

Efficient Geographic Routing in Multihop Wireless Networks

Seungjoon Lee Bobby Bhattacharjee

Department of Computer Science
University of Maryland
College Park, MD 20742 USA
{slee, bobby}@cs.umd.edu

Suman Banerjee

Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706 USA
suman@cs.wisc.edu

Technical Report CS-TR 4625, University of Maryland, College Park

Abstract—We propose a new link metric called *normalized advance (NADV)* for geographic routing in multihop wireless networks. NADV selects neighbors with the optimal trade-off between proximity and link cost. Coupled with the local next hop decision in geographic routing, NADV provides an adaptive and efficient cost-aware routing strategy. Depending on the objective or message priority, applications can use the NADV framework to minimize various types of link cost.

In this paper we present efficient methods for link cost estimation and perform detailed simulations in diverse scenarios. Our results show that NADV outperforms current schemes in many aspects: for example, in high noise environments with frequent packet losses, the use of NADV leads to 83% higher delivery ratio. When compared to centralized routing, geographic routing using NADV finds paths whose cost is close to the optimum.

Index Terms—System design, Simulations

I. INTRODUCTION

Geographic routing (or position-based routing) uses location information for packet delivery in multihop wireless networks [1], [2], [3], [4], [5]. Neighbors locally exchange location information obtained through GPS (Global Positioning System) or other location determination techniques [6]. Since nodes locally select next hop nodes based on this neighborhood information and the destination location, neither route establishment nor per-destination state is required in geographic routing. As large-scale sensor networks become more feasible, properties such as stateless nature and low maintenance overhead make geographic routing increasingly more attractive [7]. Also, location-based services such as geocasting [8] can be best realized using geographic routing.

The most popular strategy for geographic routing is simply forwarding data packets to the neighbor geographically closest to the destination [1], [2], [3]. Although this *greedy* method is effective in many cases, packets may get routed to where no neighbor is closer

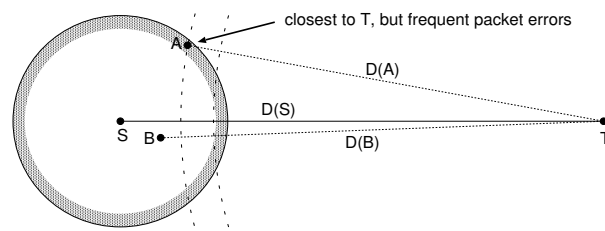


Fig. 1. An example scenario for geographic routing. While node A is closest to T , it is experiencing a high packet error rate. Consequently, higher performance can be achieved if S forwards packets to B .

to the destination than the current node. Many *recovery* schemes have been proposed to route around such *voids* for guaranteed packet delivery as long as a path exists [1], [2], [3], [9]. These techniques typically exploit planar subgraphs (i.e., Gabriel graph, Relative Neighborhood graph), and packets traverse faces on such graphs using the well-known *right-hand* rule. Most geographic routing protocols use one-hop information, but generalization to two-hop neighborhood is also possible [10].

In this paper we propose the use of a new link metric called *normalized advance (NADV)* in geographic routing. Instead of the neighbor closest to the destination, NADV lets us select the neighbor with the best trade-off between link cost and proximity. In Figure 1, for example, although A is closest to destination T , it is experiencing high packet error probability. B is slightly farther from T than A , but provides a higher quality link. In this scenario, forwarding packets to B is better, and NADV chooses B over A . We show that a path chosen by NADV approaches the optimal minimum cost path in networks with sufficiently high node density. Our proposed metric is best understood in the context of greedy mode in geographic routing, but it can also be used with schemes that route around voids [3], [9].

Coupled with the localized nature of geographic routing, the use of NADV provides a unique opportunity for

adaptive routing — a feature not offered by most existing on-demand routing protocols. For example, suppose that an on-demand routing protocol (e.g., DSR [11]) finds a minimum cost path and that the link costs change while the path is in use (e.g., due to mobility or environment changes). If the source wants to find a better path, it needs to flood a new route discovery request. This solution may incur high control overhead, and it is also difficult to know when the source should initiate the flooding. In contrast, as long as link cost estimation schemes can track link costs change, NADV immediately reflects the change, which in turn would result in the selection of best next hop in geographic routing.

We present NADV in the context of a general framework for efficient geographic routing. To our best knowledge, only the *SP-Power* scheme in [12] considers link costs in selecting next hops in geographic routing, but it focuses exclusively on power consumption. In contrast, the NADV framework can accommodate a variety of different cost types. Depending on system objectives or message priority, applications can use the NADV framework to take different routing strategies. For example, an urgent message can be routed along the path that minimizes the end-to-end latency, and a low-priority message may take a path that minimizes power consumption for longer network lifetime.

For the effective use of NADV, we present techniques for efficient and adaptive link cost estimation. Previous works use additional probe messages for link cost estimation in the bootstrapping phase [13], [14]. However, such control messages consume already scarce network resources. Also, network environments may change over time, and old link estimates may become obsolete. We propose to exploit MAC-level information, so that link cost estimation is adaptive to changing environments, yet incurs minimal control overhead. We also provide multiple techniques thus enabling nodes to choose the best scheme for the current network and system setting. In a resource-rich network, for example, nodes can use a method that uses probe messages. In the case of a dense large-scale network with limited resources, such probe messages may prove to be costly, and nodes can use an alternate scheme that uses no extra control messages.

We have performed simulation experiments to evaluate the effectiveness of NADV and link cost estimation techniques. When compared to the current geographic routing scheme in challenging environments with frequent packet losses, NADV leads to 83% higher packet delivery ratio on average (from 15% to 98%). The number of MAC-level data transmissions and end-to-end delay also decrease significantly (by up to 70%). The simulation results also show that when link costs

change, the use of NADV in geographic routing enables adaptive path migration, where the quality of found paths is close to the optimum by the centralized algorithm.

The rest of this paper is organized as follows. In Section II we define the new link metric. Link cost types and estimation techniques are described in Section III. We describe simulation models in Section IV, and actual simulation results are presented in Section V. Section VI presents related work, and Section VII concludes.

II. NEW LINK METRIC FOR GEOGRAPHIC ROUTING

In this section, we introduce a new link metric for geographic routing and discuss its optimality in an ideal setting. Here, we assume link cost is positive and known a priori. We discuss link cost estimation in Section III.

A. Background

In this paper we differentiate link *cost* and link *metric*. An example of link *cost* is power consumption. Even though two neighbors require the same power consumption, however, in geographic routing we prefer the neighbor closer to the destination. We define link *metric* as “degree of preference” in path selection.

In many geographic routing protocols, the current node S greedily selects the neighbor that is closest to destination T whenever possible [1], [2], [3]. This strategy tries to maximize the *advance* (*ADV*) of next hop n , which is defined by [15]:

$$ADV(n) = D(S) - D(n), \quad (1)$$

where $D(x)$ denotes the distance from node x to T . (Note that *ADV* is an example of link metric.) The implicit goal of this strategy is to minimize the hop count, without taking link costs into account. However, different wireless links can have different link costs. For example, Lundgren et al. [16] identify *gray zones*, where due to high error probability, nodes cannot exchange long data packets in most cases. Therefore, the simple policy using *ADV* may use poor quality links and lead to unnecessarily high communication cost [13].

