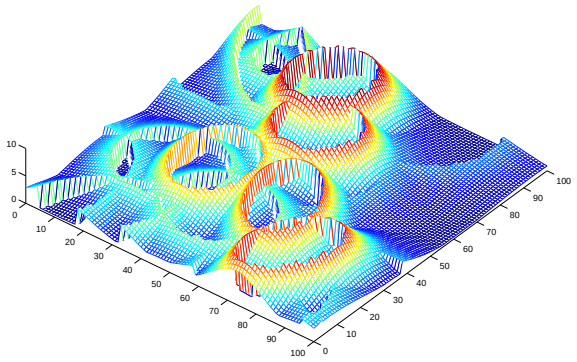

Figure 4: Randomly generated manifold

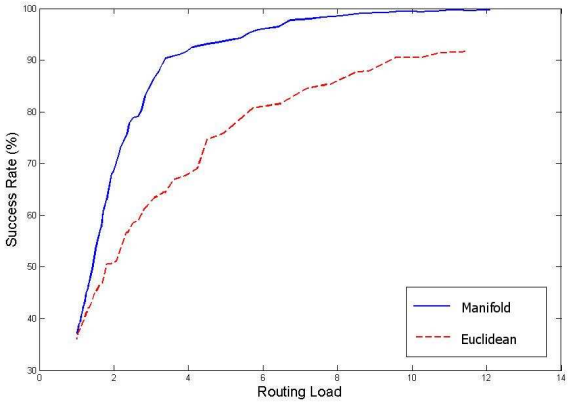

Figure 6: Routing on dbm0

of nodes required to achieve connectivity vs the radius of transmission for the two different ways of choosing sensor locations.

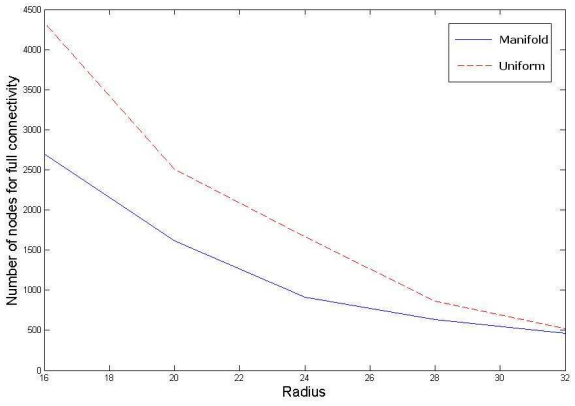

Figure 5: Number of nodes required to achieve perfect connectivity at different radii

We also measured the number of nodes for full *coverage* vs the radius and separately, the fraction of pairs connected vs number of nodes and fraction of the grid covered vs number of nodes. In every case, manifold sampling was significantly better.

5.3 Routing

We used the following measures for evaluating routing in simulations: (1) Success rate, i.e. fraction of node pairs that can communicate (2) Routing load, i.e., the average number of packets forwarded in the network per node. We implement the forwarding rule described in Section 4 using the manifold distance as well as Euclidean distance. We first compared these methods on each of the Rutgers-noise data sets by routing between every pair of nodes.

On each of the three Rutgers-noise data sets, at every radius of transmission that we used, manifold routing had a higher success rate and smaller load. The routing load can be adjusted by varying the width parameter α of the routing algorithm. Figure 6 plots the success rate vs the average load for the dbm0 data set. We see that to ensure 90% success rate, manifold routing incurs a routing load of less than 4, where as Euclidean routing incurs a routing load of 11.5, nearly 3 times as much. We found similar plots for the other two data sets.

We also evaluated routing on randomly generated manifolds. For these, we picked network node locations in two ways. First, we picked every other grid point vertically and horizontally and routed between random pairs. Second, we picked N nodes, by picking $N/2$ random grid points and the other $N/2$ either at random from the manifold (when using manifold distance for forwarding) or uniformly from the grid (when using Euclidean distance). Messages were routed between random pairs drawn from the common nodes (the first $N/2$). The motivation for the last choice was to study the effect on routing of choosing relay nodes in the two different ways. Figure 7 shows the nodes retransmitting packets while attempting to route between two points (bottom left hand star and top right hand star). Manifold routing is successful for α up to 0.5. In the figure, the nodes marked in white circles are the only ones retransmitting at $\alpha = 0.5$ and as α goes to zero, the darker ones also start retransmitting and finally at $\alpha = 0.0$, all marked nodes are retransmitting. Euclidean routing tends to flood almost the entire grid when it succeeds.

