GHVR: One-to-One Gradient-Height Vector Routing in Wireless Mesh Networks

Jai-Jin Lim
SAMSUNG ELECTRONICS,KOREA
Email: jjlim@samsung.com

Kang G. Shin
Univ. of Michigan,Ann Arbor,USA
Email: kgshin@eecs.umich.edu

Taejoon Park
Korea Aerospace University,KOREA
Email: tjpark@kau.ac.kr

***Abstract*— This paper addresses the problem of routing packets between any nodes in a wireless mesh network (WMN) equipped with a set of reference landmarks. These reference landmarks are pre-selected Internet gateways or special nodes that broadcast beacon messages periodically. WMN nodes are then assigned multiple gradient-based coordinates, which are heights relative to the landmarks. These multiple heights of each node are treated as if they were the actual multi-dimensional geographic locations of that node. The proposed routing protocol, called GHVR, runs geographic routing with these multi-dimensional geographic locations. Unlike other similar approaches that use discrete hop counts to the landmarks, GHVR uses *continuous* heights expressed in real numbers. Our ns-2 simulation results demonstrate that GHVR requires fewer landmarks and improves routing performance by almost a factor of 2 over that of using the discrete hop-count coordinate.**

I. Introduction

Although communications from many nodes to a wired Internet gateway represent a typical interaction pattern for wireless mesh networks (WMNs), other types of communication scenarios are also possible. For example, users in a community-based WMN may want to access data or video files owned and exported by other users, or exchange shared secrets with each other, thereby requiring efficient one-to-one routing.

This paper explores how to design such a one-to-one routing protocol for WMNs. One approach to this problem is to use conventional proactive distance-vector routing [1] or link-state routing [2]–[4], or reactive on-demand routing [5]. However, one-to-one routing in WMNs tends to be occasional. Thus, keeping a proactive routing protocol running for all destinations is not cost-effective. Likewise, running a reactive routing protocol becomes expensive and suffers the scalability problem as the number of one-to-one communication sessions increases.

We present a novel routing protocol called *Gradient-Height Vector Routing* (GHVR) for efficient one-to-one communications in WMNs. It does not flood route-discovery requests nor does it maintain route information between nodes. It only requires information on routes to a set of pre-selected gateways. We eliminate the route request step by running geographic routing [6]–[8] over *virtual coordinates* extracted from the routes to the set of pre-selected gateways; we refer to the set of pre-selected gateways used to construct virtual coordinates as *landmarks*. Running geographic routing with the virtual coordinates is then realized by making the routing protocol treat the virtual coordinates as if they were real, geographic locations.

GHVR is a virtual coordinate routing protocol designed to achieve the flexibility, scalability, and simplicity of geographic routing even when the true geographic locations of nodes are not available. Other routing protocols [9]–[11] are similar to GHVR in the spirit: they are all landmark-based (or gradient-based) coordinate routing with coordinates represented as a set of scalar values such as hop counts or distances to the landmarks. The type of scalar value indicates if coordinates are *discrete* or *continuous*. For example, the coordinates defined in terms of hop count are discrete, whereas the coordinates using distance are continuous. GHVR differs from other protocols [9]–[11] in that GHVR is built on continuous coordinates while the others on discrete coordinates.

Continuous coordinates are finer-grained than the discrete coordinates, and make a significant improvement of routing performance over the discrete counterpart. Defining continuous landmark-based coordinates in networked systems is not new. For example, several techniques to produce fine-grained and accurate landmark-based coordinates built on a routing protocol have already been proposed in [12], [13].

However, a clear distinction should be made between coordinates *built on* a routing protocol and coordinates *for building* a routing protocol itself. Coordinates built on a routing protocol are mainly used to infer the end-to-end delay between two arbitrarily-chosen nodes before they communicate with each other. A global virtual coordinate is built from actual round-trip time measurements with the selected landmarks; these landmarks also measure round-trip times among themselves. Measuring the round-trip time requires routing to be implemented first between landmarks, and between landmarks and nodes. It is also assumed that the same routing algorithm is used between any pair of nodes; otherwise, any inference from measurements becomes meaningless. Unfortunately, all of existing continuous coordinates [12], [13] fall into this

978-0-7695-3450-3/09 $25.00 © 2009 IEEE

category.

Coordinates for building a routing protocol are used to route packets between a pair of nodes when they need to communicate with each other. A global virtual coordinate is built using scalar values, typically hop count [9]–[11], from nodes to each landmark that periodically broadcasts beacons. How to route packets between nodes has not yet been decided and therefore, needs to be designed differently from the broadcast. If there are d landmarks in the network, the "location" of each node will be represented by a d-dimensional vector. Routing packets from one node to another is then a simple matter of running the standard geographic routing with the thus-obtained global virtual coordinate.

Because of differences in both the underlying assumptions and the intended application context, the same coordinate determination techniques as developed on top of an existing routing protocol cannot be used for building a new routing protocol itself. This paper elaborates on the latter. Even if we limit ourselves to landmark-based coordinates for routing, there are several design factors that affect routing performance, including the coordinate type and the definition of distance function over the virtual coordinate space.

This paper makes the following main contributions:

- It presents a one-to-one routing protocol, GHVR, that routes packets with landmark-based virtual coordinates. The proposed landmark-based virtual coordinates are based on the estimation of each link's length on the minimum-cost paths to the landmarks, and are finer-grained than the existing discrete coordinates that use hop counts.
- It presents comprehensive and comparative results as to how (i) different types of landmark-based coordinates and (ii) different types of the distance functions over landmark-based coordinates affect routing performance.

The rest of this paper is organized as follows. Section II describes the related work. Section III details GHVR. Section IV provides extensive evaluation of the proposed scheme under realistic wireless network models. Section V concludes the paper.

II. Related Work

Defining a continuous virtual coordinate in networked systems is not new, but has never been tried for designing a routing protocol. For example, Vivaldi [13] is a decentralized network coordinate system that assigns hosts synthetic coordinates (expressed in real numbers) such that the distance between two hosts in the virtual coordinate space accurately predicts the actual communication latency between them. In other words, Vivaldi is designed to predict the latency prior to communication between nodes.

Unfortunately, Vivaldi and most of the work related to Vivaldi require that routing be implemented before coordinates are established. However, GHVR and other existing work [9]–[11] define gradient-based coordinates first, then use the coordinates to route packets between a pair of nodes. Thus, the assumptions and the design requirements for the gradient-based coordinates are totally different from those of Vivaldi.

