GPER: Geographic Power Efficient Routing in Sensor Networks*

Shibo Wu, K. Selçuk Candan
Department of Computer Science, Arizona State University, Tempe, AZ, USA
E-mail: {shibo.wu, candan}@asu.edu

Abstract

This paper presents a new Geographical Power Efficient Routing (GPER) protocol for sensor networks. Each sensor node makes local decisions as to how far to transmit: therefore, the protocol is power efficient, highly distributed and scalable. In GPER, given a final destination, each node first establishes a sub-destination within its maximum radio range. The node, however, may decide to relay the packet to this sub-destination through an intermediary node, if this will preserve power. Furthermore, this intermediary node may act independently and alter the subdestination based on its own power range and neighborhood status. Simulation results show that the routing power consumption using GPER is close to optimal obtainable based on full knowledge of the network. GPER provides 60%-90% savings over other power-sensitive routing solutions. For sensor networks with highly varying node densities, we propose an extension, GPER-2, which captures the network topology better. Simulations show that although GPER works well, GPER-2 can improve on GPER upto 20%, especially when variations are large.

1. Introduction

The applications of wireless sensor networks vary from personal area networks, where all wireless devices are located physically close to each other (in potentially architected configurations), to wide-area networks, where sensors are placed (potentially randomly) on a very large open terrain for *in situ* observations. The advance of low-cost, miniaturized sensor technologies makes it possible to deploy a large number of detection equipment on an unknown and uneven terrain for various measurement and surveillance applications. The *sensor nodes* are delivered and scattered on a specific region and *prime nodes* act as conduits between sensors and the external data processing units. Various types of data collected and filtered by the sensors are delivered to the prime nodes (nodes that are connected to an external network) via self-organizing wireless sensor networks. These sensor networks serve as the information conduit between the sensing devices and the deliberative and reactive processes that lie within or outside the network. In wide area wireless sensor networks, such as the next generation smart dust-style sensing environments, the number of sensors deployed in the system can be large. The network therefore has to function in a fully distributed and scalable manner. Furthermore, in most cases, it is impractical to replace or recharge the batteries of the already deployed sensors. Therefore, network protocols must preserve power.

In this paper, we introduce the *Geographical Power Efficient Routing* (GPER) protocol, in which each node makes local decisions as to how far to transmit the data; therefore, the protocol is highly power efficient, distributed, and scalable. In GPER, given a final destination, each node establishes a sub-destination within its immediate neighborhood, defined as the maximum distance it can transmit to. The packet may, however, be transferred to an intermediary relay node if this is likely to preserve power. This intermediary node, then, may alter the subdestination based on its own radio range and its neighborhood status. The results presented in Section 4 show that, for networks with uniform or close to uniform node distributions, the routing power consumption using GPER is close to the optimal power consumption obtainable with full knowledge of the network. In order to better deal with networks with varying node densities, we extend the GPER protocol and introduce an *overlay based routing protocol* (GPER-2) that captures the network topology. Section 4 shows that although GPER works well in networks with node density variations, GPER-2 can further improve on GPER's routing results.

1.1. Wireless Network Model

In this section, we provide an overview of the network model GPER relies on:

* This work is partially supported by NSF grant # 0308268

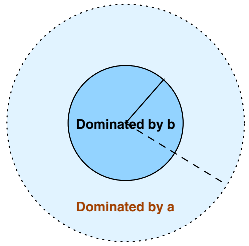


Figure 1. The power model $\rho = a \times \delta^{\gamma} + b$: in short distances the constant term, b, is dominant [12]; in longer distances, $a \times \delta^{\gamma}$ dominates[4, 10, 21].

- A set, S, of nodes is located in a two dimensional geographic area, G. Each node $v_i \in S$ has coordinates, $coord(v_i) = \langle x_i, y_i \rangle$. For the sake of simplicity, we assume no two nodes are colocated.
- Each node knows its own coordinates. This can be achieved either through an internal GPS device or through a separate calibration process.
- The location of a node acts as its ID and its network address. Therefore, there is no need for a separate ID establishment protocol. omni-directional or uni-directional relaying of packets. Each packet is marked with the location of the next hop and the corresponding node picks up the packet. We use "transmit to node v_i", as a shorthand for "transmit towards the location $coord(v_i)$".

Our network model, therefore, is similar to the network models in [7, 23, 19]. A noted difference is with the GeRaF network model [25], where nodes may turn off to save power. In GeRaF, when a node wants to transmit a packet towards a destination, it broadcasts a message in its entire radio range. Depending on the status of the nodes close to the target, zero or more active nodes will receive the message; hence the actual node that will receive is not known a priori by the sender, but rather is decided (probabilistically) after the transmission has taken place, according to nodes' own locations towards the destination. In this paper, we do not consider the case where nodes can turn on and off to save power. However, we note that since, in GPER, nodes are identified by their locations, GPER protocol in this paper can also be extended using contention resolution and retransmission protocols to reduce the *active* time of the nodes as in [25].

1.2. Power Model and Effects on Routing

Each node functions with the support of a battery and has a limited power. Furthermore, each node is able to adjust its transmission power but can not exceed a maximum.

In this paper, we use the commonly accepted channel path loss model, $\rho = a \times \delta^{\gamma} + b$, where ρ denotes the transmission power and δ denotes the distance between the sender and the receiver [4, 10, 21, 23, 11, 19, 12]. Here, γ, is the power loss constant and is typically between 2 and 4 [18]. a and b are the distance-relative and constant terms of the power consumption.

In a large body of work, the constant term, b, is assumed to be negligible [4, 10, 21, 23, 11, 19]. In [12], however, it is shown that in very short distances b can be quite significant compared to $a \times \delta^{\gamma}$. Therefore, we model the power consumption as shown in Figure 1:

- Each node, v, has a maximum communication range, $range(v)$. We call the set of nodes within this range the *neighborhood* of v and denote as $N_v (\subseteq S)$.
- Each node, v, has a communication range, $inner(v)$, within which the constant term b is dominant. We call the set of nodes within this range the *inner neighborhood* of v and denote this set as $I_v (\subseteq N_v)$.

The overall power consumption in the network, is based on the values of a, b, γ, and the density of nodes.

