Efficient Geographic Routing in Multihop Wireless Networks*

Seungjoon Lee Bobby Bhattacharjee
slee@cs.umd.edu bobby@cs.umd.edu
Department of Computer Science
University of Maryland
College Park, MD 20742 USA

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

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 enables 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.

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 81% higher delivery ratio. When compared to centralized routing under certain settings, geographic routing using NADV finds paths whose cost is close to the optimum.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

General Terms

Design, Performance

Keywords

Wireless multihop networks, geographic routing, routing metric, link cost estimation

1. 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 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 among S's neighbors, the link between S and A is experiencing high packet error probability. B is slightly farther from T than A, but provides

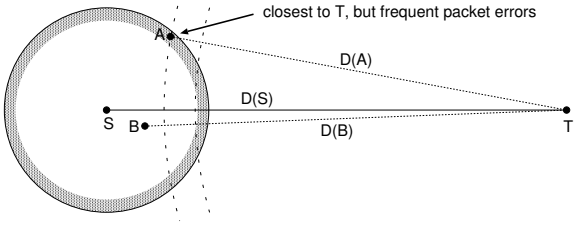


Figure 1: An example scenario for geographic routing. While among S's neighbors, node A is closest to T, the link between S and A is experiencing a high packet error rate. Consequently, higher performance can be achieved if S forwards packets to B.

*This work was supported in part by NSF CAREER Award ANI 0092806.

MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.

a higher quality link from S. 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].

Due to the local rule for next hop decision, the use of NADV in geographic routing provides a unique opportunity for adaptive routing—a feature not offered by most existing on-demand routing protocols. For example, suppose that a source uses an on-demand routing protocol (e.g., AODV [11]) to find a minimum cost path. In dynamic ad hoc networks, it is possible that the link costs change while the path is still in use (e.g., due to mobility or environment changes). If the source detects the change and wants to find a better path, typical on-demand routing protocols require the flooding of a new route discovery message. 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 the best next hop in geographic routing.

We present NADV in the context of a general framework for efficient geographic routing. Although a few recent geographic routing schemes consider link costs in the next hop selection [12, 13, 14], they are limited to one specific objective. For example, the *SP-Power* scheme in [12] focuses exclusively on the minimization of transmission 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 to increase the overall network lifetime.

For the effective use of NADV, we present techniques for efficient and adaptive link cost estimation. Some of previous work uses additional probe messages for link cost estimation in the bootstrapping phase [15, 16]. However, such control messages consume already scarce network resources. Also, network environments may change over time (e.g., due to mobility), 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 81% higher packet delivery ratio on average (from 16% to 97%). The number of MAC-level data transmissions and end-to-end delay also decrease significantly (by up to 60%). 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 found by the centralized algorithm.

The rest of this paper is organized as follows. In Section 2 we define the new link metric. Link cost types and estimation techniques are described in Section 3. We describe simulation models in Section 4 and present actual simulation results in Section 5. Section 6 presents related work, and Section 7 concludes.

2. 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 3.

2.1 Background

In this paper we differentiate link *cost* and link *metric*. An example of link *cost* is the power consumption required for a packet transmission over the link. We define link *metric* as "degree of preference" in path selection. For example, even though two neighbors require the same power consumption, in geographic routing we prefer the neighbor closer to the destination. The goal of this section is to propose a new link metric for geographic routing that can be generalized to various cost types (e.g., power consumption, link delay).

In many geographic routing protocols, the current node S greedily selects the neighbor that is closest to destination T whenever possible [1, 2, 3]. The implicit goal of this strategy is to minimize the hop count between source and destination. Let us consider the amount of decrease in distance by a neighbor n, which we call the *advance (ADV)* of n [17]:

$$ADV(n) = D(S) - D(n), \tag{1}$$

where $D(x)$ denotes the distance from node x to T. Then, the above strategy tries to maximize the ADV of next hop, and ADV is the link metric in this case. However, this link metric ADV does not take link cost into account, while different wireless links can have different link costs. For example, Lundgren et al. [18] 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 [15].

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.

2.2 Normalized Advance

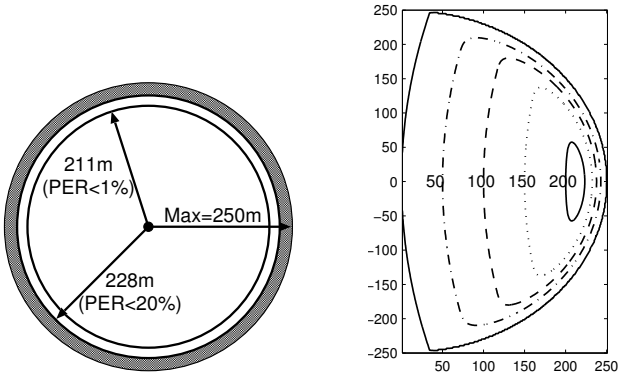
We now introduce a new metric called *normalized advance (NADV)*. Suppose we can identify the link cost $Cost(n)$ of the link to neighbor n. Then the normalized advance of neighbor n is simply:

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

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 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:

- As shown in Section 2.3, the path found by using NADV approaches the optimal path under certain conditions. The



(a) Illustration of gray zone

(b) NADV contour map

Figure 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.

experiment results in Section 5 show that the use of NADV significantly improves path quality in realistic environments as well.

