Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks

Yan Yu, Ramesh Govindan, Deborah Estrin
Email: yanyu/destrin@cs.ucla.edu, govindan@ISI.EDU

08/14/2001

Abstract

Future sensor networks will be composed of a large number of densely deployed sensors/actuators. A key feature of such networks is that their nodes are untethered and unattended. Consequently, energy efficiency is an important design consideration for these networks. Motivated by the fact that sensor network queries may often be geographical, we design and evaluate an energy efficient routing algorithm that propagates a query to the appropriate geographical region, without flooding. The proposed Geographic and Energy Aware Routing (GEAR) algorithm uses energy aware neighbor selection to route a packet towards the target region and Recursive Geographic Forwarding or Restricted Flooding algorithm to disseminate the packet inside the destination region.

We evaluate the GEAR protocol using simulation. We find that, especially for non-uniform traffic distribution, GEAR exhibits noticeably longer network lifetime than non-energy-aware geographic routing algorithms.

1 Introduction

Future sensor networks will be composed of a large number of densely deployed sensor nodes. Each node in the sensor network may consist of one or more sensors, a low power radio, portable power supply, and possibly localization hardware, such as a GPS (Global Positioning System) unit or a ranging device. A key feature of such networks is that their nodes are untethered and unattended. Consequently, they have limited and non-replenishable energy resources. Therefore, energy efficiency is an important design consideration for these networks.

In this paper we study energy efficient geographic packet forwarding techniques. Disseminating information to a geographic region is a very useful primitive in many location-aware systems, and especially sensor networks. For example, a sensor net application may be interested in "what is the average temperature in a region R in the time period (t_1, t_2)" [10]. The region can be expressed, for example, by a rectangle in 2-space. In order to fulfill the above communication task, this query needs to be disseminated to the sensors in the specified region.

An efficient way to disseminate the geographic query to a specified region is to leverage the location knowledge in the query and to route the query directly to the region instead of flooding it everywhere. Previous research has studied how to geographically route a packet to a target location in an ad-hoc network [12]. However, our work compares with previous geographic routing as follows:

1. Unlike unicast communication in previous systems, we study the problem of forwarding a packet to all the nodes inside a target region, which is a common primitive in data-centric sensor net applications [10].
2. Furthermore, our work does not assume the need for a *location database* that maps node identifier to node location. We expect sensor networks to be *data-centric*, where communication primitives are expressed not in terms of node identifiers but in terms of named data.
3. Our work also assumes static (*i.e.*, immobile) sensors. This does not simplify the geographic routing problem, but does enable some of the route learning techniques we use.
4. Like previous work, however, we do assume the existence of a localization system [22, 2, 18, 7, 19] that enables each node to know its current position.
5. Motivated by the stringent energy constraint in sensor networks, we use energy aware metrics, together with geographical information, to make energy efficient routing decisions. In previous work, balancing energy usage has not been an explicit design goal.

Our Geographic and Energy Aware Routing (GEAR) technique uses energy aware and geographically informed neighbor selection heuristics to route a packet towards the target region. Within a region, it uses a recursive geographic forwarding technique to disseminate the packet. Although the energy balancing design of GEAR is motivated by sensor net applications, our protocol is generally applicable to ad-hoc networks.

We simulated GEAR for uniform and non-uniform traffic distributions, and compared its performance to GPSR [12], which is a non energy-aware geographic routing algorithm. For non-uniform traffic, GEAR delivers 70% to 80% more packets than GPSR. For uniform traffic, GEAR successfully delivers between 25% and 35% more packets than GPSR.

However, in both cases, GEAR performs significantly better in terms of *connectivity after partition*—the fraction of pairs remaining connected after a "partition" (when all sources are partitioned from their respective target regions). We are currently implementing a prototype of GEAR protocol in a moderate size testbed.

The rest of the paper is organized as follows. We briefly discuss related work in Section 2. We explain GEAR in detail in Section 3, and present our simulation results in Section 4. We discuss some design details and describe the on-going implementation work in Section 5. Section 6 presents our conclusions.

2 Related work

2.1 Geographic ad-hoc routing

Most previous geographic routing protocols use greedy algorithms to forward the packet to the destination. They differ in how they handle communication holes.