Several routing protocols, such as LCR [9], BVR [11], and GLIDER [10], define synthetic coordinates with respect to a set of pre-selected nodes called *landmarks* [9], [10] or *beacons* [11].[1] They use hop count along the minimal-hop path (e.g., LCR or GLIDER), or the minimal-cost path to the set of selected landmarks (e.g., BVR).

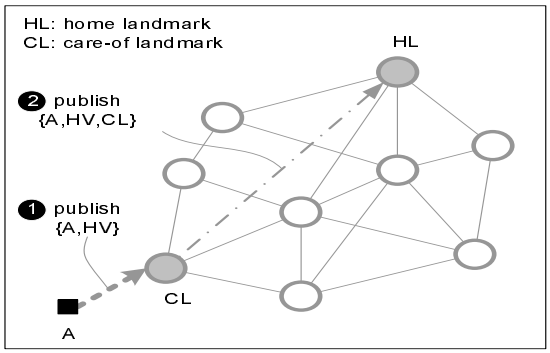
LCR identified delivery-failure problems, such as routing loops or greedy routing failures, associated with the hop-based discrete coordinate. Instead of seeking to improve the granularity of a coordinate, LCR uses a loop-avoidance mechanism and time-to-live (TTL)-based packet dropping. These mechanisms only address how to avoid wasting wireless bandwidths rather than offering a reliable mechanism to deal with the problem associated with the discrete coordinate.

The original geographic routing protocols [6], [7], [14] did not take link-quality into account. Later, two similar geographic link metrics such as packet-reception-rate (PRR)×DISTANCE [15] and normalized advance (NADV) [8] were proposed to make the original geographic routing aware of link-quality. The greedy forward definition used in GHVR is NADV in the gradient-based coordinate space.

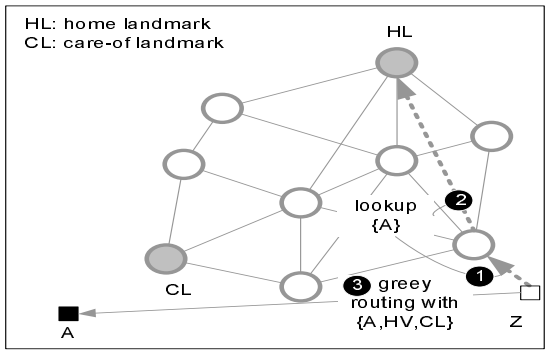
Research over the last few years has focused on understanding and characterizing link-quality in wireless sensor networks [16] or mesh networks [17]–[21]. New link or path metrics of wireless link-quality have been proposed, including the expected transmission count (ETX) [17], the expected transmission time (ETT), the weighted cumulative ETT (WCETT) [20], and the effective number of transmissions (ENT) [22].

Even though these metrics have proved to be effective in improving throughput, their broadcast-based measurements create mismatches between broadcast-based and unicast-based neighbor sets. For example, in 802.11-based wireless networks, broadcast is sent at a default rate of 2Mbps, whereas unicast uses a default rate of 11Mbps. This difference in data rate makes the range of broadcast much greater than that of unicast. The hybrid link-quality measurement scheme in GHVR recognizes such disparity and incorporates it in the calculation of link-quality. More recent work has addressed accurate link measurement, e.g., efficient and accurate link-quality monitor (EAR) [23], which also uses unicast- and broadcast-based link-quality measurements.

[1] GLIDER only uses virtual coordinate routing as intra-tile routing in its entire routing framework.



(a) Address publish



(b) Address lookup and data forwarding

Fig. 1. Illustration of one-to-one GHVR

III. Gradient-Height Vector Routing

A. Overview

Fig. 1 illustrates how GHVR works in general. Unlike the Internet routing where an IP address is used not only to identify a node, but also to route packets, the landmark-based routing uses the coordinates only for routing packets. Thus, it needs a location service that maps the node identifier to the landmark-based coordinate.

In its simplest form, a location service can be implemented by using the "home" and "care-of" landmarks (HL, CL). The home landmark is uniquely determined by hashing a node ID, whereas the care-of landmark is the closest (e.g., in number of hops or estimated distance) landmark to the node. In Fig. 1(a), node A publishes periodically to CL, its ID and landmark-based coordinates. The landmark-based coordinates of a node are also called the "height vector (HV)" of that node. Also, CL periodically publishes a list of nodes registered with itself to their respective HLs. For a member node A, the information reported by CL to A's HL is a triplet <A,HV,CL>. CL may aggregate the information for the same HL and minimize the report traffic.

When a node Z in Fig. 1(b) wants to send data to node A without knowing A's HV, it has to take several steps as follows. First, it issues a query for A's HV to the nearest landmark. If that nearest landmark (or Z's CL) has no entry for A's HV, it relays the received query to A's HL, which always knows A's HV. After obtaining

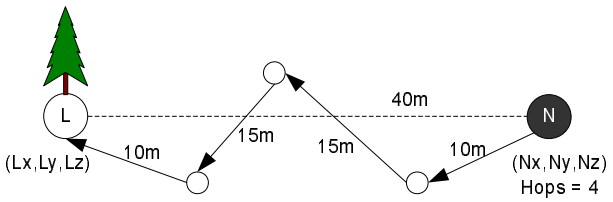


Fig. 2. A reference model for various landmark-based virtual coordinates.

A's HV from either Z's CL or A's HL, node Z will send packets to A by using <A,HV,CL> as the "address" for routing packets. Note that CL in this routing address is the nearest landmark to A.

Optimizing the location service is beyond the scope of this paper. In fact, the problem of designing a scalable location service in the landmark-based coordinate space itself is extensive enough to constitute a paper on its own. Here we only want to show how GHVR works for one-to-one communications.

B. Type of Coordinates

The reference model in Fig. 2 illustrates different types of the landmark-based coordinates. The *absolute* coordinate represents true geographic locations (Nx, Ny, Nz), which are irrelevant to the landmark-based coordinates. The *straightline* coordinate of node N is its Euclidean distance the landmark L, i.e., 40m. The *hops* coordinate is hop count (4) on the minimum-cost path. The *path-distance* coordinate is the sum of the lengths of links of the minimum-cost path, which is 50m in Fig. 2. In a line topology, *path-distance* and *straightline* coordinates are the same, and represent the *ideal* landmark-based coordinates. Note that values of these two coordinates will depend on the method used to estimate the length of a link. When nodes' geographic locations are unknown, all but the hops coordinate are difficult to obtain. In the absence of nodes' geographic location information, one must estimate the length of each link to calculate path-distance. We call this estimated path-distance coordinate the *extrapolated* coordinate, and use it for GHVR. A node can estimate the length of its link to another node using the strength of signal received from that node. As noted in [24], using the received signal strength, however, needs to sample the environment to train the distance measurement system due to the lack of an accurate mathematical model of the environment. Usually, an estimation error cannot be improved significantly without using a complex model of the environment.