- It is general and accommodates various types of cost metrics, so that applications can utilize the NADV framework for different objectives. We further describe this feature in Section 3.
- Loop freedom is guaranteed as long as we select a node with positive NADV [17].

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 zone.[1] 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. We next provide the theoretical rationale behind using NADV in geographic routing.

2.3 Optimality of NADV in an Idealized Environment

We now show that in an idealized environment, paths found by using NADV are optimal. 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 *unknown* increasing convex function of distance (e.g., transmission power consumption [12, 19]). Let *DIST* be the distance between the source and the destination, which we assume is relatively large. Since the cost function is increasing, and we can find a node at an arbitrary point, an optimal path will use only nodes along the straight line between the source and the destination. Also, since link cost is a convex function of distance, the sum of link costs is minimized when all links have the same distance. As a result, the optimal policy is to choose nodes on an equidistant basis along the line that connects the source and the destination.

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:

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

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 $\frac{Cost}{ADV}$, or equivalently *maximum NADV* $= \frac{ADV}{Cost}$.

In practical wireless networks, the above assumptions are unlikely to be true. In low-density networks, nodes may not be able to use the greedy forwarding rule, and the recovery procedure will likely result in performance degradation [3]. Also, although many existing schemes are based on the simplified model, and there usually exists strong correlation [18, 20], the link cost is not a strict function of distance in practice. In Section 5, we use simulations to show that NADV significantly improves performance in realistic environments as well.

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.

[1]The bit error function used here increases rapidly after a certain distance. A detailed description on the error model is in Section 4.1.

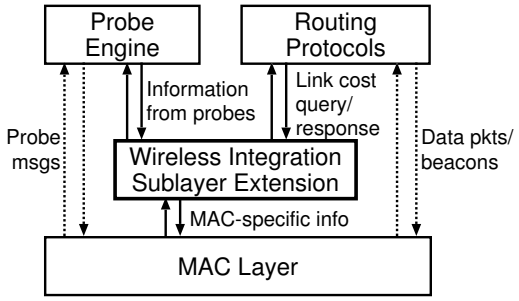


Figure 3: Wireless Integration Sublayer Extension (WISE) abstracts the performance of wireless link into numeric values and provides simple primitives for upper layer protocols to retrieve specific performance values. For link cost estimation, WISE exploits MAC layer specific information. If available, WISE can leverage probe messages as well.

3. LINK COST TYPES AND ESTIMATION

For effective link cost inference, we propose a new sublayer named *Wireless Integration Sublayer Extension* (WISE) located on top of the MAC layer (See Figure 3). The WISE closely coordi-

nates with the MAC layer for efficient link cost estimation. It also provides simple primitives for upper layer protocols to retrieve the inferred performance values. When additional control messages are available [15, 16, 21], WISE extracts relevant link cost information from them. Otherwise, WISE exploits MAC-specific information to infer the communication performance. For example, the WISE 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 WISE 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 7.

3.1 Packet Error Rate (PER)

Most recent attention has been on how to find a high performance path considering wireless link errors [22, 15]. In this scenario, we use the following as link cost: $C_{error} = 1/(1 - PER)$. It denotes the expected transmission count (ETX) proposed in [15].[2] We use the following link metric, which is the expected advance per transmission:

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

An equivalent link metric is independently developed in [13, 14], and we discuss them in Section 6.

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 [15, 16], the WISE can extract the link error probability from them. Since most geographic routing schemes use periodic message exchange between neighbors, we can also use the reception ratio to infer the link error probability. However, such control messages are usually shorter than data packets, and as a result, a node may experience higher PERs for actual packets [15]. To obtain more accurate link cost estimation, we need to adjust PER depending on the data packet length. Assuming independent bit errors, we can adjust PER as follows: If we use l-bit probe messages, from the observed $PER(l)$, 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:

$$PER(L) = 1 - (1 - PER(l))^{L/l}. \quad (5)$$

More advanced bit error models also can be employed for more accurate PER estimation [23].

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 used broadcast transmissions, and to reduce the effect of other transmissions, we performed the experiments in an empty parking lot. In Table 1, we report representative results for the observed and estimated PERs of 1024-byte packets.[3] We observe that the estimated PERs for 1024-byte packets (bottom row) are reasonably close to the observed values (middle row).

	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 1: Observed and estimated PERs for five experiments with varying distance.