Finn [4] is the earliest work known in geographical routing. He used restricted flooding search to navigate around holes. One drawback of this mechanism is the difficulty in determining an appropriate scope for the search. GPSR, by Karp et al [12], elegantly avoids this problem by deriving a planar graph out of the original network graph. In GPSR, the packet follows the perimeter of the planar graph to circumvent holes. The derived planar graph is much sparser than the original one, and the traffic concentrates on the perimeter of the planar graph in perimeter mode. Thus, the nodes on the planar graph tend to be depleted quickly. In addition, nodes are assumed to operate in promiscuous listening mode and consequently consume energy [21].

Scalable Location Update-based Routing Protocol(SLURP) [24] constantly maintains approximate location information of nodes in the network, and finds accurate routes to specific nodes on demand. It uses approximate geographic routing to route a packet to the region that contains the destination, and once the packet is inside that region, it uses source routing to reach the destination. It relies on route request to circumvent holes. The route request/ reply overhead and constant snooping mode in SLURP make it unsuitable for sensor net applications.

Less directly relevant is the work of Imielinski and Goel [9], who propose querying and monitoring DataSpace. One primitive in this application is to send a query to a datacube, which is analogous to forwarding a query to a certain region in this paper. They first use geographic routing to forward the query to the geonode responsible for the datacube specified in the query. Then the corresponding geonode multicasts the query to relevant nodes based on its prebuilt indexing structure. However, it is difficult to build and maintain an efficient index structure under the high level of dynamics that sensor networks are exposed to.

Ko et al. propose Location Aided Routing (LAR) [13], which limits the search for a new route to an estimated "request zone". The geographic location information is not used to make routing decision, but to limit the route request flooding to a smaller region. The request zone is estimated based on the destination's previous known location and its known mobility pattern. However, when mobility information is not accurate, the request zone may have to be enlarged to the whole network.

Li et al [15] propose a scalable and distributed location database service, which tracks mobile nodes' locations. It selects multiple location servers to store each node's location. Queries for a mobile node's location are resolved using the predefined identifier ordering and spatial hierarchy to find a location server for that node. As mentioned before, such a location database service is not necessary in our target application.

2.2 Other related work

Ad-hoc routing Dynamic Source Routing (DSR) [11] floods route request on-demand, and suggests using promiscuous listening to aggressively learn new routes. Sensor networks have more stringent energy constraint than do ad-hoc networks, and bandwidth requirements will be lower in sensor networks. Therefore, flooding route request or operating in promiscuous mode make DSR undesirable for sensor networks. Other existing Ad-hoc protocols [17, 16] also require greater energy resources and higher bandwidth demand than is anticipated for sensor networks. However, some of our techniques might be applicable in energy efficient versions of these protocols.

Energy aware routing Rather than using traditional metrics such as hop-count or delay for finding routes, Woo et al. [23] proposes five energy aware metrics such as "maximize time to partition" and "minimize maximum node cost". These are important metrics for energy efficient routing, however, it is difficult to directly implement them in a local algorithm when even the global version of the same problem is NP-complete.

Chang et al. [3] proposed a class of flow augmentation algorithms and a flow redirection algorithm which balance the energy consumption rates among the nodes in proportion to the energy reserves. The limitation of this approach is that it requires the prior knowledge of the sets of origin and destination nodes and the information generation rates at the origin nodes, consequently, the topology and the traffic are fixed at least between consecutive computations.

LEACH [8] proposes a clustering based protocol that utilizes randomized rotation of local cluster heads to evenly distribute the energy load among the sensors in the network. It is similar in spirit with other energy aware routing protocols in terms of load balancing. However, their underlying assumption is different from ours. They assume adjustable transmitting power and assume that the cluster head talks directly to the gateway node. We assume each node has fixed transmission power – optimizing transmission power in a multi-hop network is outside the scope of this paper.

PAMAS [20] proposes a new power aware multi-access protocol for ad-hoc radio networks. It conserves battery power at nodes by intelligently powering off nodes that are not actively transmitting or receiving packets. Their energy efficient MAC design complements our work.