Let us consider a situation, where there are three nodes: a source, s, a destination, d within the range of s, and a third node, c, halfway between s and d (Figure 2). Let us denote the distance between s and d as δ. If for example the power loss constant is 4 and a and b are 1 and 0 respectively (the two nodes are not too close to each other), then, the power consumption, $\rho(s, d)$, for s to send a packet directly to d is $\rho(s, d) = \delta^4$. If s sends the packet to d through c, on the other hand, the total power consumption will be

$$\rho(s, d) = \delta(s, c)^4 + \delta(c, d)^4 = 2 \times (\frac{\delta}{2})^4 = \frac{1}{8} \times \delta^4.$$

Therefore, in this example, using c located between s and d as a relay rather than forwarding the message directly to the destination can reduce the power consumption 8 times. Obviously, this example assumes that although d is within the radio range of s, these two nodes are not too close to each other (i.e., the constant term b is negligible). If, on the other hand, d was very close (and hence the constant term b was dominating the power consumption), then s should transmit to directly to d [12]. Therefore, we can summarize the advantages and disadvantages of using an intermediary relay node vs. transmitting the package as far as possible within the radio range as follows:

- *Advantage:* If the distances are sufficiently long and the term b is less significant than $a \times \delta^{\gamma}$, using an intermediary node may save power.

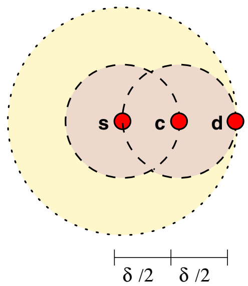


Figure 2. The effects of using an intermediary node for routing

- *Advantage:* In both uni-directional and omni-directional radio scenarios, using high power transmission increases the network contention (Figure 2) and reduces the utilization of network [6, 19].
- *Disadvantage:* Using more hops may increase the total end-to-end transmission delay.

In this paper, we focus on the total power consumption. Therefore, we see that under right conditions, using relays may benefit the network. Furthermore, smaller-ranged transmissions may reduce network contention and may help improve the end-to-end delay.

1.3. Contributions of this Paper

In this paper, we introduce the *Geographical Power Efficient Routing* (GPER) protocol. The contributions of this work are as follows:

- A protocol *RouteWithinNeighbors* that enables each sensor to choose the best next node in its radio range (Section 2.1).
- The GPER protocol, which builds on *RouteWithinNeighbors* and novel dynamic subdestination adjustment and forced routing techniques, to establish routes for destinations that are not within the radio range of the source (Section 2.2).
- The GPER-2 protocol, which creates an overlay on top of the wireless network and emulates GPER on this overlay to achieve improved routing in networks with large variations in density (Section 3).

Section 4 shows that GPER and GPER-2 achieve routing power consumption that is close to the optimal power consumption. GPER works well in networks with limited node density variations and GPER-2 further improves these results when the variations are large.

In the next subsection, we provide an overview of the literature.

1.4. Related Work

Several metrics [20] have been proposed for measuring power consumption. The most commonly used metrics are energy consumption for each packet [23, 21, 19, 10, 18] and the system lifetime [2, 11]. In the literature, power consumption is assumed to be distance-sensitive [4, 10, 21, 23, 11, 19]. In [12], however, it is shown that in very short distances constant power consumption may dominate.

In wide area wireless sensor networks with large number of nodes, traditional table-driven and on-demand routing protocols are not directly applicable [16, 5, 14, 13, 15, 17]. First, for wide area sensor networks, the global routing table can grow unmanageably large. Secondly, exchanging routing tables until a stable state is reached becomes unacceptable for sensor networks with large number of nodes. In [19] and its variant [10], the route table is generated by running Bellman-Ford algorithm; however each node only exchanges route tables with a subset of its neighbors, which is called the enclosure of this node. Sending messages directly to a node outside the enclosure will cost more energy than forwarding the message through nodes in the enclosure. In [11], the best path is chosen among the minimal power consumption paths and paths that maximize minimal residual power with the trade-off determined by a parameter. This algorithm is centralized because each node must know the remaining power of all nodes and power consumption to transmit a packet along any two nodes in the network. In [2], the authors developed a flow redirection algorithm to take message flow from the path of shortest lifetime and give it to the path of longest lifetime. To calculate the lifetime of the nodes, the message rate must be known. To reduce the information exchange overhead in wireless networks, several on-demand routing protocols have been proposed [5, 14, 15]. In its most basic form, an on-demand routing protocol will flood a route discovery message into the network and obtain the best path to the destination in the response from the destination. This will cause significant overhead if the number of nodes is large. A number of methods have been proposed to constrain the number of nodes that will rebroadcast the route discovery message [8].

Geographic routing is favored in sensor networks since the coordinates of the nodes actually imply the topology of the network. In some cases, e.g. data centric storage in which event data are hashed to geographic locations by event type [7], geographic routing is demanded. In [21], the topology of the network is assumed to be known for each node. k minimum energy node-disjoint and link-disjoint paths are calculated by executing a minimum weight k node-disjoint paths algorithm. In [7], a greedy routing algorithm called GPSR aiming at minimizing the number of hops in a mobile wireless network is proposed. The algorithm chooses the neighbor closest to the destination as the next hop. The work in [9] is similar to GPSR in that their *face routing* is actually one type of *planar perimeter routing* in GPSR. The focus of the GPSR is to minimize the number of hopes in the network and maximize the data packets transmitted successfully. Our focus, on the other hand, is minimization of the power usage. In [23], the routing algorithm selects the neighbor whose relay region covers the destination as the next hop. This works pretty much the same as GPSR and does not provide much power savings compared to GPSR.

Finally, we note that like most state-of-the-art algorithms, given a pair of source and destination nodes which communicate often, GPER would follow more or less the same low-cost path each time. Consequently, the nodes on this path could quickly run out of power. To help avoid such repeated use situations, we are currently investigating multi-path extensions to the algorithms presented in this paper.

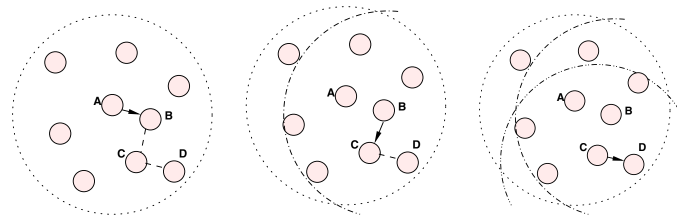
2. GPER: Geographical Power Efficient Routing

GPER consists of two complementary protocols. The first one aims to identify the best next hop within the nodes that are in the radio range of the source. Unlike the others in the literature [7, 9], this algorithm not only considers which neighbor node is closest to the destination, but also how much power can be saved if intermediary relays are used to reach this neighbor. The second algorithm aims to establish routes between nodes that are not in the radio ranges of each other. This algorithm builds on the first one by introducing novel dynamic subdestination adjustment and forced routing techniques.