We note that compared to our experiment results, the previous results in [15] show different trends between packet length and error rate. In contrast to our experiments, their experiments were done in the presence of obstacles and other potential transmissions, and such difference in experimental environments is likely to lead to different patterns in bit errors (e.g., bursty vs. independent bit errors). We believe that more work should be done to identify the relationship between packet error rate and packet size in various environments. Although this experiment is not comprehensive, the result suggests that 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 [24]. Assuming an AWGN (Additive White Gaussian Noise) channel, in the case of BPSK (Binary Phase Shift Keying), the bit error rate is given by:

$$p_b = 0.5 \times erfc(\sqrt{\frac{P_r \times W}{N \times f}}), \quad (6)$$

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 SNR = $10 \log \frac{P_r}{N}$ (dB) for each received packet, and using such SNR values and Eq. 6, a node can calculate p_b for its neighbors. Using an appropriate bit error models [23], we can infer the packet error rate from p_b. 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.

Eq. 6 is useful primarily in free-space environments, but not applicable for indoor environments, where signal path characteristics are more complex. The measurement results using a rooftop mesh network show that it is hard to predict link quality using SNR [25]. However, in a different measurement study using a sensor network, Zuniga et al. [20] report that empirical results closely match their analytical models. We plan to identify the relationship between SNR and link error rate in various environments.[4]

[2]The estimation techniques described here can easily incorporate ACK frame loss probability as in [15], but here we have simplified the description for brevity.

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

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

Neighborhood Monitoring for PER Estimation

A node can also use passive monitoring to infer link PERs as in [26]. 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 from B to 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 WISE 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, WISE infers the PER of wireless link to neighbor n as follows:

$$PER_n \leftarrow (1 - \alpha)PER_n + \alpha F, \tag{7}$$

where α denotes the weight parameter. In the simulation study in Section 5, 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 [15].

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.

3.2 Delay

If link delay C_{delay} is used as link cost to reduce the path end-to-end delay, we can use $NADV_{delay} = \frac{ADV}{C_{delay}}$. 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 [27]. 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 WISE 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 queueing 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.

3.3 Power Consumption

Many wireless systems have a control mechanism for transmission power adjustment to save battery and reduce interference [21, 24]. We assume that using such a mechanism, nodes know the appropriate transmission power level (p_{tx}) to each neighbor. Then, the WISE can retrieve the p_{tx} value and calculate the actual system power consumption C_{power} considering additional components of power consumption [28]. If C_{power} is used as link cost, a geographic routing protocol can use $NADV_{power} = \frac{ADV}{C_{power}}$ to find a path that minimizes power consumption to deliver packets to a destination.

	Noise power (×1.0e-12 W)				
	0.8	1.0	1.2	1.4	1.6
(dBm)	(-91.0)	(-90.0)	(-89.2)	(-88.5)	(-88.0)
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 2: Bit error rate values with different levels of noise.

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 [29]). However, in this paper we limit our attention to the cost types discussed above and report simulation results in the following sections.

4. 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 5.

We place nodes uniformly at random on a 1000m by 1000m square. Unless otherwise stated, 100 static nodes are used in the simulation.[5] 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. We usually use D=900. 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 GPSR.[6] 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 [30]. 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.

4.1 Error Model

To simulate a lossy channel, we use two different error models. First, assuming the use of BPSK modulation in the physical layer, we simulate packet errors using Eq. 6 as bit error model. (We assume independent bit errors for simplicity.) In the default *ns-2* propagation model, the signal strength is reduced proportionally to

[5]We also experimented using sparser networks with 50 nodes. However, in scenarios with high packet error rates, networks frequently became disconnected (e.g., due to repeated beacon message losses).

[6]Available at http://www-2.cs.cmu.edu/~bkarp/gpsr/gpsr.html

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

Table 3: Constants used to calculate medium time in Eq. 8.

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 [31]. Then the received signal strength for a node 250 meters away is -85dBm. The noise channel bandwidth in Eq. 6 is set to 2MHz. In this model, we use ambient noise environments, where the noise value is identical everywhere. Therefore the quality of a link depends only on the distance between two nodes, and C_{error} is a convex function of distance. In Table 2 we tabulate the used noise values and corresponding bit error rates (BERs).[7]

To examine the performance of NADV in the presence of randomness in packet errors [25], we also perform simulations using a random packet error model. In this model, for each wireless link, we assign a packet error rate, which is distributed uniformly at random between 0 and a maximum value (*max-PER*). We vary the maximum packet error probability for different degrees of packet losses. In practice, shorter packets such as periodic beacons experience lower error probability [15], and we adjust the error probability for these packets according to Eq. 5. Clearly, link cost is not a function of distance in this model.