Clearly, when choosing next hops we want to avoid neighbors with very low quality links. At the same time, we want to gain as large advance as possible for fast and efficient packet delivery. The goal of our work is to balance the trade-off, so that we can select a neighbor with both large advance and good link quality. We can achieve this goal by using the new metric proposed next.

B. Normalized Advance

We now introduce a new metric called *normalized advance (NADV)*. Suppose we can identify the link cost $Cost(n)$ of the link to neighbor n . Then the normalized advance of neighbor n is simply:

$$NADV(n) = \frac{ADV(n)}{Cost(n)}. \quad (2)$$

Intuitively, NADV denotes the amount of advance achieved per unit cost. For example, suppose we know that only $P^{succ}(n)$ fraction of data transmissions to neighbor n are successful. If we use $1/P^{succ}(n)$ as link cost, $NADV(n) = ADV(n) \times P^{succ}(n)$, which means the expected advance per transmission.

We now show the path optimality when using NADV. The goal of routing in this discussion is to minimize the sum of link costs along the found path. We make two assumptions: (1) we can find a node at an arbitrary point, and (2) link cost is an increasing convex function of distance (e.g., power consumption [12], [17]). Let $DIST$ be the distance between the source and the destination, which we assume is relatively large. Then, the optimal policy is to choose nodes on an equidistant basis along the line that connects the source and the destination. The optimality of this simple strategy may not be intuitively obvious, but this is true because the underlying link cost function is convex. Now, it remains to find the optimal interval. Suppose ADV_X is an interval, and $Cost_X$ is the corresponding link cost. Then we want to minimize:

$$\begin{aligned} Total\ Cost &= (Link\ Cost) \times (Hop\ Count) \\ &= Cost_X \times \left\lceil \frac{DIST}{ADV_X} \right\rceil \\ &\approx DIST \times \frac{Cost_X}{ADV_X}. \end{aligned} \quad (3)$$

The last line comes from the assumption of large $DIST$, which makes the rounding error negligible. From Eq. (3) we can find the minimum cost path by iteratively selecting nodes with minimum $\frac{Cost}{ADV}$. Equivalently, the node maximizing $NADV = \frac{ADV}{Cost}$ approaches the optimal point as node density becomes higher.

We propose to use NADV as link metric in geographic routing, such that a node forwards packets to the neighbor with largest NADV. Besides obvious simplicity, NADV has the following desirable properties:

- The path found by using NADV approaches the optimal path in dense networks. The experiment results in Section V show that the use of NADV significantly improves path quality in sparse environments as well.
- It is general and accommodates various types of cost metrics, so that applications can utilize the NADV

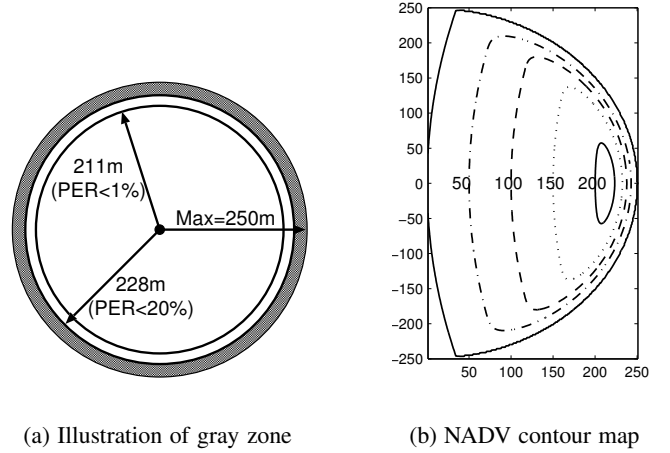


Fig. 2. Illustration of gray zone and corresponding contour map of NADV. (a) Two inner circles represent the border lines for 1% and 20% packet error rates (PERs) for a 1024-byte frame, respectively. (b) The corresponding contour map of NADV when the packet error probability determines link cost. The current node is at (0,0), and the destination is 900 meters away on the X-axis. Values within the plot denote the NADVs of corresponding lines.

framework for different objectives. We further describe this feature in Section III.

- Loop freedom is guaranteed as long as we select a node with positive NADV [15].

Using NADV, we can select neighbors that balance the advance against the link cost. Depending on the link cost values, NADV can select a neighbor with strictly less advance (e.g., node B over A in Figure 1). We further illustrate this feature in Figure 2. Figure 2-(a) shows the degree of packet errors to simulate a gray zone¹. In Figure 2-(b), we present the corresponding contour map of NADV when link cost is a function of packet error probability. We can observe that compared to their ADV values, points within the gray zone have relatively low NADV values. As a result, by using NADV, we can easily avoid neighbors in the gray zone.

Although the concept of NADV is simple, the implementation for practical use involves a number of challenges. Link cost estimation is one of the most critical elements, and in the next section, we describe a set of methods to infer various types of link costs and show how the NADV framework utilizes them.

III. LINK COST TYPES AND ESTIMATION

For effective link cost inference, we propose a new sublayer named *Adaptation Sublayer (ASL)* located on

¹The bit error function used here increases rapidly after a certain distance. A detailed description on the error model is in Section IV-A.

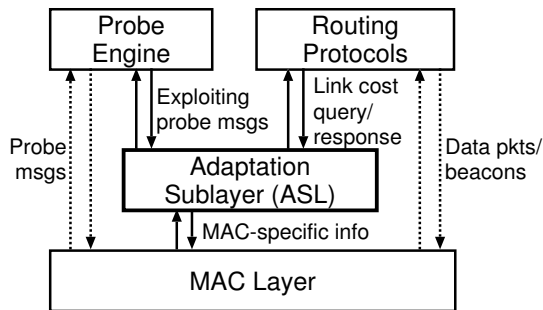


Fig. 3. Adaptation Sublayer (ASL) abstracts the performance of wireless link into numeric values and provides simple system calls for upper layer protocols to retrieve specific performance values. For link cost estimation, ASL exploits MAC layer specific information. If available, ASL can leverage probe messages as well.

top of the MAC layer (See Figure 3). The ASL closely coordinates with the MAC layer for efficient link cost estimation. It also provides simple system calls for upper layer protocols to retrieve the inferred performance values. When additional control messages are available [13], [14], [18], ASL extracts relevant link cost information from them. Otherwise, ASL exploits MAC-specific information to infer the communication performance. For example, the ASL retrieves the current transmission power from the MAC layer and calculates overall power consumption needed for a packet transmission. We note that this sublayer approach and the estimation schemes below can be used in any cost-aware wireless routing.