Fortunately, GHVR doesn't require such a complex model since it only require proportional proximity among nodes for routing packets. Moreover, we need to devise as simple a link-length estimation method as possible for ease of implementation without modifying wireless network device drivers.

To meet the above needs, we develop a simple

Input: an ETX probe piggybacked with the periodic HELLO and a locally-observed data transmission result.
Data: a link table $\mathcal{L}$

1 $l \leftarrow \mathcal{L}$.lookup(nb);
2 **if** *upon receiving an ETX probe* e **then**
3 $\quad l$.update_bwdrate(); /* d_r */
4 $\quad l$.fwdrate $\leftarrow e$.bwdrate; /* d_f */
5 $\quad dlvrate \leftarrow l$.fwdrate $\times\, l$.bwdrate;
6 **if** *upon receiving an ACK event* a **then**
7 $\quad dlvrate \leftarrow 1/a$.txattempts;
8 **if** *upon receiving a FAILURE event* f **then**
9 $\quad dlvrate \leftarrow 1/(f$.txattempts $+ 1)$;
10 $\quad l$.dlvrate $\leftarrow$
$\quad \alpha \times dlvrate + (1-\alpha) \times l$.dlvrate;
11 $\quad$ return;
12 $l.dlvrateFilter$.put($dlvrate$);
13 l.dlvrate $\leftarrow \alpha \times l.dlvrateFilter$.get() + $(1-\alpha) \times l$.dlvrate;

Algorithm 1: Hybrid link-quality measurement used to estimate the per-link length.

link-length estimation method based on the link-quality that represents the packet reception ratio over a link. Given the link-quality (dlvrate), we assume that dlvrate $\propto 1/d^2$ with respect to distance (d). Our choice for this exponent is based on the fact that the received signal strength attenuates proportionally to $1/d^2$ in open space and that the received signal strength directly affects the receiver's successful packet reception. Use of other path exponents is also possible. This link-length estimation may be less accurate than other more accurate (but expensive) methods, but our evaluation in Section IV shows that GHVR with the proposed estimation method performs well, even when compared to that with true link-lengths.

Every node runs the link-quality measurement scheme described in Algorithm 1. This link-quality measurement scheme is based on the well-known ETX [17] which uses periodic broadcasts and is calculated for each link using the forward-delivery ratio d_f and the backward-delivery ratio d_r such that ETX=$(d_f d_r)^{-1}$. Since broadcast-based ETX does not reflect properly the link-quality for unicast, dlvrate in Algorithm 1 adjusts the broadcast-based link-quality with each success or failure result of unicast. Using dlvrate $\propto 1/d^2$, our link-length estimate from dlvrate between two nodes is $\sqrt{\text{dlvrate}^{-1}}$.

Our heuristics for determining the height or coordinate of a node with respect to a particular landmark then

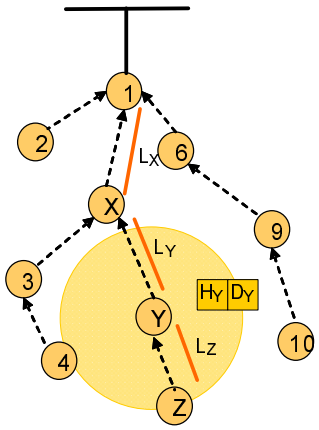


Fig. 3. After overhearing that node Y advertises its current height (H_Y) and height-determinant (D_Y), node Z updates its own H_Z and D_Z based on both the measured length-estimate (L_Z) and the advertised information (H_Y and D_Y).

takes an induction step over the minimum-cost path.[2] For immediate 1-hop neighbors of the landmark, their initial height relative to the landmark is assigned their initial length-estimate to the landmark. Referring to Fig. 3, we assume that node Z has node Y as the next hop on its minimum-cost path to the landmark (node 1), and that Y broadcasts periodically its current height (H_Y) and height-determinant (D_Y), whose meaning is the sum of the height-determinant advertised by a parent node and the link length-estimate made by itself. Every neighbor of Y then updates the quality of its link to Y.

Besides updating the link-quality, Z updates H_Z and D_Z of its own since Y is the next hop on the minimum-cost path from Z to the landmark:

$$H_Z \leftarrow H_Y + L_Z H_Y / D_Y \text{ and } D_Z \leftarrow D_Y + L_Z \quad (1)$$

where L_Z is the length-estimate made by Z for the link between Y and itself.

Link-quality measurements are in general noisy and hence need to be smoothed. Algorithm 1 is 'sensitized' to the latest link-quality measurement by using a large smoothing constant α and eliminates instantaneous noises in the measurement by using the *median filter* [25] (lines 11 $\sim$ 12). Currently, α is set to 0.5 and a 3-span median filter that eliminates one spike in the input is used. Likewise, other measurements such as length-estimate, height, height-determinant, and costs are also smoothed.

C. Distance Function

The distance function used by GHVR is the standard Euclidean distance function over the landmark-based coordinate space. However, since the landmark-based coordinates are not necessarily orthogonal to each other, one might wonder whether the standard Euclidean distance function can still serve as a good distance metric

[2]We assume that the link cost is dlvrate^{-1}, even though any definition of the link cost can be used.

to represent the proximity among nodes in the landmark-based coordinate space.

Given two l-dimensional virtual coordinates $\underline{u} = (u_1, u_2, \ldots, u_l)$ and $\underline{v} = (v_1, v_2, \ldots, v_l)$, the following distance functions $\delta(\underline{u}, \underline{v})$ have been introduced in the literature: (1) the standard Euclidean distance, (2) the *Centered* distance [10] $\delta(\underline{u}, \underline{v}) = \Sigma((u_i^2 - \ddot{u}) - (v_i^2 - \ddot{v}))^2$ where $\ddot{u} = \Sigma u_i^2/l, \ddot{v} = \Sigma v_i^2/l$, (3) the *Weighted* Manhattan distance [11] $\delta(\underline{u}, \underline{v}) = 10 \times \Sigma_{i \in C_k(v)} \max(u_i - v_i, 0) + \Sigma_{i \in C_k(v)} \max(v_i - u_i, 0)$, where $C_k(v)$ is the set of k closest (in number of hops) landmarks to node v.