In some of our simulations, we compare NADV against another scheme called *blacklisting* [32, 13]. This scheme uses a fixed threshold, and when selecting a next hop, a node excludes neighbors with low-quality link based on the threshold. For example, if we use a threshold value of 0.5, then a node excludes neighbors that are closer to the destination and belong to the lower half in the link quality. Among the remaining neighbors, the blacklisting scheme selects the neighbor with largest ADV.

4.2 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) [33]. 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 [30]. 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 3, we calculate the *medium time* for an exchange sequence of a L_{DATA}-byte data frame as follows:

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

4.3 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 $2 \le n \le 6$). Suppose that 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 [24, 31], we assume in this paper that the transmission power is restricted to one of L levels in the set $P = \{p_1, p_2, ..., p_L\}$. In this scenario, it is best for a node to use the smallest power level no less than $d^n S_{min}$:

$$p_{tx} = min\{p_m : d^n S_{min} \le p_m,\ 1 \le m \le L\}. \quad (9)$$

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 [31], 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 [19, 12, 28] and assume that each packet forwarding consumes the following amount of energy:

$$C_{power} = 1 + c\,p_{tx}, \quad (10)$$

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 WISE can retrieve. Actual c values of different interfaces range between 0.17 and 1.30, and we use $c = 1.0$ in the simulation [28]. Previous measurement results show that the energy consumption of an idle or receiving wireless interface card is comparable to that of a transmitting one [34]. Although NADV can include these aspects of such energy consumption, for ease of comparison against an existing scheme, we focus on the energy consumption due to transmission. In this model, link cost is a non-decreasing step function.

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. 10 as power consumption equation, the current node selects the neighbor that minimizes the following formula:

$$E = (1 + c\,t_{px}) + \frac{D}{R}(c(n-1))^{\frac{1}{n}} + \frac{D}{R}c(c(n-1))^{\frac{1-n}{n}} \quad (11)$$

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

5. SIMULATION RESULTS

In this section we present the results of simulation experiments. We begin with the effect of wireless link errors. We first assume the perfect knowledge of link error rates when we investigate the performance. Then we compare the performance when we combine our proposed estimation techniques with NADV. We then 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.

5.1 Experiments with Perfect Estimation of Link Errors

We first present the results when nodes experience packet losses due to wireless link errors. In this section, we assume that there exists a perfect estimation scheme that provides accurate link cost

[7] Noise 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.

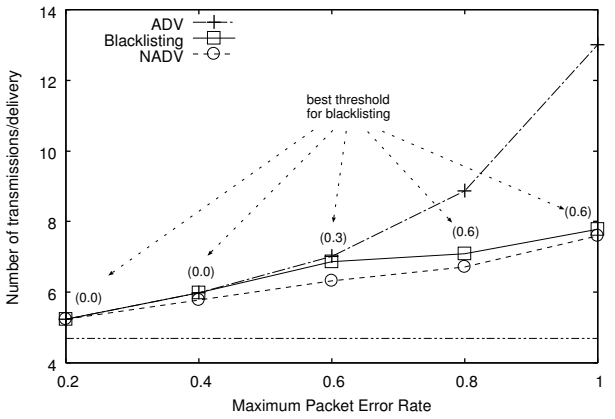


Figure 4: The number of MAC-level data transmissions per delivered packet with different degrees of packet errors. C_{error} is used as link cost. Unless the error rate is low, next hops chosen by ADV can cause multiple retransmissions, and ADV significantly increases the number of data transmissions. As marked in the bottom, when *max-PER*=0.2, the average path length using ADV is 4.7 hops.

values, and compare the performance when NADV and other geographic routing schemes operate based on the knowledge. We later present results when we combine NADV with the proposed PER estimation techniques described in Section 3.1.

In the first set of experiments, we use the random packet error model described in Section 4.1, where packet error rates are distributed uniformly at random between 0 and *max-PER*. Although in this model the frequency of links at a given error rate is similar to the previous measurement results in [25], this model does not consider the correlation between the distance and link error. As a result, on average, packets sent to distant neighbors have the same error probability as those sent to close neighbors. In fact, this setting is in favor of ADV, because in practice, transmissions to neighbors with large ADV are likely to suffer from frequent errors [18, 20]. We use an average of ten runs, each with different placement of stationary nodes. To avoid the packet errors due to the ARF algorithm, we fix the data transmission rate at 1 Mbps in this set of experiments.

In Figure 4, we report the number of MAC-level data transmissions (including retransmissions) per delivered packet for each scheme when we vary the value of *max-PER*. In this set of experiments, GPSR employs MAC-level failure notification, and all results are based on 100% packet delivery. We can observe that as the packet error rate increases, the data transmission overhead of ADV increases abruptly (up to 71% higher than that of NADV). This is because ADV often selects neighbors with low-quality link, which causes repeated retransmissions. In contrast, NADV intelligently avoids nodes with high PER, and although the data overhead of NADV increases as *max-PER* increases, the number of data transmissions is much smaller than that of ADV. Each transmission requires network bandwidth as well as node resources (e.g. battery power), and NADV uses system resources more efficiently.