To illustrate how to use the NADV framework to meet different performance objectives, we discuss three specific types of link cost: packet error rate, link delay, and energy consumption. We also describe how ASL is used to estimate link costs in diverse operating environments. This paper focuses on the independent use of each link cost, and the issue of interdependence among multiple cost criteria is discussed in Section VII.

A. Packet Error Rate (PER)

Most recent attention has been on how to find a high performance path considering wireless link errors [19], [13]. In this scenario, we use the following as link cost: $C_{error} = 1/(1 - PER)$. It denotes the expected transmission count (ETX) in [13]². We use the following link metric, which is the expected advance per transmission:

$$NADV_{error} = \frac{ADV}{C_{error}} = ADV(1 - PER). \quad (4)$$

²The estimation techniques described here can easily incorporate ACK frame loss probability as in [13], but here we have simplified the description for brevity.

	Experiment Number				
	E1	E2	E3	E4	E5
Observed (16-byte)	0.016	0.055	0.008	0.102	0.516
Observed (1024-byte)	0.129	0.488	0.020	0.508	0.910
Estimated (1024-byte)	0.155	0.452	0.080	0.682	0.999

TABLE I

OBSERVED AND ESTIMATED PERs FOR FIVE EXPERIMENTS WITH VARYING DISTANCE.

We next present four PER estimation methods for $NADV_{error}$, each of which requires a different degree of control overhead and message format modification.

Using Probe Messages for PER Estimation

If a node is already using probe messages [13], [14], the ASL can extract the link error probability from them. However, since such probe messages are usually shorter than data packets, a node may experience higher PERs for actual packets [13], and we adjust PER as follows. Assuming independent bit errors, if we use l -bit probe messages, we infer bit error rate p_b using: $PER(l) = 1 - (1 - p_b)^l$, or equivalently $p_b = 1 - (1 - PER(l))^{1/l}$. Then, for a L -bit data frame we can use: $PER(L) = 1 - (1 - PER(l))^{L/l}$. More advanced bit error models also can be employed for more accurate PER estimation [20].

To see whether this simplistic extrapolation will work in practice, we performed the following experiments. A designated sender alternately broadcasts 16-byte and 1024-byte UDP packets every 0.5 seconds for 256 seconds, and a receiver records the received packet size and sequence number. Then, the goal is to estimate the PER of 1024-byte packets using the statistics of 16-byte packets. We use two IBM Thinkpad laptops and two IEEE 802.11b ORiNOCO cards communicating in ad-hoc mode. To obtain different PER values, we have varied the distance between the two laptops. We performed the experiments in an empty parking lot to reduce the effect of multipath fading and other transmissions. In Table I, we report representative results for the observed and estimated PERs of 1024-byte packets³. We observe that the estimated PERs for 1024-byte packets (third row) are reasonably close to the observed values (second row). Although this experiment is not comprehensive, the result suggests this technique is promising for estimating PERs of longer data packets.

Using Signal-to-Noise Ratio for PER Estimation

We can also estimate p_b using Signal-to-Noise Ratio (SNR) and theoretical error models for different modulation schemes [21]. Assuming an AWGN (Additive White

³Calculation includes extra 88 bytes of headers in lower layers.

Gaussian Noise) channel, in the case of BPSK (Binary Phase Shift Keying), the bit error rate is given by:

$$p_b = 0.5 \times \operatorname{erfc}\left(\sqrt{\frac{P_r \times W}{N \times f}}\right), \quad (5)$$

where P_r is the received power, W the channel bandwidth, N the noise power, f the transmission bit rate, and erfc the complementary error function. Most wireless cards typically measure $\text{SNR} = 10 \log \frac{P_r}{N}$ (dB) for each received packet, and using such SNR values and Eq. (5), a node can calculate p_b for its neighbors. Then, due to possibly asymmetric link quality, it should inform its neighbors of respective SNR values. This can be done either via additional control messages or by modifying the beacon message format to include the information. This scheme is useful primarily in free-space environments, but not applicable for indoor environments, where signal path characteristics are more complex ⁴.

Neighborhood Monitoring for PER Estimation

A node can also use passive monitoring to infer link PERs as in [22]. For example, in IEEE 802.11 networks, node A can monitor frames sent by neighbors. In that case, using the MAC sequence number A can count how many frames from neighbor B it has missed, and infer the PER of link $B \rightarrow A$. Again, since the quality of two directional links may differ, A needs to inform B of the PER estimation as in the previous scheme.

Self Monitoring for PER Estimation

The previous methods require either additional control messages or the modification of beacon message format. When these are not possible, we suggest the following technique. Whenever a node transmits a data frame to neighbor n , the MAC-layer informs the ASL whether the transmission was successful or not. Let us define an indicator variable F ; $F = 1$ when a frame exchange failed, and $F = 0$ otherwise. Then, ASL infers the PER of wireless link to neighbor n as follows:

$$\text{PER}_n \leftarrow (1 - \alpha)\text{PER}_n + \alpha F, \quad (6)$$

where α denotes the weight parameter. In the simulation study in Section V, we use $\alpha = 0.1$, and the default PER value is set to 0. Note that $F = 1$ even when an ACK frame failure occurs in IEEE 802.11 networks [13].

⁴We also measured SNR values in the previous experiments, but were not able to obtain meaningful results because the firmware and driver return values that are too coarse-grained (decibel (dB) values with integer precision).

To track the link quality change even when no packets are forwarded to n , we use an *aging* scheme and periodically reduce PERs of unused links. When this reduction makes the estimated PER become lower than the actual one, packets may be forwarded to n , but the estimated PER will increase after transmission failures. The magnitude and frequency of reduction should balance such overhead and prompt adjustment. In the simulation, we multiply PERs of unused links by 0.9 every 30 seconds.

B. Delay

If link delay C_{delay} is used as link cost to reduce the path end-to-end delay, we can use $\text{NADV}_{\text{delay}} = \frac{\text{ADV}}{C_{\text{delay}}}$. We can think of two types of link delay. First, due to the broadcast nature of wireless medium, it is desirable to minimize the *medium time*, the time spent in sending a packet over the link [23]. When the underlying physical medium supports multi-rate transmissions (e.g., the IEEE 802.11 standard), it is a function of the current transmission rate. The ASL can easily retrieve the current value of transmission rate from the MAC layer and calculate the necessary medium time to the neighbor.

The other is *total delay*, which denotes the time from the packet insertion into the interface queue until the notification of successful transmission. It includes the queuing delay, backoff timeout, contention period, and retransmissions due to errors or collisions. Using this value as link cost can potentially enable packets to detour congested areas. The design of a routing scheme with such detouring capability is a part of our future work, and we use the medium time as C_{delay} in this paper.

C. Power Consumption

Many wireless systems have a control mechanism for transmission power adjustment to save battery and reduce interference [18], [21]. We assume that using such a mechanism, nodes know the appropriate transmission power level (p_{tx}) to each neighbor. Then, the ASL can retrieve the p_{tx} value and calculate the actual system power consumption C_{power} considering additional components of power consumption [24]. If C_{power} is used as link cost, a geographic routing protocol can use $\text{NADV}_{\text{power}} = \frac{\text{ADV}}{C_{\text{power}}}$ to find a path that minimizes power consumption to deliver packets to a destination.