2.1. Power Efficient Routing within the Radio Range

Given a source, s, and a destination node, $d \in N_s$, within the neighborhood of s (i.e., within the radio range of the source node), the minimum energy consuming transmission strategy may either be

- to directly transmit the message to d, if d is in the inner range of s ($d \in I_s$) where the constant power consumption term is dominant, or



(a) (b) (c)

Figure 3. Example: routing within the radio range: to save power, A uses an intermediary node instead of directly transmitting to, D, which is in its neighborhood.

RouteWithinNeighbors(d)
- s is the current node (the node in which this routine is executed at);
- d, within the radio range of s, is the destination node
- this routine identifies the next node on the minimum power consumption path from s to d

{
1. if $d \in I_s$, then the next node is d itself; return;
2. s constructs a neighborhood graph $NGraph_s = (V_s, E_s, l)$ such that
   - all the nodes in V_s are neighbors of s; i.e., all the nodes in V_s are within the radio range of s;
   - there is an edge in E_s for every pair of nodes in V_s; and
   - for each edge $e \in E_s$, the label $l(e)$ gives the expected power consumption between the end nodes of e
3. s runs a shortest path algorithm on its neighborhood graph $NGraph_s$ with s as source and d as destination;
4. the routine returns the next node on the path between s and d;

}

Figure 4. Algorithm used by node s to identify the next relay node on the path to node d, when the node d itself is within the neighborhood (within the radio range) of s

- to use an intermediary node to relay the message to d, if d is not in the inner range ($d \in N_s$ and $d \notin I_s$) and, hence, the distance-sensitive power consumption term is dominant.

The following example demonstrates this second case. Consider the scenario in Figure 3. In this figure, node A wants to send a message to node D, which is in A's radio range. Let us assume that, based on its power status and its knowledge about the neighbors in its radio range, A discovers that the minimum energy path to D is $A \to B \to C \to D$ and forwards the packet to B (Figure 3(a)). Next B finds out that the minimum energy path from B to D is $B \to C \to D$ and forwards the packet to C (Figure 3(b)). C in turn forwards the packet to the destination D (Figure 3(c)).

The task of the source node s, then, is to select the location of the next relay node in its radio range. When this neighborhood is sparse, it may be possible for each node to maintain a view of its neighborhood[1]. In this case, s can find the next relay node using a shortest path algorithm (where the edge weights correspond to the expected power consumption based on the power model $a \times \delta^{\gamma} + b$. Figure 4 depicts the outline of the *RouteWithinNeigbors* algorithm for power efficient routing within the radio range.

Unfortunately, the neighborhood graph, $NGraph_s$, constructed by s in *RouteWithinNeighbors* may not reflect the actual topology of the nodes in V_s, since it is possible that the distance of two nodes, $v_i, v_j \in V_s$ exceeds their respective radio ranges. In this case, communication through the edge $\langle v_i, v_j \rangle \in E_s$ between these two nodes may be physically impossible. This problem can be addressed by letting s to periodically learn the power ranges of nodes in its neighborhood (power range) and eliminating the invalid edges in E_s. If, on the other hand, the power range of the nodes in the network are constant and identical, such a corrective action is not necessary as the minimum energy consumption path between s and d will never contain a physically impossible edge. To see this, consider the following: suppose nodes, v_i and v_j, and the edge $\langle v_i, v_j \rangle$ appears in the minimum energy consumption path from s to d. If $\rho(v_i, v_j) > \rho(s, d)$, then s could transmit directly to d.

If the number of nodes in the radio range is small, each node can use *small* routing tables for its immediate neighborhood. During initialization, each node runs the algorithm once for the small number of nodes in its neighborhood and saves the coordinates of the first relays for each one.

2.2. Power Efficient Routing outside the Radio Range

RouteWithinNeighbors is applicable when the destination is within the radio range of the current node. In order to leverage *RouteWithinNeighbors* when dealing with the more general case where destination node is beyond the radio range, we introduce the concept of *dynamically adjusted subdestination* nodes. Given a destination node, d, the source node, s, selects the node u, within its neighborhood, closest to d as the subdestination. In a sense, this is similar to the GPSR [7] protocol which tries to minimize the number of hops that the packet takes by *greedily* choosing the neighbor closest to the destination. On the other hand, in GPER, once the subdestination node, u, is identified, *RouteWithinNeighbors* can use local relay nodes to efficiently deliver the packet to u instead of directly transmitting it to u. Furthermore, as described next, each local relay node may make dynamic adjustments on the subdestination; hence the packet may skip u altogether on its way to d.

1 Since the number of nodes in the radio range of a given node is much smaller than the total number of nodes distributed in a wide area wireless network, it may be possible for each node to learn and save the coordinates of its neighbors (N_v and I_v) at the system initiation phase.

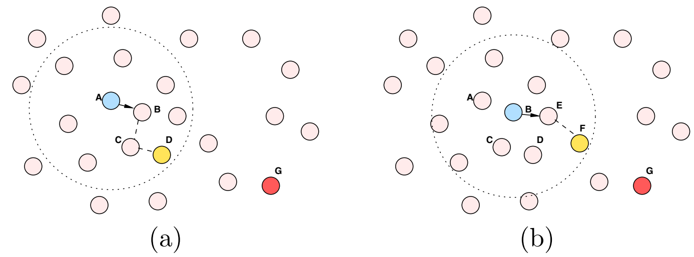


Figure 5. Dynamic subdestination selection: (a) A chooses D as its subdestination and, at the next step, B chooses F as its subdestination

2.2.1. Dynamic Subdestination Adjustment In *RouteWithinNeighbors*, the source node sets up a *locally optimal* minimum power consumption path to its subdestination *assuming* that the packet will be forwarded through the nodes on the chosen path. On the other hand, *irrespective of the assumptions made by this node*, the next node acts independently and calculates a new subdestination based on its own power range and neighborhood. Therefore, rather than committing to a fixed subdestination until it is reached, each relay node makes adjustments and prevents costly deviations from the destination due to earlier misjudgments.