In Figure 4, we also compare the performance of NADV against the *blacklisting* scheme described in Section 4.1. Blacklisting uses a fixed threshold, and to find the best threshold, we consider nine different blacklisting threshold values between 0.1 and 0.9 with an increment of 0.1. (The use of threshold value 0.0 in blacklisting corresponds to ADV.) In Figure 4, we plot the best result for blacklisting in each setting and mark the corresponding threshold value in parenthesis. We can observe that depending on the network environment, different threshold values lead to best results for blacklisting and that a fixed threshold value in blacklisting does not work well. When packet errors are frequent, it is better to use a high threshold value in blacklisting and exclude many neighbors with low-quality links. However, when there are few low-quality links, the use of a high threshold value may exclude useful neighbors and lead to longer paths. In contrast, NADV adapts to the changing network environment and is able to achieve low data transmission overhead in all cases.

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 4. Specifically, as *max-PER* changes from 0.2 to 1.0, the average packet latency of ADV increases from 54.9ms to 151.6ms. The performance degradation by NADV is less severe (increase from 54.9ms to 81.8ms). We later present more results on end-to-end latency. Instead of NADV, we also experimented using different combination of ADV and link cost, and NADV outperformed them as well. A more conservative link metric (e.g., $ADV/Cost^2$) results in longer paths, while a different metric such as $ADV/\sqrt{Cost}$ often underestimates high-cost links and causes more retransmissions due to packet errors.

In the previous experiments, we assumed the perfect knowledge of link cost. We next investigate the performance of NADV used with the proposed PER estimation techniques.

Name	Description
NADV-Beacon	Using periodic beacon messages (Eq. 5)
NADV-SNR	Using SNR (Eq. 6)
NADV-Self	Using own data packets (Eq. 7)
NADV-Perfect	Assuming the perfect knowledge of link cost

Table 4: Different scenarios when NADV and the PER estimation techniques are combined.

	max-PER		
	0.6	0.8	1.0
NADV-Self	7.0 (0.6)	7.8 (0.8)	8.9 (1.1)
NADV-Beacon	6.6 (0.7)	7.2 (0.6)	7.8 (1.2)
NADV-Perfect	6.3 (0.5)	6.7 (0.7)	7.6 (1.2)

Table 5: The number of data transmissions per delivered packet when NADV and proposed PER estimations are used. Values in parentheses are the standard deviations. When there are no link errors, the average path length is 4.7 hops.

5.2 Experiments using Proposed PER Estimation Techniques

In Table 4, we tabulate three schemes when we use NADV and the proposed PER estimation techniques together, in addition to case with the perfect estimation (NADV-Perfect). Note that none of the three estimation schemes use extra control messages. However, in the case of NADV-SNR and NADV-Beacon, we modify the periodic beacon format to include reverse link information, and the message length slightly increases. Storage overhead for the link cost estimation is also negligible since each node in GPSR already maintains neighbor information.

In the first set of experiments, we use the random packet error model used in the previous section and compare the actual per-

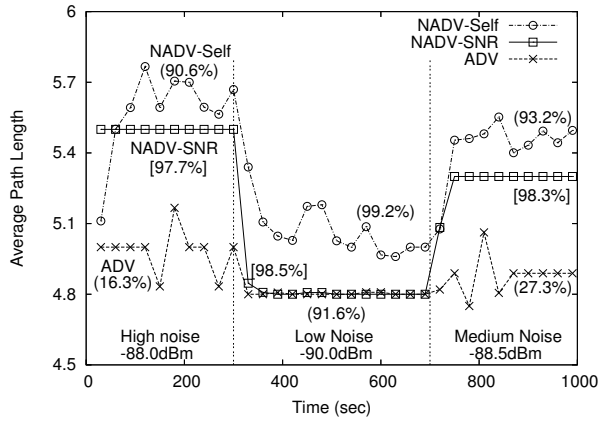


Figure 5: The average path lengths of NADV and ADV. The noise value changes from high (-88.0dBm) to low (-90.0dBm), and finally to medium (-88.5dBm). Numbers next to the lines are corresponding delivery ratios in each phase. PER estimation schemes enable NADV to choose appropriate neighbors and maintain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

formance of NADV against the case with the perfect knowledge of link cost. In this model, there is no correlation between SNR and packet error rate, and we experiment with NADV-Beacon and NADV-Self only. In Table 5, we show the data transmission overhead of NADV schemes for high-error scenarios. In this table, the overhead of NADV-Beacon and NADV-Self is reasonably close to that of NADV-Perfect, and we can infer that our proposed schemes provide good link cost estimation. Although NADV-Self has the most flexibility in deployment (e.g., no modification of protocol message format), its performance is slightly worse than NADV-Beacon.