So far, we have listed interesting cost types and shown how the NADV framework can incorporate them. The NADV framework still can include other types of link cost as well (for example, *reluctance* metric in [25]). However, in this paper we limit our attention to the cost types discussed above and report simulation results in the following sections.

IV. SIMULATION MODEL

We use *ns-2* simulations to evaluate the system performance when we employ the proposed NADV metric and link cost estimation schemes. In this section, we describe various aspects of simulation in detail. We present the simulation results in Section V.

We place nodes uniformly at random on a 1000m by 1000m square. Unless otherwise stated, 100 static nodes are used in the simulation. We usually use only one source-destination pair to capture the individual performance effects accurately. In this scenario, denoting the lower left corner of the square as (0, 0), the static source is located at (50, 500). The destination is placed at (50+ D , 500), where D is the distance between the source and the destination. The source generates a CBR (Constant Bit Rate) flow, which sends a 1024-byte UDP packet every two seconds from 10 seconds to 1000 seconds of simulation time. The maximum transmission range R is 250 meters.

For geographic routing, we use the simulation code for GPSR⁵. We have slightly modified the next hop selection algorithm to include NADV. The simulation code for GPSR provides an option about whether to exploit transmission failure notification from the MAC layer [2]. If a node exploits the option, then upon receiving a notification, it selects the next best neighbor for retry until the forwarding is successful. This option leads to higher delivery ratio with higher resource consumption. When not using the notification, a node does not attempt to retransmit to other neighbors. We explore both cases in the simulation. The beaconing period in GPSR is set to 1.5 seconds. We use the IEEE 802.11b standard for the underlying MAC layer protocol [26]. We assume the location of the destination is known to the source.

In the following subsections, we describe models of individual simulation components in more detail.

A. Error Model

To simulate a noisy channel, we assume that the physical layer uses the BPSK modulation. Assuming independent bit errors, we simulate packet errors using Eq. (5) as bit error model. In the default *ns-2* propagation model, the signal strength is reduced proportionally to d^2 if the distance d is smaller than a certain threshold. Otherwise, the path loss is proportional to d^4 . In this experiment scenario the transmit signal power is fixed at 20 mW (or 13dBm) supported in Cisco Aironet 350 interface cards [27]. Then the received signal strength for a node 250 meters away is -85dBm. The noise channel

Interval	Time (μ s)	Frame	Length (bytes)
T_{PHY}	192	L_{RTS}	20
SIFS	10	L_{CTS}	14
DIFS	50	L_{ACK}	14

TABLE II

CONSTANTS USED TO CALCULATE MEDIUM TIME IN EQ. (7).

bandwidth in Eq. (5) is set to 2MHz. We report the results in ambient noise environments, where the noise value is identical everywhere. Therefore the quality of a link depends only on the distance between two nodes⁶.

B. Transmission Rate Adaptation and Link Delay

Most IEEE 802.11b wireless cards dynamically adjust the data transmission rate b_{data} using Automatic Rate Fallback (ARF) [28]. In ARF, according to MAC transmission failures or successes, each node adjusts b_{data} to 1, 2, 5.5, or 11 Mbps. For control frames such as RTS and CTS, nodes use another transmission rate b_{basic} , which is fixed at 1 Mbps in the simulation study. We incorporate ARF algorithm to the MAC simulation code.

We assume the underlying MAC uses DCF and performs an RTS/CTS exchange for each packet [26]. Each MAC frame is prefixed by 192 bits of PHY preamble and header, and frames are separated by SIFS or DIFS. Then using the constants in Table II, we calculate the *medium time* for an exchange sequence of a L_{DATA} -byte data frame as follows:

$$C_{delay} = 8 \frac{L_{RTS} + L_{CTS}}{b_{basic}} + 8 \frac{L_{DATA} + L_{ACK}}{b_{data}} + 4T_{PHY} + 3SIFS + DIFS \quad (7)$$

C. Power Consumption Model

As discussed above, the strength of transmitted signal decreases in proportion to d^n , where d is distance, and n is the *path loss exponent* (usually $2 \leq n \leq 6$). Suppose a receiver network interface requires a received signal strength of at least S_{min} for successful reception. To simplify the description we assume that the transmission power should be at least $d^n S_{min}$ for successful reception at a receiver whose distance is d . Modeling after most wireless systems and products [21], [27], we assume in this paper that the transmission power is restricted to one of L levels in the set $P = \{p_1, p_2, \dots, p_L\}$. In this scenario, it is best for a node to use the smallest power level no less than $d^n S_{min}$:

$$p_{tx} = \min\{p_m : d^n S_{min} \leq p_m, 1 \leq m \leq L\}. \quad (8)$$

⁵Available at <http://www-2.cs.cmu.edu/~bkarp/gpsr/gpsr.html>

⁶Note that C_{error} is a convex function of distance in this model.

(dBm)	Noise power ($\times 1.0\text{e-}12$ W)				
	0.8	1.0	1.2	1.4	1.6
BER at 220m	6.0e-8	1.1e-6	7.8e-6	3.2e-5	9.1e-5
BER at 240m	4.4e-6	3.5e-5	1.4e-4	3.9e-4	8.3e-4

TABLE III

BIT ERROR RATE VALUES WITH DIFFERENT LEVELS OF NOISE.

Simplifying the ns -2 propagation model, we fix $n = 4$ in the power consumption experiments. Also, we focus on the relative magnitude of power consumption and use $S_{min} = 1/(R)^n$, where R is the maximum transmission range. Based on the specification of the Cisco Aironet 350 card [27], we use the following set $P = \{0.01, 0.05, 0.2, 0.3, 0.5, 1.0\}$.

We also simplify a widely used power consumption model [17], [12], [24] and assume that each packet forwarding consumes the following amount of energy ⁷:

$$C_{power} = 1 + c p_{tx}, \quad (9)$$

where c is a proportionality constant to the transmission power component. Note that $c = 0$ degenerates C_{power} to the hop count metric. c is a hardware-specific constant, which we assume ASL can retrieve. Actual c values of different interfaces range between 0.17 and 1.30, and we use $c = 1.0$ in the simulation [24].

In some of our simulations, we compare $NADV_{power}$ against SP-Power [12]. Given a power consumption equation, the authors of [12] derive a formula for link metric and prove that the node selection based on the metric is optimal in an ideal setting. If SP-Power uses Eq. (9) as power consumption equation, the current node selects the neighbor that minimizes the following formula:

$$E = (1 + c t_{px}) + \frac{D}{R} (c(n-1))^{\frac{1}{n}} + \frac{D}{R} c (c(n-1))^{\frac{1-n}{n}} \quad (10)$$

where D denotes the distance between the neighbor and the destination.

V. SIMULATION RESULTS

In this section we present the results of simulation experiments. We first discuss the effect of wireless link errors. Then we consider the cases when delay and power consumption are used as link costs in turn. Finally we compare geographic routing using NADV against idealized routing.