The term Weighted Manhattan distance is defined in this study, but not in [11]. Also, note that it violates the definition of the metric since it is not symmetric, i.e., $\delta(\underline{u}, \underline{v}) \neq \delta(\underline{v}, \underline{u})$. Unfortunately, these distance functions have not been comparatively evaluated against each other.

D. Greedy Forward

Packets are forwarded greedily to one of nodes in the neighborhood. Formally represented in a graph (V, E) with a given distance function $\delta : V \times V \rightarrow \mathbb{R}^+$ in the landmark-based coordinate space, node c forwards packets greedily to its neighbor $i \in N(c)$ such that $i = \arg\max_{x \in N(c)}(\delta(c, d) - \delta(x, d))/C_{cx}$ and $\delta(c, d) - \delta(i, d) > 0$, i.e., making positive forward progress, where $N(c)$ is the set of node c's neighbors, and C_{cx} is the link cost between nodes c and x. Note that the distance function used by GHVR is the standard Euclidean distance function over the landmark-based coordinate space.

The above greedy forwarding strategy is in fact NADV [8] over landmark-based coordinates, which is known to be the optimal forwarding strategy for geographic routing with lossy wireless links. If the positive forward progress condition does not exist, the greedy forwarding cannot guarantee loop-freedom in the path.

E. Recovery from Greedy Failure

GHVR can suffer from the *void problem* [6], [7] as geographic routing, although WMNs are likely to be densely populated. The void problem refers to a situation where no neighbors are geographically closer to the destination than the current node. In such a case, the greedy forward fails since there is no next hop that makes positive progress toward the destination.

Several recovery procedures against the void problem in the landmark-based coordinate space have been proposed in the literature. LCR [9] backtracks to the best upstream node, whereas BVR [11] backtracks to the parent toward the landmark closest to the destination and makes that landmark flood the route request whenever necessary. Unlike LCR or BVR, GLIDER [10] never backtracks or falls back on the closest landmark; it always floods packets whenever it encounters a greedy failure.

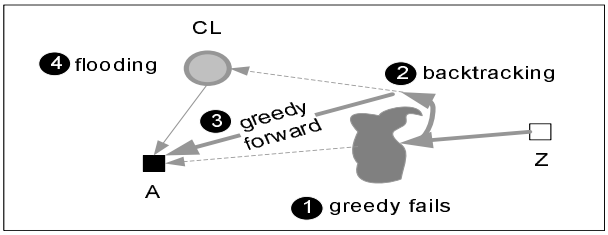


Fig. 4. Illustration of recovery from a greedy failure.

TABLE I
CONFIGURATIONS FOR 802.11B WITH THE SHADOWING MODEL.

Path loss exponent		4	
Shadowing deviation		4,5,7,9	
Reference distance		10	
Frequency	2.4 GHz	Transmit power	15 dBm
CPThresh	10	CSThresh	-105 dBm
RXThresh (11 Mbps)†		-79.84 dBm, (95% at 50m)	
RXThresh (5.5 Mbps)†		-85.68 dBm, (95% at 70m)	
RXThresh (2 Mbps)†		-90.05 dBm, (95% at 90m)	
RXThresh (1 Mbps)†		-94.31 dBm, (95% at 115m)	
Data Rate		11 Mbps (w/o auto rate)	
Basic Rate		2 Mbps	
MAC headers		(24,14,14,28) bytes @ Basic Rate	
PLCP length		192 bits @ 1 Mbps	
SIFS	10us	DIFS	50us

† These thresholds are configured for the path loss exponent 4 and the shadowing deviation 4, and remain unchanged regardless of changes in both parameters.

The recovery mechanism used by GHVR is based on BVR's solution. Fig. 4 illustrates this mechanism. While backtracking to the destination's CL, greedy forwarding will be resumed whenever there exists a greedy next hop toward the destination. Packets may get to CL before reaching the destination. In such a case, CL of the destination performs efficient flooding based on the history of packet forwarding to CL. Note that a history of packet forwarding to CL can be accumulated since every node periodically publishes its identifier (ID) and landmark-based coordinate to the nearest landmark, as illustrated in Fig. 1(a).

IV. EVALUATION

A. Wireless Channel

To make our *ns-2* simulation as realistic as possible, we followed several recommendations in [26], one of which is to use the shadowing propagation model, because it can simulate wireless link characteristics such as link asymmetry and time-varying link-quality. The *ns-2* shadowing model and other configurations are set up as in Table I; these settings are derived from our implementation of the Proxim Orinoco adapter [27] for a semi-open environment.

B. Topologies and Flows

An MIT indoor wireless sensor network testbed called MistLab [28] is reproduced as an unplanned topology in our *ns-2* simulation. The simulated MistLab, simply

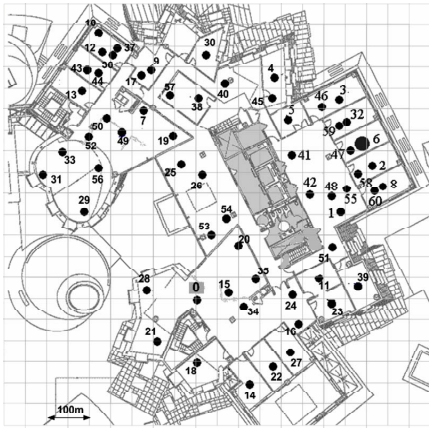


Fig. 5. MistLab [28], a non-uniform real-world wireless sensor network testbed with 61 nodes over a field of 16,000 square feet, is reproduced in our *ns-2* simulation as an nsMistLab 802.11b wireless network over a field of $1km \times 1km$.

called nsMistLab, has sixty-one 802.11b nodes spread non-uniformly over a square field as shown in Fig. 5. Besides this basic topology, 100 nodes are randomly added in the field. A landmark is also installed at each of the four corners of the field.

Every node except node 0 becomes a data sink. Node 0 sends continually an average MAC-level payload of 1500 bytes, or 1500+$U(-750, 750)$ bytes, to every other node in the network. Data is sent every 4 seconds while changing the destination in a round-robin manner.

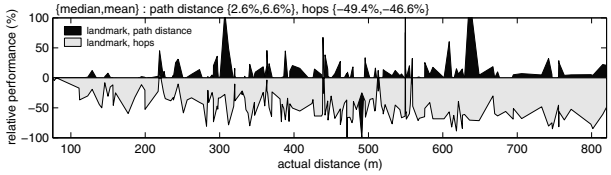
C. Performance Metrics and Questions

Since the primary goal of this paper is to compare continuous and discrete coordinates, only the greedy forwarding defined in Section III-D is considered because recovery from a greedy failure has nothing to do with the type of coordinate used. We also limit the maximum number of transmission attempts per packet at each hop. Thus, packets suffering from too many transmission failures at each hop will be dropped after reaching a pre-defined retry limit. Currently, the retry limit is arbitrarily set to 40 per packet at each hop. Throughout the simulation, we refer to this bounded greedy forwarding as simply *greedy routing*.