In the previous experiments, we used the random packet error model. In the next experiments, we use the other error model described in Section 4.1, where a bit error occurs according to Eq. 6 (See Table 2). To identify the performance of proposed PER estimation schemes, we consider three simulation scenarios. First, we change the ambient noise power over time and observe how each estimation technique adapts to varying environments. Second, we increase the number of data flows to vary the network contention level. Third, we consider the scenario with node mobility.

Changing Noise Power

In these experiments, we investigate the adaptiveness of PER estimation schemes, and 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. In Figure 5 we plot the average path lengths for ADV and NADV when we vary the ambient noise power value. We do not employ MAC-level failure notification in GPSR, and the values in the parentheses show the average delivery ratios for different scenarios. Since the performance of NADV-Beacon is similar to that of NADV-SNR, we do not show the result of NADV-Beacon for clarity.

In Figure 5, the length of the path chosen by ADV is always shortest, but the packet delivery performance is always worse than that of NADV. For example, in the high-error scenarios (noise value=-88dBm), the difference in packet delivery ratio is more than

	Number of Flows			
	1	2	4	8
NADV-Beacon	98.6	98.7	97.3	97.6
NADV-SNR	99.2	99.3	97.9	98.1
NADV-Self	98.8	99.0	97.4	97.8
ADV	71.9	73.8	71.3	77.6

Table 6: Data delivery ratio (in %) when the number of data flows is varied.

81% (16.3% vs. 97.7%). In this 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 explicitly utilizes the link characteristic value, and in this simulation model, NADV-SNR exhibits more accurate estimation and faster convergence. NADV-Self occasionally employs slightly longer paths than NADV-SNR, but it is also able to adapt to environment changes.

Varying the Number of Data Flows

In the previous experiments, we use only one pair of source and destination. When we have more source-destination pairs, the network contention increases, in which the proposed techniques may estimate PER values incorrectly. For example, although received SNR values may be high, a node can experience high packet error rates due to increased collisions. To identify the performance of estimations schemes in this scenario, we vary the the number of data flows in the next set of experiments. We choose source-destination pairs uniformly at random. As in the previous experiments, we do not use the MAC-level notification of GPSR and report the average packet delivery ratio for each scenario. Among the values in Table 2, we fix the noise value at -89.2dBm, which is the closest to the median of noise measurements in our building.

In Table 6, we report the average data delivery ratios in different scenarios. In these experiments, all NADV schemes perform similarly. Although the delivery ratio appears to decrease when we increase the number of flows, the amount of degradation is small. When we experimented using 16 flows without packet errors, we observed low delivery ratios due to the network saturation, and eight data flows in this setting corresponds to relatively high network utilization. These results show that our proposed estimation techniques work well in the presence of high network load. When we use ADV, the average delivery ratio lies between 70% and 80%, depending on the random node placement.

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 pair does 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.

In Figure 6, we present the end-to-end latency results with varying mobility. As mentioned before, the data transmission overhead shows a similar trend to Figure 6. We observe that average latencies increase as node mobility becomes higher. This is because frequent

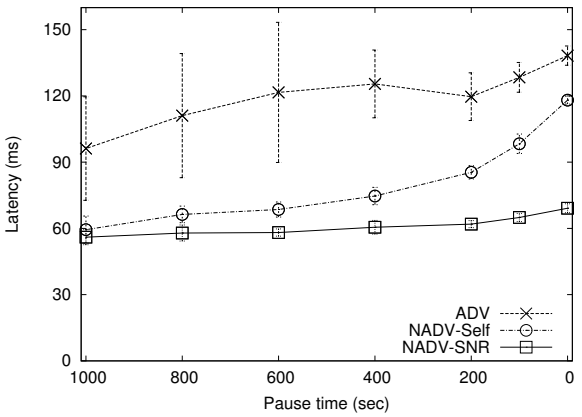


Figure 6: Average end-to-end latency when nodes are mobile. C_{error} is used as link cost. We changed the pause time for different degrees of mobility. NADV and the proposed cost estimation schemes are effective with node mobility.

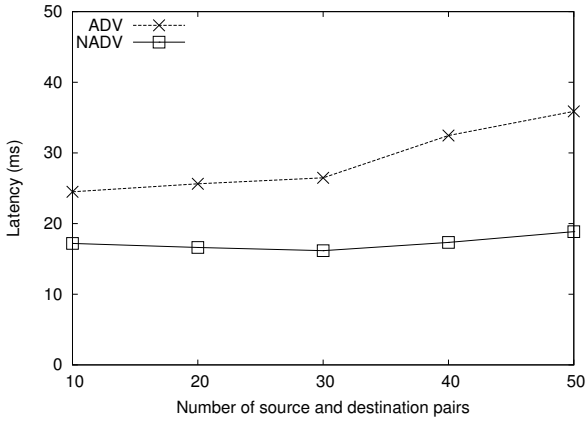


Figure 7: Average end-to-end delay with multiple flows. ARF and C_{delay} are used. When NADV is used, the network can support more flows without significant increase in the latency.