Figure 5 shows an example. In this figure, the source, A, is trying to route a packet to destination G. For this purpose, it chooses a subdestination, D in its radio range, and a path $A \rightarrow B \rightarrow C \rightarrow D$ to this subdestination. It then forwards the packet to the next hop on this path, which is B (Figure 5(a)). Once it receives the packet, B considers its own neighborhood and establishes F as its own subdestination for reaching G (Figure 5(b)). B then chooses the path $B \rightarrow E \rightarrow F$ and routes the packet to E instead of to C as was assumed priorly by A. This dynamic subdestination selection process will continue as E considers its own neighborhood for the next step and the packet may never need to reach the node D chosen as the subdestination by the source node A.

2.2.2. Forced Routing to Prevent Infinite Loops As described above, after s identifies its subdestination u to reach d, it calculates a low power

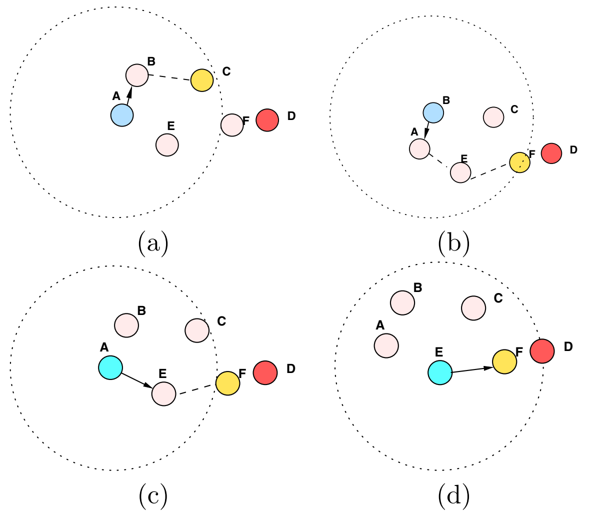


Figure 6. A loop due to dynamic subdestination selection and force routing solution: (a) A sends the packet to B and (b) B sends it back to A and (c) A sends it to E under force routing and (d) E sends it to F under force routing

route to u using *RouteWithinNeighbors*. In most cases, the next relay node r on the resulting path will be closer to the destination d than s. However, if r is further to d than s, forwarding the packet to r may lead into a loop. Figures 6 (a) and (b) illustrate how this can happen: in Figure 6(a), B is closer to D than A. First, A chooses C as the subdestination to D and forwards the packet to B; then B chooses F as the subdestination to D and forwards the packet back to A, since A is on the minimum energy consumption path from B to F. This causes an infinite loop between node A and B.

In order to guarantee that the routing algorithm is free of such infinite loops, in GPER we introduce a *forced routing* mechanism as a preventive measure. Forced routing is applied when a potential loop is identified: let us assume that s is trying to route a packet to the final destination d, and it identifies the subdestination u in its neighborhood. Furthermore, let us assume that s decides to route the packet to r on a low power path to u. If it turns out that the next hop r is further to the final destination d than s itself, s declares a *forced routing* status. Under forced routing, the subdestination u is kept fixed until it is reached. Furthermore, the *RouteWithinNeighbors* is augmented to constrain the path such that $\delta(r, u) < \delta(s, u)$, i.e. r is closer to u than s. The augmented version of the in-neighborhood routing algorithm is called $RouteWithinNeighbors_{forced}$.

It is possible that a given packet reaches the same node more than once even when the forced routing scheme described above is used. Each such loop is *temporary*. Consider Figure 6 once again. In Figure 6(b), when B realizes that A is further to the destination D than itself, it marks the packet as in force routing mode and fixes the packet's subdestination to F. After this, B may still forward the packet back to A, since A is closer to the newly fixed subdestination F than B. This time, on the other hand, A will recognize that the packet is in force routing mode with fixed subdestination F. Therefore, this time A will decide to use E instead of B as the next hop (Figure 6(c)). Figure 6(d), then, shows how the packet proceeds towards the fixed subdestination F. Therefore, loops of the form $A \rightarrow B \rightarrow A$ are temporary and the packet will approach its destination.

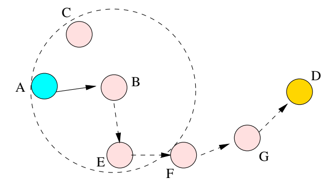


Figure 7. Lack of a suitable subdestination at B

To show that the result is indeed free of infinite loops, we need to show that the number of times each packet is forwarded is finite. Let us define $ForceRoutingRun$ as the period which starts when the packet enters force routing mode and ends when the packet reaches the corresponding fixed subdestination. During a $ForceRoutingRun$, each forward will bring the packet closer to the fixed subdestination, so the number of forwards in each $ForceRoutingRun$ is finite. Assuming that there is a subdestination closer to the final destination, then the total of forwards will be finite.

Therefore, forced routing guarantees that the path generated by GPER is free of infinite loops when the identified subdestination is closer to the destination than the current node. In an arbitrarily structured wireless network, however, there may be cases in which there may not be a suitable subdestination (Figure 7). In next subsection, we examine this.

2.2.3. Planar Perimeter With Forced Routing So far we assumed that each node can identify a proper subdestination within its neighborhood (i.e., radio range); however, there are scenarios in which no neighbor is closer to the destination than the current node; i.e., there are no suitable subdestinations. For example in Figure 7, node B realizes that all of its neighbors $\{A, C, E\}$ are further than itself to the destination D. Such scenarios have been extensively studied in [7] and planar perimeter routing is introduced to gradually forward the packet to a node closer to the destination than the current node. When there is no suitable subdestination, under perimeter routing, each packet traverses the graph of the network using a right-hand rule, which requires that if the node visited before B was A, then the next edge to traverse is the first counter-clock-wise edge about B from edge (A, B). In Figure 7, B would choose C as the next node to follow. To ensure that routes will be found when they actually exist, the graph of the network is planarized before the next edge to traverse is calculated [7].