⁷Energy consumption of idle network interfaces is an independent issue, which we do not consider in this paper.

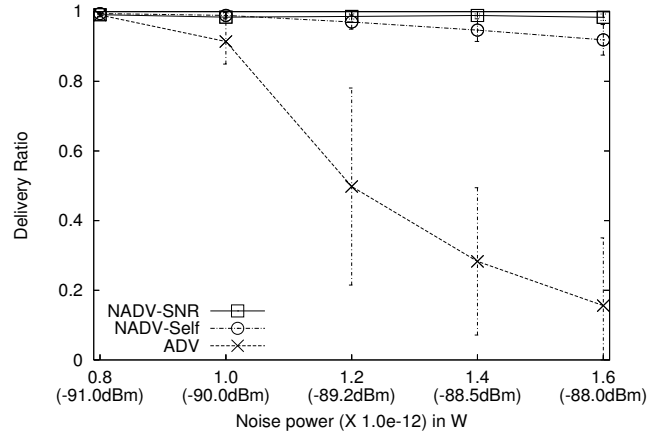


Fig. 4. Packet delivery ratios when noise levels change. Values in parentheses on the X-axis are noise powers in dBm. Each point is an average of ten results, and error bars denote the corresponding standard deviations. The use of ADV may lead to abrupt performance degradation, but with NADV, the delivery ratios are maintained high.

A. Experiments in Lossy Environments

In this subsection we present the results when nodes experience frame losses due to wireless link errors. We fix the data transmission rate at 1 Mbps in this set of experiments ⁸. We use the error model described in Section IV-A. In Table III we illustrate the used noise values and corresponding bit error rates (BERs) ⁹. There is only one source-destination pair, and the distance between them is 900 meters.

We consider two PER estimation methods in the simulation: the method using SNR values, and the self-monitoring scheme. When combined with NADV, they are called *NADV-SNR* and *NADV-Self*, respectively. In the case of NADV-SNR, we modify the periodic beacon format to include reverse link information, and the message length slightly increases accordingly in this scenario. Note that neither scheme uses extra control messages. Also, storage overhead for the link cost estimation is negligible since each node in GPSR maintains neighbor information. We first present results with static nodes. The experiment results with mobile nodes are presented later in this subsection.

Experiments with Static Nodes

We first compare packet delivery ratios when we do not employ MAC-level failure notification. In Figure 4,

⁸In this experiment, ARF would adjust the transmission rate to 1 Mbps for most links. However, ARF occasionally attempts a higher data rate [28], which would lead to more packet errors.

⁹Noise values from more than 20000 measurements in our building range from -91dBm to -73dBm, with the median at -89dBm. The noise value used in Figure 2 is -89.2dBm.

we plot the delivery ratios achieved by ADV and NADV, respectively, when we vary the noise power values. We use an average of ten runs, each with different node placement, and the error bars in the figure are the standard deviations. When wireless link errors are rare (noise=-91dBm), the use of ADV in geographic routing leads to relatively good performance. However, as the channel condition degrades, the performance gap between NADV and ADV grows larger. For example, when ADV is used with noise power=-89.2dBm, less than 50% of packets can reach the destination on average. However, when NADV is used, the delivery ratio is maintained high (> 97% on average for both NADV schemes). Since NADV-SNR explicitly utilizes the link characteristic value, it leads to paths of higher-quality links, and the delivery ratios are slightly higher than those of NADV-Self.

The performance gap between NADV and ADV is explained as follows. The neighbor with maximum ADV is relatively far from the current node, and the corresponding link may experience a higher PER. However, when using ADV, the current node blindly uses the neighbor with maximum ADV and suffers from the poor quality link. In contrast, when using NADV, the link estimation schemes find out that the link quality is poor and accordingly reduce the NADV value for the neighbor. Then, the node chooses a different neighbor with a larger NADV value, and the overall network performance improves. We can also observe the high *variance* of delivery ratios when ADV is used. For example, when noise power=-89.2dBm, the delivery ratios of ten runs range between 21.0% and 97.2% for ADV. The reason is that, in some fortunate cases, all forwarding nodes selected by ADV can possibly be outside of gray zones, and the forwarding does not experience high PERs. In contrast, NADV selects the neighbor with the best trade-off and consequently leads to stable performance (> 92% in the worse case).

In Figure 5 we plot the average path lengths for each scheme. To investigate the adaptiveness, we start with a high noise value, change to a low noise value after 300 seconds, and change again to a medium noise value after 700 seconds. The length of the path chosen by ADV is always shortest, but the packet delivery performance is much worse than that of NADV. In the high noise scenario, although ADV does not adapt to the environment, beacons from far neighbors are frequently lost, and the path length increases. In the case of NADV, the PER estimation schemes dynamically assign appropriate link costs. As a result, NADV uses different neighbors according to the current environment, and the path length change is more noticeable. NADV-SNR exhibits more

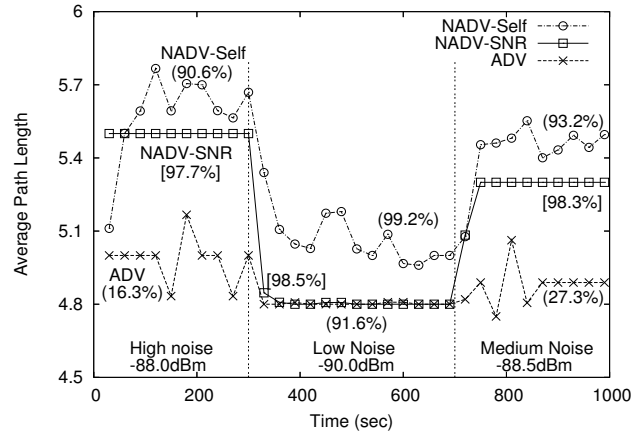


Fig. 5. The average path lengths of NADV and ADV. The noise value changes from high (-88.0dBm) to low (-90.0dBm), and finally to medium (-88.5dBm). Numbers next to the lines are corresponding delivery ratios in each phase. PER estimation schemes enable NADV to choose appropriate neighbors and maintain high delivery ratios.

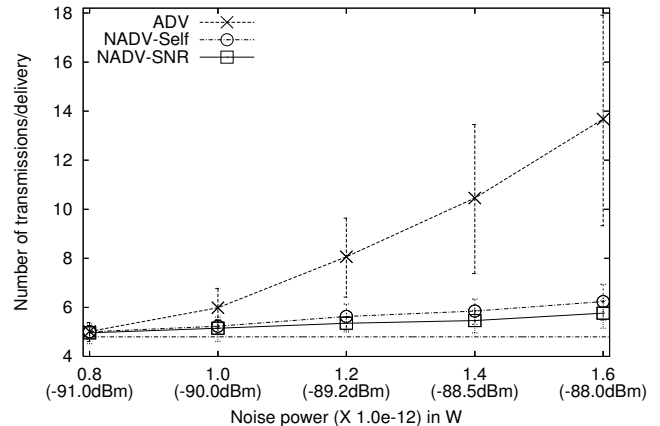


Fig. 6. The number of MAC-level data transmissions per delivered packet with different degrees of packet errors. C_{error} is used as link cost. Due to multiple retransmissions, ADV causes a higher number of total transmissions than NADV. As marked in the bottom, the average path length found by ADV when noise=-91dBm is 4.8 hops.

accurate estimation and faster convergence. NADV-Self occasionally employs slightly longer paths than NADV-SNR, but it is also able to adapt to environment change.