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 7: Average end-to-end latency (in ms) with different source-destination distances.

link failures cause more retransmissions. Compared to ADV, both NADV schemes achieve lower 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% lower 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. NADV-Beacon also requires a certain number of beacon messages for good estimation, and the results are similar to those of NADV-Self (within 5% difference in all cases), which we do not show here for clarity.

To summarize, the proposed link estimation schemes are effective even with node mobility, and NADV combined with them provides an efficient and adaptive geographic routing strategy. 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.

5.3 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 error model using Eq. 6, and ARF is used for rate adjustment. In this model, due to the interaction with ARF, link cost is not a convex function. 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 used, and the delivery ratios are 100% in all cases. Each value in this experiment is an average of ten runs.

In Table 7, 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. Compared to ADV, 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.

When using *NADV*$_{delay}$ in this simulation scenario, 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 the medium time, it may degrade the overall performance, especially when multiple traffic flows exist in the network. To investigate this potential problem, we perform experiments using different numbers of source-destination pairs, which we select uniformly at random.

In Figure 7, 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 wireless 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 load. Specifically, in the case of 10 flows, NADV decreases the average latency by 30%, but with 50 flows the improvement is 48%.[8] In the case of 50 flows, only 2 flows experience slight increase ($<$ 2ms) 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 in the network.

[8]In the experiments for Table 6, we use the fixed data transmission rate of 1 Mbps, and we observe network saturation when we send more than 8 packets per second. In the experiments for Figure 7, the data transmission rate can be up to 11 Mbps, and NADV can support more data flows without network saturation.

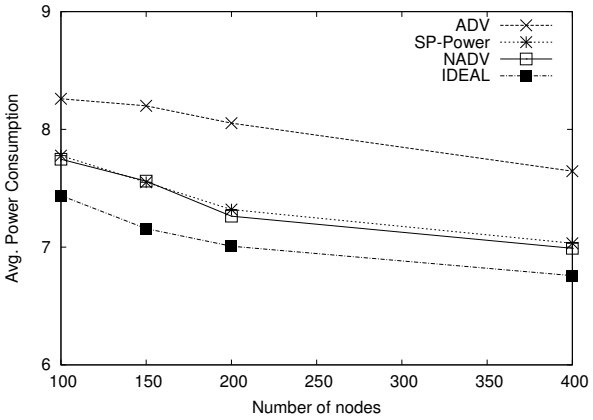


Figure 8: Average power consumption with different schemes. In dense networks, as more neighbors are available, power consumption decreases. The power consumption values by NADV and SP-Power are similar, which are close to the optimal value.

5.4 Using Power Consumption as Link Cost

We compare *NADV* ($\equiv NADV_{power}$) against the metric proposed in the *SP-Power* scheme [12]. When the power consumption equation is $C_{power} = 1 + c\,t_{px}$, NADV needs to know the current transmission power p_{tx}, which we assume is available through a control mechanism. SP-Power requires the exact value of path loss exponent, which we also assume is available. In practice, however, the path loss exponent estimation is not trivial, and depending on the measurement parameters, the estimated values can vary significantly [24, 35]. 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 compare the performance of optimal paths found by the centralized algorithm.

In Figure 8, 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 consumption.[9] 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 settings (e.g., distance, continuous power adjustment, different path loss exponents, and proportionality constants c). For other aspects of energy consumption (e.g., in idle or receiving mode), we expect that $NADV_{power}$ and SP-Power will consume a similar amount of energy and that their performance will be close to each other as well.

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 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 [21, 24].

[9] In Figure 8, the performance difference between the optimal case and ADV is not large. It is because the constant term in Eq. 10 constitutes a significant power consumption regardless of the transmission power, as is the case with most existing products [28].

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

Table 8: The average costs of paths found by respective routing schemes when link costs change.

5.5 Experiments with Generic Cost

Recently, new metrics are being proposed for various multihop routing purposes. For example, Draves et al. [36] propose the WCETT (Weighted Cumulative Expected Transmission Time) metric to improve network throughput in multi-radio mesh networks. As multihop wireless networks become more widely used for different objectives, we expect to see other new routing metrics proposed to achieve specific goals. In this section, we apply NADV to a generic cost metric to see whether the use of NADV can be generalized to other types of link cost. We use the following link cost:

$$C_{generic} = 1.0 + r\left(\frac{d}{R}\right)^2,\ 1 \le r \le 5, \qquad (12)$$

where r is a uniformly distributed random number, d is the distance between two nodes, and R is the maximum transmission range. The above link metric attempts to capture both the correlation with distance and the random property of link quality [25, 20]. In this subsection, we assume the availability of accurate and up-to-date link cost information.