In GPER, we adopt the planar perimeter routing approach presented in [7] to tackle the scenarios when no suitable subdestination is available. On the other hand, to save power, we implement perimeter routing approach through *RouteWithinNeighbors* which may use intermediary relay nodes when this helps reduce the power consumption. However, to ensure that *RouteWithinNeighbors* implements perimeter routing without making dynamic adjustments which may destroy the overall counter-clockwise progression of the perimeter routing approach, we also use forced routing along with *RouteWithinNeighbors*. Consequently, when the next hop for planar perimeter routing is determined (for instance C in the above example), the packet is also marked to be in force routing with this fixed subdestination (i.e., C will be the fixed subdestination until it is reached). The proof that there is no infinite loops under planar perimeter mode extended by force routing follows the fact that original planar perimeter routing has no infinite loops [7].

2.2.4. GPER Protocol Figure 8 presents the geographic power efficient routing protocol (GPER) which takes all the above issues into account.

Results presented in Section 4 show that GPER works well, especially for sensor networks that have uniform or close to uniform sensor distributions and the power consumption of the resulting routes are close to optimal. An important aspect of GPER is that this performance is achieved with only local information; that is, the coordinates of the neighbors are the only information needed. There is no need for exchanging routing tables with other nodes or broadcasting any route discovery messages. The overhead of GPER is the same as GPSR since GPER is using similar local information and forced routing in GPER does not incur any additional power consumption overhead.

The assumption of uniformity, however, does not always hold. Wireless sensor network topologies generally have heterogeneous sensor densities, therefore routing algorithm should scale to variations in sensor density. Again, as shown in Section 4, GPER works well even when the sensor density is not uniform. On the other hand, we see that we can further improve the power consumption by taking into account the topology of the network density.

$RoutePacket_s(pack, d)$

- s is the current node (the node in which this routine is executed at);
- *pack* is the packet being routed
- d, outside of the radio range of s, is the destination node
- this routine routes *pack* to the next node on the minimum power consumption path from s to d

{

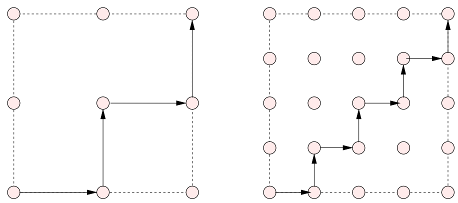
1. s finds the neighbor that is closest to d and save to n;
2. if (`ForcedRouting` flag is on for *pack*)
   (a) s extracts the `Subdest` of *pack* and save to n;
   (b) if ($s == n$)
      i. s clears the `ForcedRouting` flag for *pack*;
3. if (`ForcedRouting` flag is off for *pack*)
   (a) if (`PeriRouting` flag is on for *pack*)
      i. extract `PeriStartAt` from *pack*
      ii. if(s is closer than `PeriStartAt` to d)
         A. clear both `PeriRouting` and `ForceRouting` flags for *pack*
      iii. else
         A. set `ForcedRouting` flag on for *pack*
         B. calculate next node for planar perimeter routing and save to n
         C. s sets n to be the subdestination, `Subdest`, of *pack*;
   (b) else if (n is further than s to d)
      i. set both `PeriRouting` and `ForcedRouting` flags on for *pack*
      ii. set `PeriStartAt` field in *pack* to s
      iii. calculate next node for planar perimeter routing and save to n;
      iv. s sets n to be the subdestination, `Subdest`, of *pack*;
4. if(`ForceRouting`flag is on for *pack*)
   (a) $NextHop = RouteWithinNeighbors_{forced}(n)$;
5. else
   (a) $NextHop = RouteWithinNeighbors(n)$;
6. if (`ForcedRouting` flag is off for *pack* and $NextHop$ is further than s to d)
   (a) s sets `ForcedRouting` flag on for *pack*;
   (b) s sets n to be the subdestination, `Subdest`, of *pack*;
   (c) $NextHop = RouteWithinNeighbors_{forced}(n)$;
7. s forwards the packet to $NextHop$;

}

Figure 8. GPER packet routing algorithm used by node

3. Overlay Routing in Non-uniform Networks

In networks with large density variations, the selected routes may run across regions of very low sensor density. Since the density of the sensor distribution determines minimum power consumption, this may cause high power consumption. To see this, consider the two sensor networks, both 8×8, shown in Figures 9(a) and



(a) (b)

Figure 9. Two 8×8 sensor networks with different node distributions: (a) sparse and (b) dense networks

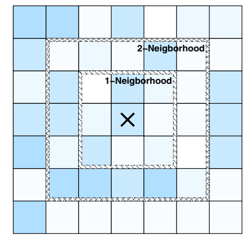


Figure 10. An overlay example: grid overlay, and 1- and 2-neighborhoods of a given cell in a grid overlay. Different shades denote different cell densities.

(b). There are 9 nodes in the network in Figure 9(a) and 25 nodes in the network in Figure 9(b). In both networks, the nodes are uniformly distributed in the space. Assuming that the power loss constant is 4, and the terms a and b are 1 and 0 respectively, the minimum power needed to route from one corner to another corner can be calculated as $4 \times 2^4 = 64$ for the network in Figure 9 (a) and $8 \times 1^4 = 8$ for the one in Figure 9(b). Therefore, we can easily see that the higher the density is, the lower the routing power consumption will be.

In networks with variations in the sensor density, it may be desirable to detour and choose regions of high density. Based on this observation, we augment our algorithm with the ability to route through densely distributed regions of the network. We call this *overlay routing*. The resulting GPER-2 protocol creates an overlay, with more or less uniform regions, on top of the existing wireless network and emulates GPER on this overlay to achieve improved routing in networks with large density variations. Several works [1, 3] create overlays for topology maintenance on sensor networks. Other related works on overlay routing can be found in [24, 22]. Next, we present an example overlay, grid overlay and describe how GPER-2 works using this example.

3.1. Example: Grid Overlay

sensors is divided into a grid of cells of the same size. The cell size is chosen to approximate uniform distribution within each cell (in the Experiments section, we will study the impact of this assumption). The result is a grid network, GN of cells, where each non-boundary cell has six neighbors. Each pair of adjacent grid cells (C_1, C_2) are connected with a logical link labeled with the expected power consumption for transmissions from nodes in C_1 to nodes C_2. We next describe how we compute the edge weights.