In Figure 6, we report the number of MAC-level data transmissions (including retransmissions) per delivered packet for both ADV and NADV. GPSR employs MAC-level failure notification in this set of experiments, and all results are based on 100% packet delivery. We can see that NADV intelligently avoids nodes with high PER, and the number of data transmissions is accordingly much smaller. Each transmission requires network bandwidth as well as node resources (e.g. battery power), and NADV uses system resources more efficiently. In

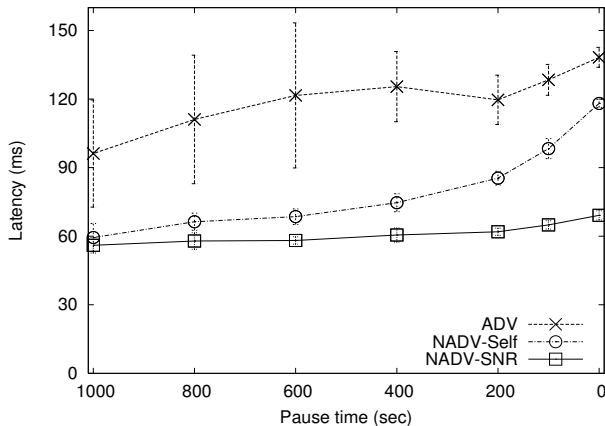


Fig. 7. Average end-to-end latency when nodes are mobile. C_{error} is used as link cost. We changed the pause time for different degrees of mobility. NADV and the proposed cost estimation schemes are effective even with node mobility.

contrast, the use of ADV leads to the waste of system resources due to repeated retransmissions (up to 130% more transmissions than NADV). Also, the ADV performance degrades rapidly as bit error rates become higher. In contrast, by leveraging PER estimation, NADV enables graceful performance degradation.

Repeated retransmissions also affect the packet delay. Although not displayed here, the end-to-end latencies of ADV also show an increasing trend similar to Figure 6. Specifically, as the noise value changes from -91dBm to -88dBm, the average latency of ADV increases from 52.3ms to 201.6ms. NADV again exhibits graceful performance degradation (increase from 52.0ms to 67.7ms for NADV-Self, and from 51.6ms to 60.2ms for NADV-SNR).

Experiments with Mobile Nodes

In the previous results, we have shown that our proposed techniques for PER estimation perform well in static networks. We now investigate how well they adapt to network topology changes. In this scenario, the source and destination pairs do not move, but the remaining 98 nodes move according to the random waypoint model. The speed is randomly chosen between 1 and 10 m/s, and we vary the pause time for different degrees of node mobility. We use the MAC-level failure notification and fix the ambient noise power at -89.2dBm, which is close to the median of noise measurements in our building.

In Figure 7, we present the end-to-end latency results with varying mobility. We observe that average latencies increase as node mobility becomes higher. This is because frequent link failures cause more retransmissions. Compared to ADV, both NADV schemes achieve lower

Distance	500m	600m	700m	800m	900m
ADV	22.9	26.5	31.7	36.2	42.9
NADV _{delay}	14.5	17.3	20.0	22.7	26.2

TABLE IV
AVERAGE END-TO-END LATENCY (IN MS) WITH DIFFERENT SOURCE-DESTINATION DISTANCES.

average latency. With NADV-SNR, PER estimation is more accurate, and the increase in end-to-end latency is minimal even with the highest mobility (50% better compared to ADV). When NADV-Self is used in high mobility scenarios, most neighboring nodes move out of range before the estimated values can converge. As a result, the performance gain is smaller than in low mobility cases. Still, its average latency is 15% shorter than that of ADV when nodes move constantly.

To summarize, NADV provides an efficient and adaptive geographic routing strategy, and the proposed link estimation schemes are effective even with node mobility. As the network environment becomes harsher, the performance of NADV degrades gracefully. In the next subsection, we discuss the results when link delay is used as link cost.

B. Using Delay as Link Cost

In this subsection, we use link delay as link cost and assume $NADV \equiv NADV_{delay}$ in this scenario. We use the same error model in the previous subsection, and ARF is used for rate adjustment. In this experiment, we use a low noise value of -91.0dBm in this set of simulations. Note that this scenario is in favor of ADV because with high noise, ADV suffers from increased end-to-end latency as previously discussed. The MAC-level failure notification is exploited, and the delivery ratios are 100% in all cases. Each value in this experiment is an average of ten runs.

In Table IV, we report the average end-to-end latency of each scheme when we vary the distance between the source and the destination. As the distance increases, packets go through more relay nodes, and the latency increases accordingly. NADV significantly decreases the end-to-end latency (by up to 35%). It is because when we use ADV, we are likely to choose far neighbors to minimize the distance to the destination. However, the transmission rates to such nodes are usually 1 or 2 Mbps, which causes the transmission to take longer. With the use of NADV, the current node may choose a neighbor that is not the closest to the destination, but the corresponding link is good enough for a higher transmission rate such as 11 Mbps. This strategy eventually leads to shorter transmission time.

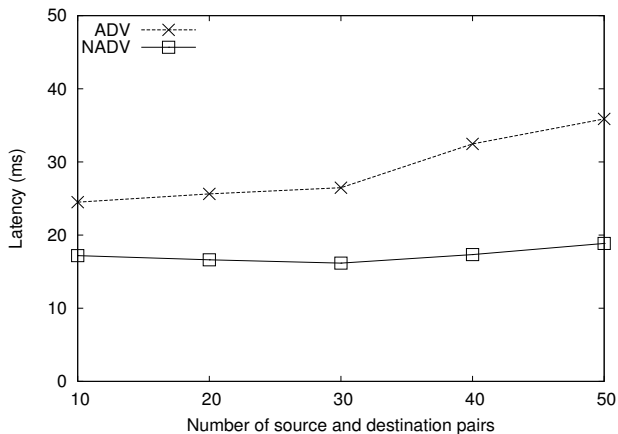


Fig. 8. Average end-to-end delay with multiple flows. ARF and C_{delay} are used. When NADV is used, the network can support more flows without significant increase in the latency.

When using $NADV_{delay}$, the current node usually selects neighbors close to itself, which leads to more relay nodes (e.g., 55% increase when the distance is 900m). Since this increase is based only on the local decision to minimize medium time, it may degrade the overall performance, especially when multiple traffic flows exist. To investigate this potential problem, we perform experiments using different numbers of source-destination pairs selected uniformly at random.