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, node mobility), 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 [11], we modify the AODV simulation code, such that AODV finds paths that minimize the sum of link costs along the paths, not hop count.

In Table 8, 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. In this table, 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.

6. RELATED WORK

Many ideas and techniques have been proposed to find minimum-cost paths in multihop wireless networks, and energy-efficient routing has been an area of intensive research. Rodoplu et al. [19] present a localized algorithm that preserves network connectivity and achieves the globally minimum-energy topology. In PARO [37], 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. [38] find a set of routes that maximize the system lifetime. More recently, wireless link errors has drawn much attention in multihop wireless networks [18]. Banerjee et al. [22] 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 their experiment results show that paths with smaller ETX perform better than shortest paths [15]. These techniques and metrics above typically focus on table-driven or on-demand routing protocols (e.g., AODV [11]); in contrast, our work provides a general framework to incorporate these metrics into geographic routing.

Traditional geographic routing schemes use only geometric information such as the length of projection (called *progress*) and angle value against the straight line between source and the destination (please see [5] and the references therein). Instead of a straight line, Niculescu et al. [4] propose a forwarding strategy based on a pre-defined curve. More recent schemes consider link costs in the next hop selection. Stojmenovic et al. [12] propose a routing metric for power-efficient routing, as discussed in Section 5. Seada et al. [13] focus on the minimum energy consumption in lossy environments and propose threshold-based schemes as well as a link metric in Eq. 4. Zorzi and Armaroli also independently propose the same link metric [14]. Our work is different from them in that we present a more general framework and provide the rationale behind the use of NADV by proving the optimal tradeoff between hop count and link cost.

Greedy forwarding using NADV still can result in the local minimum problem. To route packets around voids, we can use existing recovery schemes with the NADV metric. For example, *Face Routing* [1] uses the right-hand rule in Gabriel graph, and 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 GOAFR+ considers link cost, it still chooses the neighbor closest to the destination in greedy mode.) Since these recovery schemes are independent of greedy forwarding, the use of NADV does not affect the recovery performance. Also, we believe that compared to ADV, greedy forwarding using NADV will have a similar frequency of encountering voids in a given network. The NADV metric can also be use in 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].

7. CONCLUSIONS AND FUTURE WORK

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.

Cost-aware routing schemes including NADV benefit greatly from fast and accurate link cost estimation, and we plan to investigate this issue further in the future. In this paper, we have treated 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.

8. ACKNOWLEDGMENTS

The authors thank Minho Shin for providing noise measurement data.

9. 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, July 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, Oct. 2001.

[11] 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.

[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] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 2004, pp. 108–121, ACM Press.

[14] M. Zorzi and A. Armaroli, "Advancement optimization in multihop wireless networks," in *Proceedings of VTC*, Oct. 2003.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Volkan Rodoplu and Teresa H. Meng, "Minimum energy mobile wireless networks," *IEEE JSAC*, vol. 17, no. 8, pp. 1333–1344, Aug. 1999.

[20] Marco Zuniga and Bhaskar Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proceedings of IEEE SECON*, Oct. 2004.

[21] IEEE 802.11h Standard, "Part 11. Amendment 5: Spectrum and transmit power management extensions in the 5GHz band in Europe," 2003.

[22] 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.

[23] Andreas Kopke, Andreas Willig, and Holger Karl, "Chaotic maps as parsimonious bit error models of wireless channels," in *Proceedings of Infocom*, Apr. 2003.

[24] Theodore Rappaport, *Wireless Communications: Principles and Practice (2nd Edition)*, Prentice Hall, 2001.

[25] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris, "Link-level measurements from an 802.11b mesh network," in *ACM SIGCOMM*, Sept. 2004.

[26] Seungjoon Lee, Suman Banerjee, and Bobby Bhattacharjee, "The case for multi-hop wireless local area network," in *Proceedings of Infocom*, Mar. 2004.

[27] 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.

[28] 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.

[29] 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.

[30] IEEE 802.11 Standard, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1999.

[31] "Cisco aironet 350 series client adapters data sheet," June 2003, Cisco Systems Inc. Available at http://www.cisco.com/.

[32] Omprakash Gnawali, Mark Yarvis, John Heidemann, and Ramesh Govindan, "Interaction of retransmission, blacklisting, and routing metrics for reliability in sensor network routing," in *Proceedings of the First IEEE Conference on Sensor and Ad hoc Communication and Networks*, Santa Clara, California, USA, October 2004.

[33] A. Kamerman and L. Monteban, "WaveLAN-II: A high performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, 1997.

[34] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, 2002.

[35] 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 raiodtelephone," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 4, pp. 721–730, November 1991.

[36] Richard Draves, Jitendra Padhye, and Brian Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. 2004, pp. 114–128, ACM Press.

[37] 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.

[38] Jae-Hwan Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proc. of IEEE Infocom*, 2000.