Figure 8 shows the trade-off between success rate and routing load for the two routing methods on the first experiment with random data using a subgrid of points as node locations. We see that to ensure 95% success rate, manifold routing incurs a routing load of 178, whereas for the same success rate Euclidean routing incurs a routing load of 485. In the second experiment, Figure 9 shows the same comparison for the randomly chosen network node locations. To achieve 95% success rate manifold routing incurs routing load of 265 whereas for the same success rate, Euclidean routing incurs a rout-

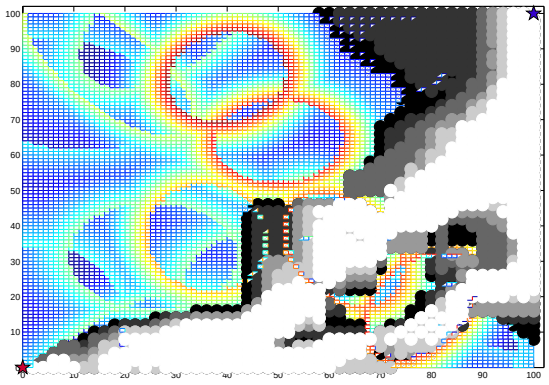

Figure 7: Manifold routing

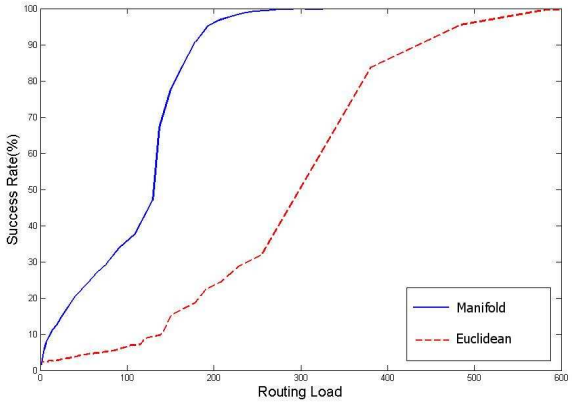

Figure 8: Routing on a random manifold

ing load of 603. Thus by tuning the width parameter α, we see that a large reduction in network load is possible for manifold routing without reducing the success rate significantly.

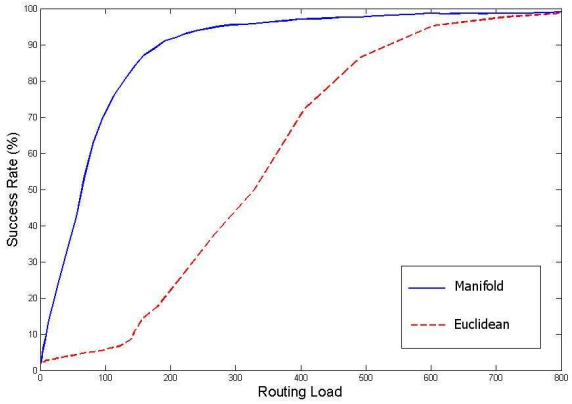

Figure 9: Routing with randomly chosen node locations

Finally, we also ran the routing experiments on random manifolds after failing each node independently with probability p ranging from 0 to 0.8. The success rate using manifold distance was always higher than that using Euclidean distance.

6. Discussion and Future Work

Our results on recovering the wireless manifold and predicting signal strengths using it provide compelling evidence that (a) the manifold captures wireless communication with high accuracy without explicitly modeling complex physical factors and (b) it can be recovered using a simple algorithm. We plan to investigate these findings more thoroughly in different test zones including large buildings and urban landscapes. We will also study the size of the manifold representation required to produce an accurate distance measure and extensions such as directed grids and nonuniform grids, e.g., allowing the grid to be finer in places. The idea of using a manifold rather than a plane seems natural for capturing other properties of a network besides connectivity and we are eager to see if this representation improves on earlier approaches assigning virtual coordinates [20].