3.1.1. Compute Edge Weights in Grid Overlay

Let M be the total of number of nodes in a given cell, C. Assuming a uniform distribution of nodes in a each cell the number of nodes between the center of the cell and the center of one of the cell's borders can be computed as $M^{1/2}/2$. If L is the length of one of the borders, then the average distance between closest neighbors in the cell is $L/M^{1/2}$. The power needed to route a packet from the center of the cell to the center of one of its borders then be calculated as

$$\begin{aligned}\rho_{hor} &= \frac{M^{1/2}}{2} \times \left(a \times \left(\frac{L}{M^{1/2}}\right)^{\gamma} + b\right) \\ &= \frac{a}{2} \times \frac{L^{\gamma}}{M^{-(\gamma-1)/2}} + \frac{b}{2} \times M^{1/2}\end{aligned}$$

Therefore, the amount of power required to transmit a packet from a cell C_1 to one of its horizontal or vertical neighbors, C_2, can be computed as

$$\begin{aligned}\rho(C_1, C_2) &= \rho_{hor,1} + \rho_{hor,2} \\ &= \frac{a}{2} \times \frac{L^{\gamma}}{M_1^{-(\gamma-1)/2}} + \frac{b}{2} \times M_1^{1/2} + \\ &\quad \frac{a}{2} \times \frac{L^{\gamma}}{M_2^{-(\gamma-1)/2}} + \frac{b}{2} \times M_2^{1/2}\end{aligned}$$

Assuming a power constant ≥ 2, the power needed to route a packet from the center of the cell to one of the cell's corners can be computed as

$$\rho_{dia} = a \times \frac{L^{\gamma}}{M^{-(\gamma-1)/2}} + b \times M^{1/2}$$

Therefore, the amount of power required to transmit a packet from a cell C_1 to one of its diagonal neighbors, C_2, can be computed as

$$\begin{aligned}\rho(C_1, C_2) &= \rho_{dia,1} + \rho_{dia,2} \\ &= a \times \frac{L^{\gamma}}{M_1^{-(\gamma-1)/2}} + b \times M_1^{1/2} + \\ &\quad a \times \frac{L^{\gamma}}{M_2^{-(\gamma-1)/2}} + b \times M_2^{1/2}\end{aligned}$$

Note that the grid network forms a structured overlay on top of the less structured wireless network. Each cell in the grid network has 8 1-hop neighbors and 24 neighbors that are accessible in 2 cell hops (Figure 10). Hence, we can define neighborhoods that are analogous to neighborhoods in GPER on the overlay grid and use the GPER algorithm on this overlay grid network. Given a cell C_i in a grid network GN, the k-neighborhood, N_i^k, of cell C_i, then, is defined as the cells that are within k-hops on the grid network, GN. The k-neighborhood of a given cell contains every cell accessible through k cell borders.

3.1.2. Grid Overlay Formation The network identifies and disseminates the cell information (number of sensors in each cell and the cell densities) as follows: At the network initiation phase, the network is flooded with a REQUEST_LOCATION message. A prime node (a node that is designated as prime at the time the sensor network is built) or an external process accessible through the prime node collects REPORT_LOCATION messages and calculates the suitable cell size and grid location using the information it receives. The prime node, then, floods a FORM_CELLS message into the wireless network. This message contains (a) the size of the cells, (b) one of the network's corners' coordinates (this will enable each node to identify which cell it is located in based on its own location), and the (c) locations of the sensor nodes closest to the center of each cell (the ties are broken arbitrarily). Upon receiving the FORM_CELLS message, each sensor determines which cell it located in, using its own coordinates and the data contained in the message. Each sensor, then, sends a JOIN_CELL message, which contains its coordinates, to the node closest to the center of its cell using GPER. At the end of this process, the center node knows the number of sensors in its cell and it disseminates this information to the center nodes of its k-neighborhood using a REPORT_NEIGHBORHOOD message. Each center node then computes costs for its k-neighborhood and distributes this to the nodes in its cell using REPORT_CELL message.

At the end of this process, each node knows (a) which cell it is located in, (b) the location of the node closest to the center of its cell, (c) the average costs (computed based on densities) of sending a packet from the center of its cell to the center of the cells in the k-neighborhood. Note that k is usually very small (1 or 2) and the number of nodes in the k-neighborhood is small (25 for 2-neighborhood, $(2k+1)^2$ in general) and this data can be cheaply kept at each node.

3.2. Power Efficient Routing within the Overlay k-Neighborhood

The source cell constructs a path of cells by running a shortest path algorithm on the overlay network of cells in its k-neighborhood. This path of cells ensures that, if during actual routing the packets follow this particular sequence of cells, then the power consumption will be close to minimum. In a sense, this is analogous to the *RouteWithinNeighbors* algorithm presented in Section 2.1, but applied at the overlay level instead of at the wireless node level. For this purpose, the source node computes the next cell on the path locally and the ID of the next cell is added to the packet. Any node that receives this packet will forward the packet towards the center of the next cell. When the packet is close enough to the center of the next cell, this process is repeated until the destination cell is reached. If the next cell is the destination cell, instead of forwarding the packet to the center of the destination cell, grid routing protocol forwards it directly to the destination node.

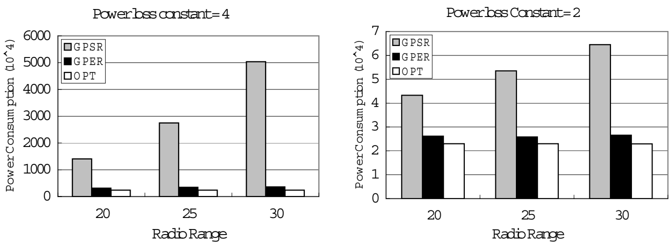
3.3. GPER-2: Power Efficient Routing outside the Overlay k-Neighborhood

In order to achieve power-efficient routing beyond the k-neighborhood, we employ an algorithm similar to GPER on the overlay network. Therefore, we call this algorithm GPER-2: at the low (node) level, GPER is used to deliver packets between wireless nodes and at the higher (overlay) level a GPER like algorithm is used to deliver packets between cells. At the overlay level, GPER-2 selects the cell (in the k-neighborhood) closest to the destination cell as the subdestination cell. It then calculates the next relay cell for this subdestination cell by running a shortest path algorithm in the k-neighborhood of cells. The packet is forwarded toward the center of the next cell until it is close enough, where a new next cell is calculated. This process continues until the destination cell is within the k-neighborhood of current cell. When cells are uniformly shaped and placed on a uniform grid, forced routing and perimeter routing may not be necessary.