Under this greedy routing, the main performance metric to evaluate is the delivery ratio. All the virtual coordinates and the absolute coordinate described in Fig. 2 are evaluated. The delivery ratio under each type of coordinate is compared to that under the extrapolated coordinate, and relative performance improvements in terms of median and mean are presented.

Our evaluation will primarily answer the following questions:

- By how much does the extrapolated coordinate outperform the hops coordinate?
- By how much does the ideal path-distance coordinate outperform the extrapolated coordinate?



(a)

	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
set	m	u	m	u	m	u	m	u
0	9.72	21.93	5.97	14.71	-49.38	-46.56	2.61	6.58
1	0.00	45.28	-4.14	8.51	-57.99	-53.70	-2.44	12.27
2	13.90	27.06	11.11	22.81	-76.09	-70.85	17.80	304.41
3	24.62	111.54	25.00	56.91	-71.47	-67.89	21.00	304.19
4	0.00	7.39	0.00	-6.09	-56.82	-46.80	0.00	4.33
5	8.11	20.01	9.84	19.27	-53.64	-50.59	4.36	10.59
6	0.00	-14.12	0.00	-19.81	-58.90	-58.07	2.90	9.94
7	1.22	17.75	6.07	18.47	-56.10	-52.93	2.38	14.69
8	10.82	42.70	10.82	50.47	-64.49	-57.60	2.60	15.90
9	-19.05	-32.69	-22.95	-31.22	-51.95	-49.67	2.56	8.83

(b)

Fig. 6. The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate over 10 different topologies. The m and u represent the median and the mean of comparison results. Node-by-node comparison results for the set 0 in (b) are plotted in (a).

- Is the standard Euclidean distance function still a good distance metric in the landmark-based virtual coordinate space?

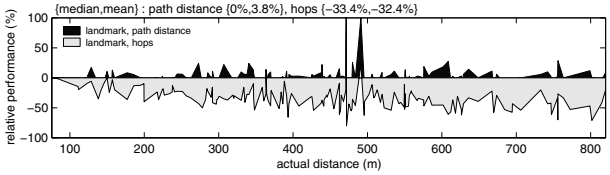
D. Main Results

Fig. 6(a) shows the node-by-node comparisons obtained by averaging the results of five independent runs with different random seeds for the same topology set 0. The x-axis is sorted according to the distance between node 0 and each destination. Fig. 6(a) indicates that the delivery ratio under the extrapolated coordinate is improved by a factor of almost 2 over the delivery ratio under the hops coordinate in both median and mean numbers.

Evaluation against additional 9 different topologies did not change this conclusion. Fig. 6(b) suggests that the extrapolated coordinate consistently outperforms the hops coordinate over these different topologies. The performance gains from the extrapolated coordinate over the hops coordinate are greater than a factor of 2 in most cases. Moreover, the extrapolated coordinate loses only 2$\sim$4% of performance over the ideal path-distance coordinate all cases but sets 2 and 3.

It is interesting to see that the greedy routing seems to perform slightly better under the path-distance coordinate than under the absolute coordinate. This might open a new design space for improving the standard geographic routing even if the nodes' geographic locations are easily obtainable.

Note that the hops coordinate can also be defined over the minimum-hop path [9], [10]. Related simulation results, not presented in this paper, show that using the minimum-hop path is not as good as using the minimum-cost path. Thus, defining landmark-based coordinates



(a)

	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
set	m	u	m	u	m	u	m	u
0	2.67	7.87	3.65	8.16	-33.42	-32.37	0.00	3.73
1	2.50	11.44	0.00	4.82	-47.66	-46.64	0.00	1.65
2	5.13	2.59	5.26	2.27	-37.14	-36.23	2.60	5.15
3	2.85	12.12	5.13	13.24	-30.49	-29.32	0.00	9.23
4	0.00	-3.52	0.00	-3.41	-39.43	-38.04	0.00	-0.15
5	2.74	7.70	5.13	8.31	-38.35	-37.09	0.00	3.06
6	0.00	-4.79	0.00	-6.79	-42.87	-41.25	2.50	3.17
7	5.72	6.10	3.08	5.95	-35.51	-34.62	0.00	1.41
8	5.79	10.96	7.40	10.19	-44.44	-43.85	2.63	3.60
9	-7.69	-23.81	-6.78	-23.04	-36.99	-38.64	0.00	2.59

(b)

	S1	S2	S3	S4	S5	S6
path loss	4	4	4	4	4	4
deviation	4	9	9	7	7	7
node	–	–	–	G0+G1	–	–

	S7	S8	S9	S10	S11	S12
path loss	4	4	4	4	4	4
deviation	7	5	5	5	5	4
node	–	G0+G1	–	–	–	G0

(c)

Fig. 7. The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate over 10 different topologies. Node-by-node comparison results for the set 0 in (b) are plotted in (a). The wireless channels are changed according to (c).

along the minimum-hop path will not be considered.[3]

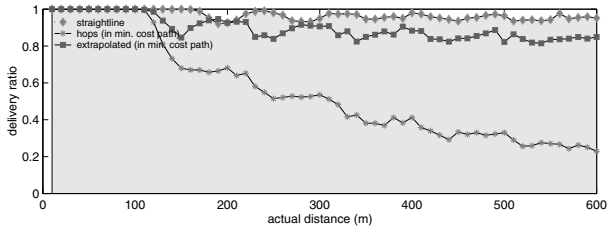
E. Impact of Wireless Channel Variations

Fig. 7(a) and (b) suggest that the extrapolated coordinate still outperforms the hops coordinate even in the face of variations in wireless channel quality and the random packet drops at some nodes. Interestingly, the extrapolated coordinate is found to be comparable to the ideal path-distance coordinate in this experiment.