In Figure 8, we plot the average end-to-end latency when we change the number of flows in the network. We can observe that with more flows in the network, ADV increases the average latency noticeably. This is because ADV holds the medium longer than necessary, leading to a higher level of network contention. In contrast, NADV maintains the aggregate medium time low enough, such that the network can support more flows without significant increase in the latency. Consequently, compared to ADV, NADV improves the latency performance even more with higher network traffic. Specifically, in the case of 10 flows, NADV decreases the average latency by 30%, but with 50 flows the improvement is 48%. In the case of 50 flows, only 2 flows experience slight increase (< 2 ms) in the end-to-end delay. This experiment result shows that the use of $NADV_{delay}$ does not negatively affect the performance of other traffic.

C. Using Power Consumption as Link Cost

We compare $NADV$ ($\equiv NADV_{power}$) against the metric proposed in SP -Power scheme [12]. When the power consumption equation is $C_{power} = 1 + ct_{px}$, $NADV$ needs to know the current transmission power p_{tx} , which we assume is available through a control mechanism. We also assume SP -Power knows the exact value of

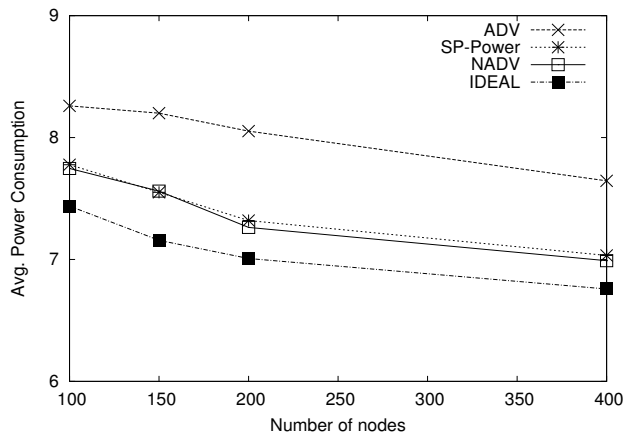


Fig. 9. Average power consumption with different schemes. In dense networks, as more neighbors are available, power consumption decreases. The power consumption values by NADV and SP -Power are similar, which are close to the optimal value.

path loss exponent. In practice, however, the path loss exponent estimation is not trivial, and depending on the measurement parameters, the estimated values can vary significantly [21], [29]. We assume that both schemes know the proportionality value c , which is a hardware-specific constant. In the following set of simulations, the distance between source and destination is 900m, and there are no packet errors. We vary the node density and use average values of 20 runs for each case. We also include paths found by the optimal centralized algorithm.

In Figure 9, we plot the average power consumption of each scheme with different node density. The amount of power consumption in each scheme decreases as node density increases. This is because with higher node density, more neighbors become available, and all schemes likely choose better next hops. We also observe that compared to ADV, both NADV and SP -Power find paths that reduce overall power consumption¹⁰. The performances of NADV and SP -Power are almost identical; NADV performs slightly better. (NADV and SP -Power find the same path in 15 cases out of 20 in the 400-node scenarios.) Even though we do not report detailed results in this paper, $NADV_{power}$ and SP -Power also show very similar performance in other environments (e.g., distance, continuous power adjustment, different path loss exponents, and proportionality constants c).

When the goal of geographic routing is to minimize the path power consumption, we argue that $NADV_{power}$ is the metric of choice. $NADV_{power}$ and SP -Power are

¹⁰In Figure 9, the performance difference between the optimal case and ADV is not large. It is because the constant term in Eq. (9) constitutes a significant power consumption regardless of the transmission power, as is the case with most existing products [24].

	ADV	Non-adaptive (AODV)	NADV one-hop	NADV two-hop	IDEAL
Initial	14.43	11.14	11.28	10.82	10.32
Changed	18.51	14.30	13.50	12.52	11.62

TABLE V

THE AVERAGE COSTS OF PATHS FOUND BY RESPECTIVE ROUTING SCHEMES WHEN LINK COSTS CHANGE.

based on a similar rationale for next hop selection and exhibit almost identical performance. However, as mentioned above, SP-Power needs to estimate the path loss exponent, which can be difficult in practice. In contrast, $NADV_{power}$ only requires t_{px} , which nodes can easily determine with the support of existing control mechanisms [18], [21].

D. Experiments with Generic Cost

In this section, we consider a generic cost metric to see whether the use of NADV can be extended for other types of link cost. We use the following link cost:

$$C_{generic} = 1.0 + r \left(\frac{d}{R} \right)^2, \quad 1 \leq r \leq 5, \quad (11)$$

where r is a uniformly distributed random number, d is the distance between two nodes, and R is the maximum transmission range. According to recent measurements [30], the quality of two links with identical distance can differ widely, and the above metric attempts to capture that effect. In this subsection, we assume the availability of accurate and up-to-date link costs.

We use the following experiment scenario. The source and the destination are 900 meters apart, and the source starts to send data packets after 10 seconds. At 30 seconds, we assume that the environment of some part of the network changes (e.g. due to new obstacles, increased interference), and we randomly select 50% of links and increase their link costs by 50%. For NADV, we additionally consider a geographic routing scheme that uses two-hop neighborhood information [10]. To compare NADV against AODV [31], we modify the AODV simulation code to find paths that minimize the sum of link costs along the paths, not hop count.

In Table V, we report average path quality of each scheme before and after the link cost change. Each value in the table is an average of ten experiments. We can see that using NADV, geographic routing (both one-hop and two-hop) can find paths comparable to the optimal paths. Not surprisingly, utilizing two-hop neighborhood information leads to higher-quality paths than the one-hop case. The performance of initial paths by AODV lies between those by one-hop NADV and two-hop NADV.

However, even after some link cost values increase after 30 seconds, AODV keeps using the initial path, and the path performance degrades accordingly. In contrast, the use of NADV enables localized geographic routing to detect the change and determine better next hops, which results in better paths.

In summary, geographic routing with NADV can find paths whose costs are comparable to the optimum. It is also able to adapt to network environment changes, due to the localized next hop decision.

VI. RELATED WORK

Many ideas and techniques have been proposed for energy-efficient routing in multihop wireless networks. Rodoplu et al. [17] present a localized algorithm that preserves network connectivity and achieves the globally minimum-energy topology. In PARO [32], a node becomes a relay node if it finds that the relaying leads to lower energy consumption. Given traffic flows and node energy levels, Chang et al. [33] find a set of routes that maximize the system lifetime. More recently, wireless link errors has drawn much attention in multihop wireless networks [16], [30]. Banerjee et al. [19] propose the use of a link metric based on link error probability. De Couto et al. use a similar metric called *ETX (Expected Transmission Count)* in real testbed experiments, and the result shows that paths with smaller ETX perform better than shortest paths [13].