GPER-2 gives an efficient way to disseminate packets when there are large variations in the density distribution of the network. As shown in the next section, with the aid of grid routing, power consumption in wireless networks with largely varying node densities can be significantly reduced.

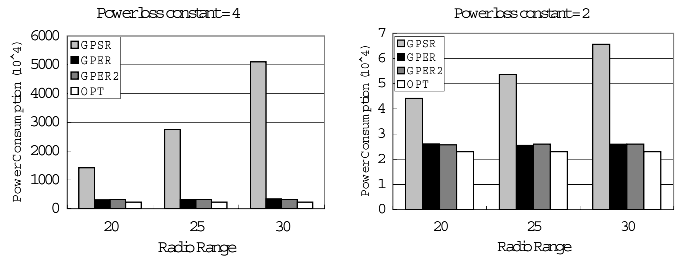
4. Experiment Results

In this section, we present experiment results that validate the efficiency and effectiveness of the geographic routing algorithms, GPER and GPER-2, presented in this paper.



(a) (b)

Figure 11. Comparison of routing power consumption for uniform sensor distribution (a) when power loss constant is 4 (b) when power loss constant is 2



(a) (b)

Figure 12. Comparison of routing power consumption for slightly non-uniform sensor distribution (a) when power loss constant is 4 (b) when power loss constant is 2

4.1. Routing in Networks With Uniform Sensor Densities

We first conducted a set of simulations to verify the GPER algorithm in uniform networks [19, 10, 23]. The setup for this set of experiments is as follows: We uniformly placed 30000 sensor nodes in an area of size 1600×1600. We varied the radio range between 20 and 30. A larger radio range means a network that is denser, in the sense there will be more nodes in each node's neighborhood (i.e., radio range). We also varied the power loss constant between 2.0 and 4.0. The terms a and b are set to 1 and 0 respectively. Overall, we experimented with six combinations of radio range and power loss constant. For each combination we ran 100 simulations to route a packet from bottom left most node to top right most node and calculated the average power consumption by using two routing protocols GPSR and GPER. The optimal routing power consumption is also calculated using a centralized shortest path algorithm which assumes full knowledge of the network and is included for comparison.

As expected, the number of nodes on GPSR paths are larger for the same routing task; yet, as Figure 11 shows, GPER saved almost 90% power relative to GPSR when the power loss constant is 4, and saved 60% when the constant is 2. The increase in the path length contributes to the delay, but not to the overall power consumption. We also see that in our experiments, the power consumption of GPER is close to the optimal values, with a maximum of 35% difference when power loss constant is 4 and a maximum of 15% difference when it is 2.

We note that as the radio range of the nodes becomes smaller, the power consumption of GPSR also gets smaller. In a sense, when the maximum transmission power is low, GPSR ends up having to route through closer nodes the way GPER naturally does, and as a results saves power. However, since GPER chooses the neighbors intelligently, GPSR's savings are nowhere close to GPER's. For GPER (as well as for the optimal route), the power consumption is not significantly affected by the radio range, as both of them already choose intelligently among available neighbors[2].

4.2. Routing in Networks With Non-uniform Sensor Densities

To observe the power consumption in networks with varying densities, we ran a second set of experiments.

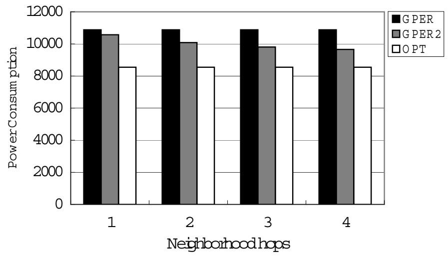
4.2.1. Routing in Lightly Non-uniform Networks In order to observe the effect of slight non-uniformities in the network, we experimented with GPER routing in slightly non-uniform networks. For these experiments, we divided the network into cells of size 100×100 and distributed the 30000 nodes, such that half of the cells received 150 nodes and half received 85 nodes. Figure 12 shows the comparison of power consumption to route a packet from a corner to the other through the diagonal. In this figure, we have also included GPER-2 routing results. The result shown in Figure 12 is similar to Figure 11. GPER power consumption is close to optimal in such sensor networks. We see that when the non-uniformities in the network are slight, GPER-2 routing is not necessary as GPER is already choosing very good routes.

4.2.2. Routing in Heavily Non-uniform Networks We used a network area of size 1600×1600 and divided it into cells of size 100×100. In order to observe the effect of different sensor distributions, we experimented with three types of non-uniform networks:

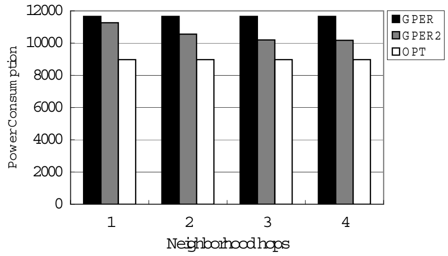
- `Random` networks in which cells in the network are labeled *dense* or *sparse* randomly (Figure 13(a). In these networks, 75% of the cells are sparse. High

---

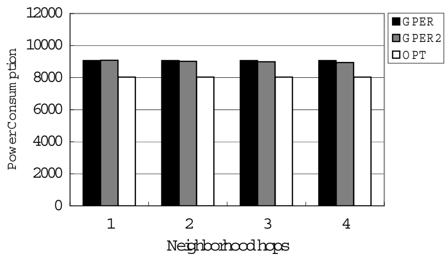
2 In our experiments, the radio range is selected in a way that the probability of dead ends, which would cause GPSR and GPER to fail, is very low



(a) Random



(b) Self-similar (fractal/power law)

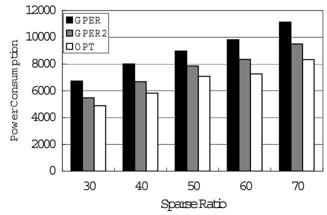


(c) Center-heavy

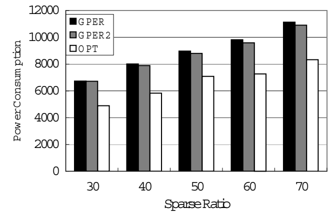
Figure 13. Power consumptions corresponding to (a) random, (b) fractal, and (c) center-heavy density distributions

density cells have 290 nodes and low density cells have 58 nodes. The total number of nodes is 30000. Nodes are uniformly distributed within each cell.