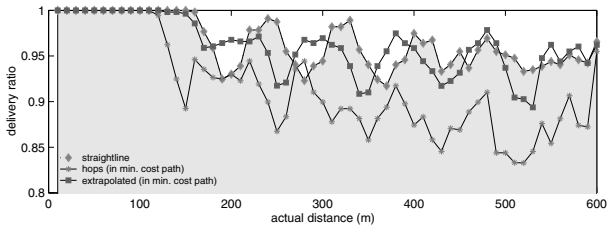
These results are obtained with the wireless channel varied as follows. A total of 12 scenarios, each lasting 500 seconds, are considered on top of the basic wireless channel setting specified in Table I. Each scenario, denoted by the labels (S0, S1, ..., S12) in Fig. 7(c), is rolled out over the entire simulation of length 6,000 seconds. Changes are made in the wireless channel parameter, the node parameter, and the random packet-loss parameter at the physical layer.

In Fig. 7(c), a large path loss (exponent) indicates a sharp drop in the average received signal strength over distance. A large (shadowing) deviation in the figure means more smoothed exponential decay but more variations in the received signal strength. The node constraint

[3]For interested readers, the hops coordinate over the minimum hop count yields the following relative performance as compared to the extrapolated coordinate for 10 different sets; {(m%,u%)}={(-71.43, -64.60), (-85.71, -78.97), (-85.62, -73.86), (-92.85, -85.08), (-85.53, -76.07), (-81.23, -73.24), (-85.14, -76.54), (-82.91, -74.73), (-87.33 -78.57), (-87.49, -76.15)}.



(a) delivery ratio under one landmark at 0



(b) delivery ratio under two landmarks at 0 and 600

Fig. 8. Performances of the *connectivity-first greedy routing* with a line topology under the straightline, hops, and extrapolated coordinates. Note the scale of Y axis in (b).

"–" means that any change will be global across all the nodes, whereas two node constraints G0 and G1 mean that a change will be applied only to those nodes satisfying G0 or G1.

Selection of nodes subject to both constraints is arbitrary. For example, the node constraint G0 that changes the shadowing deviation parameter will be applied to those nodes whose IDs are 1 (modulo 3). Similarly, the node constraint G1, which drops the received packets randomly at the physical layer with a probability of 0.3, will be applied to the nodes whose IDs are 54, 20, 35, 24, 11, 51, 42, 41, 45, and 26, regardless of the type of topologies used. Note that the node constraint G1 is not applied in the last scenario S12.

F. Impact of Greedy Routing Variations

We define an enhanced version of the greedy routing, called *connectivity-first greedy routing* (CFGR), which trades optimality for reliability in link selection. CFGR maintains and uses a routing table that consists of neighbors whose measured bidirectional link-quality is known to be above some threshold. Thus, unlike the original greedy routing, it attempts to directly transmit packets if the destination is found in the routing table. Otherwise, it follows the original greedy routing. Note that evaluation under the original greedy routing represents the lowest performance that could be attained without modifying the optimal link-selection strategy described in Section III-D.

Using CFGR under the hops coordinate turns out to significantly improve the delivery ratios for immediate neighbors, as seen in Fig. 8(a). This is because nodes keep transmitting packets within the maximum retry limit until the transmission becomes successful. However, we still observe performance decline over distance. Such a decline is probably due to the disparities in trans-

	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
set	m	u	m	u	m	u	m	u
0	2.63	4.88	2.57	4.16	-28.95	-28.50	0.00	1.00
1	5.31	17.85	2.53	8.61	-31.14	-31.31	-1.22	-0.16
2	5.48	4.85	5.13	-0.29	-26.49	-24.11	2.81	5.26
3	4.35	8.57	2.99	6.65	-25.32	-25.18	2.50	3.81
4	2.36	0.61	0.00	0.54	-27.15	-26.62	2.50	4.53
5	3.08	8.98	2.63	8.67	-29.57	-29.15	0.00	2.63
6	0.00	-3.05	0.00	-6.38	-34.22	-33.20	2.53	5.38
7	3.64	6.32	3.59	7.11	-31.14	-29.29	0.39	4.03
8	7.90	10.93	5.56	9.85	-40.61	-36.23	0.13	0.28
9	-2.90	-21.36	-7.91	-22.75	-31.47	-31.88	2.56	3.33

Fig. 9. The improved(+) or reduced(-) delivery ratios (%) of the *connectivity-first* greedy routing relative to the extrapolated coordinate over 10 different topologies.

	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
set	m	u	m	u	m	u	m	u
0	2.50	3.46	2.50	3.98	-22.50	-23.71	0.00	-0.52
1	3.92	17.74	2.67	16.65	-39.15	-38.76	0.00	1.06
2	2.50	2.60	2.53	3.98	-26.60	-27.14	0.00	1.64
3	2.56	6.21	2.56	7.57	-20.00	-21.61	0.00	4.46
4	2.50	8.57	0.00	4.55	-23.73	-25.10	0.00	-0.16
5	2.50	6.75	2.50	6.51	-21.95	-22.92	0.00	1.53
6	2.50	5.64	2.50	4.82	-24.39	-25.90	0.00	2.28
7	2.50	2.24	2.50	3.10	-25.32	-26.40	0.00	0.78
8	2.50	7.07	2.50	5.73	-28.58	-29.68	0.00	1.71
9	0.00	1.07	0.00	2.00	-25.00	-27.72	0.00	0.32

Fig. 10. The improved(+) or reduced(-) delivery ratios (%) of the greedy routing relative to the extrapolated coordinate when a variant of the best auto rate algorithm [29] is used.

mission range between broadcast and unicast. Recall that the hops coordinate between nodes is likely to differ when nodes are separated by the broadcast coverage. However, the broadcast coverage is much wider than that of unicast in the IEEE 802.11-based radio. Thus, the next hop in the hops coordinate is likely to be situated near the perimeter of broadcast coverage of the current node. This, in turn, increases the chance of dropping unicast packets.

Using additional landmarks mitigates this problem to some degree, because it produces finer-grained coordinates even under the hops coordinate. Nevertheless, performance under the hops coordinate is still the poorest among the virtual coordinates studied, as shown in Fig. 8(b). This is also confirmed by [11]; according to the simulation results in Section 4.2 of this paper, 10 destination-dependent landmarks out of 30 randomly-elected landmarks achieved the greedy routing performance comparable to that of using true positions.

Additional simulations over nsMistLab support our claim that the extrapolated coordinate is better than the hops coordinate. As shown in Fig. 9, the hops coordinate consistently yields about 30% less performance than the extrapolated coordinate.