Takagi et al. use a metric called *progress*, which is the length of projection along the straight line between the source and the destination [34]. *Compass routing* [35] uses the angle value as link metric in geographic routing. Stojmenovic et al. [12] propose a routing metric for power-efficient routing, as discussed in Section V. Instead of a straight line, Niculescu et al. [4] propose a forwarding strategy based on a pre-defined curve. To route packets around voids, *Face Routing* [1] uses the right-hand rule in Gabriel graph. GPSR employs a similar scheme called *perimeter mode* [2]. Terminode routing uses *Anchored Geodesic Packet Forwarding (AGPF)* similar to loose source routing [9]. Kuhn et al. present GOAFR+, which is efficient on average cases and worst-case optimal [3]. Although their work considers link cost, they still choose the neighbor closest to the destination in greedy mode. Also related is geocasting, which is similar to multicast, but delivers data packets to nodes located inside a certain region [8]. Geographic routing may exploit location service systems [7] and location computation systems [6]. More information about position-based routing can be found in [5].

VII. CONCLUSIONS

We have introduced NADV as link metric for geographic routing in multihop wireless networks. Geographic routing with NADV provides an adaptive routing strategy, which is general and can be used for various link cost types. We have presented techniques for link cost estimation. In the simulation experiments, the combination of NADV and cost estimation techniques outperforms the current geographic routing scheme. NADV also finds paths whose cost is close to the optimum.

In this paper, we have investigated each link cost type independently. However, if we consider multiple interdependent costs simultaneously, choosing the next hop based on one cost type may not be always the best choice for other costs. Our future work is to design a link cost model that balances multiple cost criteria, which would allow the NADV framework to leverage the combined link cost to find a low cost path. We also want to implement the NADV framework on real testbeds and evaluate the performance in practice when it is combined with different link cost types and estimation schemes.

REFERENCES

- [1] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, and Jorge Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proceedings of the 3rd International Workshop on Discrete algorithms and methods for mobile computing and communications*. 1999, ACM Press.
- [2] Brad Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th ACM/IEEE MobiCom*. 2000, pp. 243–254, ACM Press.
- [3] Fabian Kuhn, Roger Wattenhofer, Yan Zhang, and Aaron Zollinger, "Geometric ad-hoc routing: of theory and practice," in *Proceedings of the 22nd annual symposium on Principles of distributed computing*. 2003, pp. 63–72, ACM Press.
- [4] Dragos Niculescu and Badri Nath, "Trajectory based forwarding and its applications," in *Proceedings of the 9th ACM/IEEE MobiCom*. 2003, pp. 260–272, ACM Press.
- [5] Ivan Stojmenovic, "Position-based routing in ad hoc networks," *IEEE Communications Magazine*, pp. 128–134, 2002.
- [6] Jeffrey Hightower and Gaetano Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [7] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris, "A scalable location service for geographic ad hoc routing," in *Proc. of ACM MobiCom*, 2000.
- [8] Young-Bae Ko and Nitin H. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*. 1999, IEEE Computer Society.
- [9] L. Blazevic, S. Giordano, and J. Y. Le Boudec, "Self organized terminode routing," *Journal of Cluster Computing*, vol. 5, no. 2, April 2002.
- [10] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023–1032, 2001.
- [11] D. Johnson and D. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, Kluwer Academic Publishers, 2001.
- [12] Ivan Stojmenovic and Xu Lin, "Power-aware localized routing in wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 11, pp. 1122–1133, 2001.
- [13] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th ACM/IEEE MobiCom*, 2003.
- [14] Abtin Keshavarzin, Elif Uysal-Biyikoglu, Falk Herrmann, and Arati Manjeshwar, "Energy-efficient link assessment in wireless sensor networks," in *Proc. of IEEE Infocom*, March 2004.
- [15] Tommaso Melodia, Dario Pompili, and Ian F. Akyildiz, "Optimal local topology knowledge for energy efficient geographical routing in sensor networks," in *Proc. of Infocom*, March 2004.
- [16] Henrik Lundgren, Erik Nordstr, and Christian Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, 2002, pp. 49–55.
- [17] Volkan Rodoplu and Teresa H. Meng, "Minimum energy mobile wireless networks," *IEEE JSAC*, vol. 17, no. 8, pp. 1333–1344, Aug. 1999.
- [18] IEEE 802.11h Standard, "Part 11. Amendment 5: Spectrum and transmit power management extensions in the 5GHz band in Europe," 2003.
- [19] Suman Banerjee and Archan Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," in *Proceedings of the 3rd ACM MobiHoc*, 2002, pp. 146–156.
- [20] Andreas Kopke, Andreas Willig, and Holger Karl, "Chaotic maps as parsimonious bit error models of wireless channels," in *Proceedings of Infocom*, Apr. 2003.
- [21] Theodore Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*, Prentice Hall, 2001.
- [22] Seungjoon Lee, Suman Banerjee, and Bobby Bhattacharjee, "The case for multi-hop wireless local area network," in *Proceedings of Infocom*, Mar. 2004.
- [23] Baruch Awerbuch, David Holmer, and Herbert Rubens, "High throughput route selection in multi-rate ad hoc wireless networks," in *First Working Conference on Wireless On-demand Network Systems (WONS)*, 2004.
- [24] Rex Min and Anantha Chandrakasan, "Top five myths about the energy consumption of wireless communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 1, pp. 65–67, 2003.
- [25] Suresh Singh, Mike Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proceedings of the 4th ACM/IEEE MobiCom*. 1998, pp. 181–190, ACM Press.
- [26] IEEE 802.11 Standard, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1999.
- [27] "Cisco aironet 350 series client adapters data sheet," June 2003, Cisco Systems Inc. Available at <http://www.cisco.com/>.
- [28] A. Kamerman and L. Monteban, "WaveLAN-II: A high performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, 1997.
- [29] Scott Y. Seidel, Theodore S. Rappaport, Sanjiv Jain, Micheal L. Lord, and Rajendra Singh, "Path loss, scattering, and multipath delay statistics in four European cities for digital cellular and microcellular radiotelephone," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 4, pp. 721–730, November 1991.
- [30] Deepak Ganesan, Bhaskar Krishnamachari, Alec Woo, David Culler, Deborah Estrin, and Stephen Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks," Tech. Rep., UCLA/CSD-TR 02-0013, 2002.
- [31] C.E. Perkins and E.M. Belding-Royer, "Ad hoc on-demand distance vector (AODV) routing," in *IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999.
- [32] J. Gomez, A. Campbell, M. Naghshineh, and C. Bisdikian, "PARO: Supporting transmission power control for routing in

- wireless ad hoc networks,” *ACM/Baltzer Journal on Mobile Networks*, 2002.
- [33] Jae-Hwan Chang and L. Tassiulas, “Energy conserving routing in wireless ad-hoc networks,” in *Proc. of IEEE Infocom*, 2000.
 - [34] H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals,” *IEEE Transactions on Communications*, vol. COM-32, 3, pp. 246–257, 1984.
 - [35] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia, “Compass routing on geometric networks,” in *Proceedings of 11th Canadian Conference on Computational Geometry*, 1999.