- `Self-similar` or fractal (power law) networks in which cells in the network are distributed in a way that the distribution of the cells at higher scales resemble distribution of the cells in lower scales (Figure 13(b). In these networks too, 75% of the cells are sparse. High density cells have 290 nodes and low density cells have 58 nodes; i.e., the total number of nodes is 30000. The nodes are uniformly distributed within each cell.
- `Center-heavy` networks in which the density of the cells is heavier in the center than in the borders (Figure 13(c). The 30000 nodes are distributed such that density is gradually decreasing: cells that are 7-8 hops away from center have 62 nodes, cells that area 5-6 hops away have 2×62 $(= 124)$ nodes, cells that are 3-4 hops away have 3×62 $(= 186)$ nodes, and cells that are 1-2 hops away have 4×62 $(= 248)$ nodes. The nodes are uniformly distributed within each cell.



(a) Uniform cells



(b)Non-uniform cells

Figure 14. Effects of the sparsity of the grid overlay and non-uniformity of the cells

Furthermore, in order to observe the impact of the neighborhood size, we have experimented with four different neighborhood sizes: 1-, 2-, 3-, and 4-neighborhoods. In the experiments, we calculated the average power consumption for routing between all pairs of center nodes in the network.

Effect of the Network Type: As shown in Figures 13(a) and (b) when non-uniformity is distributed randomly or in a self-similar fashion, GPER-2 can improve the results of GPER upto 20% as there are opportunities to avoid sparse regions. On the other hand, when the non-uniformity is center-heavy and gradually changing, GPER already functions almost as good as GPER-2 (Figure 13(c)). This shows that GPER is sensitive to abrupt and large changes in the node density (in random and fractal distributions), whereas GPER-2 functions well in all cases.

Effect of Neighborhood Size: Figures 13 (a), (b) and (c) show that as the neighborhood size increases, the power consumed by GPER-2 reduces because more information about the neighboring cells are available to the decision maker node[3]. As the neighborhood size goes from 3 to 4, the additional reduction is not large, which means that, in this setup, more information regarding the rest of the network is not necessary.

Effect of Sparsity: In these experiments, presented in Figure 14, we varied the ratio of the sparse cells in the network. In this experiment setup, each high density cell has 300 nodes and each low density cell has 60 nodes; the total number of nodes depend on the *sparsity ratio*. We ran five sets of experiments: for sparsity ratios of $30\%, 40\%, 50\%, 60\%$, and 70%. In each experiment, we averaged the power consumption for all pairs of nodes in the network. Neighborhood size is fixed to 3 for GPER-2. The power loss constant and radio range is set to 2.0 and 30 respectively.

3 The information needed for this is very small and local and can easily be stored at each node

Figure 14(a) shows the comparison of average power consumption. We can see that as the network becomes sparser, the total power consumption increases as expected. In all cases, GPER-2 further reduces the power consumption of GPER upto 15-20%; though both are close to the optimal.

Effect of Cell Uniformity on GPER-2: GPER-2 algorithm assumes that each cell in the overlay has close to uniform density. In order to observe the impact of cell uniformity on GPER-2, we shifted each cell 50 units (half of a cell) in both directions such that edge borders no longer cleanly separate high and low density areas. Consequently, each cell may have significant non-uniformity in itself. Figure 14(b) shows the results. As expected, in this case, the savings that GPER-2 provides over GPER are very small (almost negligible) but positive; i.e., even in this case, GPER-2 does not introduce significant errors in the selection of the routes.

5. Conclusions and Future Work

In this paper, we presented a new *Geographical Power Efficient Routing (GPER)* protocol. The protocol is highly power efficient, distributed, and scalable. The simulation results showed that the routing power consumption using GPER is close to the optimal identifiable using full knowledge of the network. The results also showed that, GPER works well in networks with node density variations; however, there is room for further improvement. For sensor networks with varying node densities, we introduced an extension called GPER-2, which captures the diverse network topology through an overlay. The results show that GPER-2 further improves GPER results when there are large variations in the network density.

References

[1] S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM*, 2003.

[2] J. H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM*, pages 22–31, 2000.

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *MobiCom*, pages 85–96, 2001.

[4] X. Cheng, B. Narahari, R. Simha, M. Cheng, and D. Liu. Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics. *Mobile Computing*, 2(3):248–256, 2003.

[5] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181, 1996.

[6] H. Karl. Relaying in wireless access networks - a capacity and energy-efficiency perspective. In *Wireless World Research Forum Kickoff Meeting*, 2001.

[7] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom*, pages 243–254, 2000.

[8] Y. B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *MobiCom*, pages 66–75, 1998.

[9] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In *PODC*, pages 63–72, 2003.

[10] L. Li and J. Halpern. Minimum energy mobile wireless networks revisited. In *ICC*, 2001.

[11] Q. Li, J. Aslam, and D. Rus. Distributed energy-conserving routing protocols for sensor networks. In *IEEE HICSS*, 2002.

[12] R. Min and A. Chandrakasan. Mobicom poster: top five myths about the energy consumption of wireless communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(1):65–67, 2003.

[13] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[14] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM*, pages 1405–1413, 1997.

[15] C. Perkins. Ad-hoc on-demand distance vector routing. In *MILCOM '97 panel on Ad Hoc Networks*, 1997.

[16] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM*, 1994.

[17] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location informaition. In *MobiCom*, pages 96–108, 2003.

[18] T. S. Rappaport. Wireless communications: Principles and practice. In *Prentice Hall*, 1996.

[19] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. In *ICC*, pages 1633–1639, 1998.

[20] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *MobiCom*, pages 181–190, 1998.

[21] A. Srinivas and E. Modiano. Minimum energy disjoint path routing in wireless ad-hoc networks. In *MobiCom*, pages 122–133, 2003.

[22] J. Sucec and I. Marsic. Clustering overhead for hierarchical routing in mobile ad hoc networks. In *INFOCOM*, pages 1698–1706, 2002.

[23] Y. Xue and B. Li. A location-aided power-aware routing protocol in mobile ad hoc networks. In *IEEE Globecom*, pages 25–29, 2001.

[24] S. Zhao, K. Tepe, I. Seskar, and D. Raychaudhuri. Routing protocols for self-organizing hierarchical ad hoc wireless networks. In *IEEE Sarnoff Symposium*, 2003.

[25] M. Zorzi and P. R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance. *IEEE Trans. on Mobile Computing*, 2:337–348, 2003.