G. Impact of Auto Rate Algorithms

To eliminate potential packet drops due to the disparity between the ranges of broadcast and unicast in IEEE 802.11, we implemented a variant of the receiver-based auto rate selection algorithm described in [29]. In the receiver-based rate selection algorithm, a receiver determines the best rate for each neighbor by considering

	absolute		landmark-based virtual coordinate					
	geo. locations		straightline		hops		path distance	
set	m	u	m	u	m	u	m	u
0	2.50	3.92	2.50	4.07	-7.88	-11.11	0.00	0.87
1	2.56	33.08	2.50	14.64	-17.58	-20.63	0.00	0.62
2	2.50	5.93	2.50	5.33	-10.25	-11.58	0.00	3.43
3	2.50	5.34	2.50	5.60	-7.50	-10.23	0.00	2.16
4	0.00	1.27	0.00	0.35	-14.81	-15.85	0.00	0.09
5	2.50	6.37	2.50	6.11	-10.58	-11.83	0.00	2.17
6	2.50	3.77	0.00	3.18	-14.63	-17.15	0.00	0.92
7	2.50	2.91	2.50	2.89	-16.44	-17.58	0.00	-1.11
8	2.50	4.58	2.56	4.42	-18.21	-20.74	0.00	0.54
9	0.00	0.18	0.00	0.21	-10.52	-15.08	0.00	1.32

Fig. 11. The improved(+) or reduced(-) delivery ratios (%) of the *connectivity-first* greedy routing relative to the extrapolated coordinate when a variant of the best auto rate algorithm [29] is used.

the channel conditions and informs a sender of the best rate. The original auto rate selection in [29] takes steps to reserve the channel before sending data, which allows the receiver to inform the sender of the best rate right before actual data delivery. On the other hand, our implementation requires every node to periodically advertise the best unicast rate for each of its neighbors by using the same logic as in [29], but based on recently-measured channel conditions.

Fig. 10 shows that the chosen auto rate algorithm improves the delivery ratio across all virtual coordinates. Nevertheless, there are still noticeable differences between the hops and extrapolated coordinates. At the same time, the differences between the ideal path-distance and the extrapolated coordinates become indiscernible, because only the ideal path-distance performs better (by about 2%) than the extrapolated, and both are the same in terms of the median.

Running the connectivity-first greedy routing with the auto rate feature enabled is shown (Fig. 11) to reduce the performance gap between the other coordinates and the extrapolated coordinate. However, as already seen in Fig. 10, the performance under the hops coordinate is still consistently inferior to that under the extrapolated coordinate, whereas the performance of the extrapolated coordinate approaches that of the ideal path-distance coordinate.

Since using either the connectivity-first greedy routing or the auto-rate feature consistently demonstrates the performance superiority of the extrapolated coordinate, our remaining evaluation is done with the original greedy routing without enabling the auto rate feature.

H. Impact of Length-estimate

Given the same wireless channel variations as in Fig. 7(c), different length-estimates are evaluated in Fig. 12. Note that given the link-quality (`dlvrate`), the default length-estimate is $\sqrt{\texttt{dlvrate}^{-1}}$ as defined in Section III-B. The results in this figure suggest that $\sqrt{\texttt{dlvrate}^{-1}}$ yields slightly better performance than $\texttt{dlvrate}^{-1}$. However, using $\texttt{dlvrate}^{-2}$ turns out to widen the performance gap between the ideal path-distance and the extrapolated. Note that this experiment

set	path-distance vs. extrapolated virtual coordinates					
	sqrt		identity		quadratic	
	m	u	m	u	m	u
0	0.00	3.73	3.23	5.42	63.06	89.25
1	0.00	1.65	0.00	5.19	58.43	175.35
2	2.60	5.15	0.00	6.46	76.72	126.11
3	0.00	9.23	5.98	18.53	46.99	71.78
4	0.00	-0.15	0.00	-0.05	64.08	123.99
5	0.00	3.06	2.67	5.88	46.90	64.12
6	2.50	3.17	5.13	8.07	72.07	96.95
7	0.00	1.41	0.00	1.92	51.72	91.58
8	2.63	3.60	0.00	-0.41	51.00	86.25
9	0.00	2.59	0.00	6.17	77.27	111.59

Fig. 12. The improved(+) or reduced(-) delivery ratios of the greedy routing relative to the extrapolated coordinate when its length-estimate is changed $\sqrt{c}$ (default), c, c^2 where $c = \texttt{dlvrate}^{-1}$.

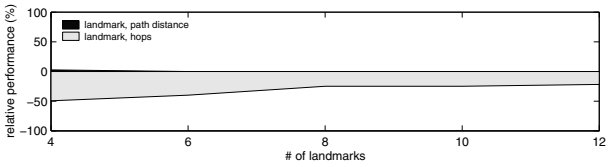


Fig. 13. The greedy routing performance (in terms of the median value) relative to the extrapolated coordinate with the number of landmarks varied.

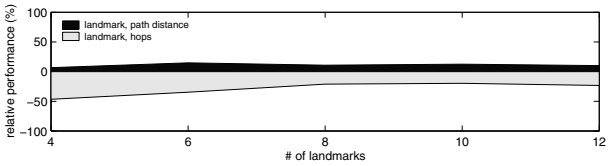


Fig. 14. The greedy routing performance (in terms of the mean value) relative to the extrapolated coordinate with the number of landmarks varied.

affects performance only under the extrapolated coordinate.

I. Impact of Placement and Number of Landmarks

The simulation result (Sec. 4.2 in [11]) suggests that there is a trade-off between the number of landmarks and routing performance as well as per-packet overhead. For example, it is shown in [11] that about 10 beacons need to be installed when landmarks are randomly elected. However, the planned installation of landmarks is shown to be much better than a large number of randomly-elected landmarks. Sec. 4.1.2 in [9] shows that a landmark at each of the four corners of a rectangle is enough to achieve high routing performance, and that having more than 4 regularly-planned landmarks makes no difference in improving routing performance.

Based on these observations, a landmark is pre-established at each of the four corners of the field, which is the default in our simulation, and then up to 8 additional landmarks are randomly placed. This mixed configuration helps us assess the impact of the placement of landmarks as well as the impact of the number of landmarks on the routing performance.

Figs. 13 and 14 summarize the results: the extrapolated coordinate works much the same as the path-distance coordinate. The median performance comparison shows that the difference between the two is indiscernible, even with random installation of additional landmarks. Although the path-distance performs 2.6% better than the extrapolated when 4 landmarks are used, the two are otherwise the same. As for the mean performance, the path-distance shows slightly improved performance as the number of landmarks increases from 4 to 12 by 2: (6.58%, 14.71%, 10.83%, 12.44%, 10.15%).