Our findings raise intriguing theoretical questions. Why is the algorithm so effective in the face of the worst-case hardness? One reason could be that the best manifold is easy to find provided it has low distortion (as seems to be in the wireless setting). Next, what graph metrics are embeddable in our manifolds? An extension of Kuratowski's characterization of planar graphs seems natural.

From our preliminary experiments, routing on the manifold is effective and can tuned for efficiency using the width parameter α. So far we have tested the routing ideas only in simulation. We plan to fully evaluate the algorithm by setting up a multi-hop wireless network and measure throughput, latency and recovery time under failures and mobility. We expect that using manifold distance will provide improvements for existing distance-based routing algorithms.

An exciting scenario for multi-hop wireless networks is providing connectivity in developing countries that face the "last-mile" problem. Broadband access is expensive and while fiber is available it comes within a mile of most users but not all the way. In ongoing work with the TeNet group in Chennai, India, we are studying manifold routing along with other known approaches in an urban area that we have already identified. This setting is particularly interesting because other solutions such as building powerful antennas or providing cable access are prohibitively expensive or otherwise impractical. Moreover, power consumption is one of the considerations and so multi-hop and low network load can both play a useful role.

Acknowledgements. We are grateful to Mostafa Ammar, Mike Best, Constantine Dovrolis, Nick Feamster, Dick Karp, Anirudh Ramachandran and Ellen Zegura for helpful comments.

REFERENCES

[1] `http://www.orbit-lab.org/`.

[2] J. B. Andersen, T. S. Rappaport, and S. Yoshida. Propagation measurements and models for wireless communications. *IEEE Communications Magazine*, 33, 1995.

[3] L. Barriére, P. Fraigniaud, and L. Narayanan. Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *DIALM '01: Proceedings of the 5th int. workshop on Discrete algorithms and methods for mobile computing and communications*, pages 19–27, New York, NY, USA, 2001. ACM Press.

[4] Y. Ben-Asher, M. Feldman, and S. Feldman. Ad-hoc routing using virtual coordinates based on rooted trees. *SUTC*, 01:6–13, 2006.

[5] N. Carlsson and D. L. Eager. Non-euclidean geographic routing in wireless networks. *Ad Hoc Netw.*, 5(7):1173–1193, 2007.

[6] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM '04*, pages 15–26, New York, NY, USA, 2004. ACM Press.

[7] G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, Mar. 1987.

[8] K. Gabriel and R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[9] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[10] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The dynamic source routing protocol for mobile ad hoc networks, 2003.

[11] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual int. conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM Press.

[12] S. K. Kaul, M. Gruteser, and I. Seskar. CRAWDAD trace set rutgers/noise/rssi (v. 2007-04-20). Downloaded from http://crawdad.cs.dartmouth.edu/rutgers/noise/RSSI, April 2007.

[13] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In *NSDI'05: Proceedings of the 2nd Symposium on Networked Systems Design & Implementation*, pages 16–16, Berkeley, CA, USA, 2005. USENIX Association.

[14] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. On the pitfalls of geographic face routing. In *DIALM-POMC '05: Proceedings of the 2005 joint workshop on Foundations of mobile computing*, pages 34–43, New York, NY, USA, 2005. ACM Press.

[15] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[16] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In *PODC '03: Proceedings of the 22nd annual symposium on the Principles of distributed computing*, pages 63–72, New York, NY, USA, 2003. ACM Press.

[17] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *DIALM '02: Proceedings of the 6th int. workshop on Discrete algorithms and methods for mobile computing and communications*, pages 24–33, New York, NY, USA, 2002. ACM Press.

[18] F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad-hoc networks beyond unit disk graphs. In *DIALM-POMC '03: Proceedings of the 2003 joint workshop on the Foundations of mobile computing*, pages 69–78, New York, NY, USA, 2003. ACM Press.

[19] 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 *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130, New York, NY, USA, 2000. ACM Press.

[20] E. K. Lua, T. Griffin, M. Pias, H. Zheng, and J. Crowcroft. On the accuracy of embeddings for internet coordinate systems. In *IMC '05: Proceedings of the ACM SIGCOMM-Usenix Internet Measurement Conference*, pages 125–138, 2005.

[21] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing, 2003.

[22] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without local information. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 96–108, New York, NY, USA, 2003. ACM Press.

[23] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.