As expected in [11], the hops coordinate performs better as the number of landmarks increases. Two points are noticeable from Figs. 13 and 14. First, installing additional random landmarks does not appear helpful even for the hops coordinate; installing landmarks regularly seems more important than installing a large number of them randomly. Second, in spite of additional landmarks the hops coordinate yields poorer performance than the extrapolated coordinate; its median and mean are (-49.38%,-40.00%, -25.00%, -25.00%, -21.82%) and (-46.56%, -34.72%, -21.09%, -19.80%,-23.28%) as the number of landmarks increases from 4 to 12 by 2, respectively.

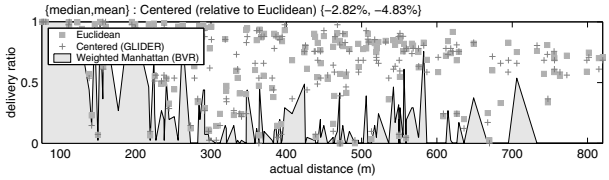


Fig. 15. The delivery ratios of the greedy routing under different distance functions with the ideal path-distance coordinate.

J. Impact of Distance Function

The authors of [9] concluded that the Euclidean distance is the better metric by comparing it with the usual Manhattan distance on the unit disc radio model. Our comparison results for the distance functions are plotted in Fig. 15 and indicate that the Euclidean distance improves routing performance more than Weighted Manhattan or Centered distance. Recall that this distance function comes into play when selecting a next hop that makes the most efficient forward progress as defined in Section III-D. The poor performance of the "Weighted Manhattan" function in BVR seems to be related to the asymmetric weight factor in its distance computation. The median and mean performances of the "Centered distance" in Fig. 15 indicate that its theoretical assumption of "achieving the non-local minima at all (Sec. V.B [10])" does not hold for a realistic wireless channel.

V. Conclusion

We first addressed the lack of a fine-grained landmark-based coordinate for routing packets in wireless networks because the existing work uses the hop count along the minimum hop or cost path to the elected landmarks. We then presented a one-to-one routing protocol based

on a novel continuous landmark-based coordinate. The proposed coordinate is called the *extrapolated coordinate* and measures the length along the minimum-cost path to the elected landmarks. We further defined the ideal path-distance coordinate and comparatively evaluated three schemes — the traditional hop count, the proposed extrapolated, and the ideal path-distance — with realistic wireless network settings. Our simulation results demonstrated that the extrapolated coordinate improved routing performance by a factor of almost 2, over the traditional hop count coordinate, and performs nearly as well as the ideal path-distance coordinate. We also investigated the different types of distance functions over the landmark-based coordinate space. Our evaluation indicated that the standard Euclidean distance definition still works best even in the virtual coordinate space.

Our future work will focus on the implementation aspects of the proposed routing protocol in a real testbed. We will also compare the proposed routing protocol with those traditional distance-vector or link-state routing protocols that are being considered for wireless mesh networks. Finally, we will study the design of a scalable location service over virtual coordinate space. Unfortunately, all of the scalable location servers proposed in the literature are only feasible when actual geographic locations are available.

REFERENCES

[1] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *SIGCOMM '94*. New York, NY, USA: ACM Press, 1994, pp. 234–244.

[2] C. A. Santiváñez, R. Ramanathan, and I. Stavrakakis, "Making link-state routing scale for ad hoc networks," in *MobiHoc '01*. New York, NY, USA: ACM Press, 2001, pp. 22–32.

[3] IETF, "Optimized link state routing protocol (olsr)," http://tools.ietf.org/html/rfc3626.

[4] ——, "Topology dissemination based on reverse-path forwarding (tbrpf)," http://tools.ietf.org/html/rfc3684.

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

[6] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *DIALM '99*. ACM Press, 1999, pp. 48–55.

[7] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM Press, 2000, pp. 243–254.

[8] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *MobiHoc '05*. New York, NY, USA: ACM Press, 2005, pp. 230–241.

[9] Q. Cao and T. Abdelzaher, "A scalable logical coordinates framework for routing in wireless sensor networks," in *the IEEE RTSS*, 2004.

[10] Q. Fang, J. Gao, L. J. Guibas, V. de Silva, and L. Zhang, "Glider: gradient landmark-based distributed routing for sensor networks," in *IEEE INFOCOM*, 2005.

[11] R. Fonseca, S. Ratnasamy, J. Zho, C. T. Ee, D. Culler, S. Shenker, and I. Stoica, "Beacon vector routing: Scalable point-to-point in wireless sensornets," in *NSDI*, 2005. [Online]. Available: citeseer.ist.psu.edu/fonseca04beacon.html

[12] H. Lim, J. C. Hou, and C.-H. Choi, "Constructing internet coordinate system based on delay measurement," in *IMC '03*. New York, NY, USA: ACM Press, 2003, pp. 129–142.

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

[14] P. Bose, P. Morin, I. Stojmenoviá, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wirel. Netw.*, vol. 7, no. 6, pp. 609–616, 2001.

[15] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2004, pp. 108–121.

[16] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2003, pp. 1–13.

[17] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *MobiCom '03*. ACM Press, 2003, pp. 134–146.

[18] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *SIGCOMM '04*. New York, NY, USA: ACM Press, 2004, pp. 121–132.

[19] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *SIGCOMM '04*. New York, NY, USA: ACM Press, 2004, pp. 133–144.

[20] ——, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04*. New York, NY, USA: ACM Press, 2004, pp. 114–128.

[21] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *MobiCom '05*. New York, NY, USA: ACM Press, 2005, pp. 31–42.

[22] C. E. Koksal and H. Balakrishnan, "Quality-aware routing metrics for time-varying wireless mesh networks," in *IEEE Journal on Selected Areas in Communications, Vol. 24, No. 11*. New York, NY, USA: ACM Press, November, 2006, pp. 1984–1994.

[23] K.-H. Kim and K. G. Shin, "On accurate measurement of link quality in multi-hop wireless mesh networks," in *MobiCom '06*. New York, NY, USA: ACM Press, 2006, pp. 38–49.

[24] E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: A comparative study," in *SECON 2004*.

[25] D. C. Montgomery, L. A. Johnson, and J. S. Gardiner, "Forecasting and time series analysis," 2nd Edition, McGraw-Hill Book Company, 1990.

[26] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *MSWiM '04*. New York, NY, USA: ACM Press, 2004, pp. 78–82.

[27] Proxim, "Orinoco 11b pci adapter," http://www.proxim.com/products/wifi/client/ 802.11b_pci/.

[28] MIT, "Mistlab," http://mistlab.csail.mit.edu/.

[29] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *MobiCom '01*. New York, NY, USA: ACM Press, 2001, pp. 236–251.