Sensor network routing mechanism Directed diffusion [10] is a data-centric protocol for sensor network applications. It achieves some level of energy savings by selecting empirically good paths, and by caching and processing data in-network. However, without proposed geographic routing support, there is initial and periodic interest and low rate data flooding throughout the network. GEAR protocol can compliment this work by efficiently route interest to the destination region, thus conserve more energy.

Gao and Pottie [6, 5] proposes a table driven, multi-path network structure for the communication between a large number of sensors and a central information gathering entity called the USER. However, their pre-built routing table and multi-path structure may not scale to large size sensor networks. Moreover, table driven approach may not be able to adapt well to network dynamics or traffic dynamics at a low cost.

Localization work There has been substantial research interest [22, 18, 2, 7, 19] in localization systems. Such systems are a prerequisite for geographical routing and other sensor net applications. Of those, Ward et al. [22] propose a ultrasonic location system based on tri-lateration principle; Bulusu et al. [2] propose a coarse-grained connectivity metric method for localization in outdoor environments in the absence of GPS; Girod and Estrin [7] propose a robust range estimation technique using acoustic and multimodal sensing; Savvides et al. [19] propose Ad-Hoc Localization system (AHLos), a find-grained localization technique for ad-hoc sensor networks.

3 Geographical and Energy Aware Routing (GEAR)

We now describe the Geographical and Energy Aware Routing(GEAR) algorithm. As mentioned in the introduction, we are interested in routing queries to regions in proposed sensor-net applications. The process of forwarding a packet to all the nodes in the target region consists of two phases:

1. Forwarding the packets towards the target region: GEAR uses a geographical and energy aware neighbor selection heuristic to route the packet towards the target region. There are two cases to consider:

   (a) When a closer neighbor to the destination exists: GEAR picks a next-hop node among all neighbors that are closer to the destination.

   (b) When all neighbors are further away: In this case, there is a hole. GEAR picks a next-hop node that minimizes some cost value of this neighbor, which we will discuss in detail in section 3.1.2.

2. Disseminating the packet within the region: Under most conditions, we use a Recursive Geographic Forwarding algorithm to disseminate the packet within the region. However, under some low density conditions, recursive geographic forwarding sometimes does not terminate, routing uselessly around an empty target region before the packet's hop-count exceeds some bound. In these cases, we propose to use restricted flooding.

Before we describe the above algorithms in detail, we state the assumptions of this work:

1. each query packet has a *target region* specified in some way (for the description of the algorithm, we assume a rectangular region specification).

2. each node knows its own location and remaining energy level, and its neighbors' locations and remaining energy levels through a simple neighbor hello protocol. Note that a node can obtain its location information at low cost from GPS or some localization system [22, 18, 2, 7, 19], which presumably is already available due to the needs of sensor net applications.

3. the link is bi-directional, i.e., if a node hears from a neighbor N_i, then its transmission range can reach N_i. This is not an unreasonable choice as most MAC layer protocols, such as IEEE 802.11, assume symmetric links.

3.1 Energy-Aware Neighbor Computation

In this section, we assume that the node N is forwarding packet P, whose target region is R. The centroid of the target region is D. Upon receiving a packet P, the node N routes P progressively towards the target region, and at the same time tries to balance the energy consumption across all its neighbors. Node N achieves this trade-off by minimizing the *learned cost* $h(N_i, R)$ value of its neighbor N_i.

Each node N maintains state $h(N, R)$ which we call its *learned cost* to region R. A node infrequently updates its $h(N, R)$ value to its neighbors. We discuss this infrequent update later. We implicitly define $h(N, R)$ in the next two paragraphs.

If a node does not have $h(N_i, R)$ state for a neighbor N_i, it computes the *estimated cost* $c(N_i, R)$ as a default value for $h(N_i, R)$. The *estimated cost* $c(N_i, R)$ of N_i is defined as follows:

$$c(N_i, R) \;=\; \alpha d(N_i, R) + (1-\alpha)e(N_i) \qquad (1)$$

where α is a tunable weight, $d(N_i, R)$ is the distance from N_i to the centroid D of region R normalized by the largest such distance among all neighbors of N, and $e(N_i)$ is the consumed energy at node N_i normalized by the largest consumed energy among neighbors of N.

After a node picks a next-hop neighbor N_{min} (we describe this neighbor selection in section 3.1.1 and section 3.1.2), it sets its own $h(N, R)$ to $h(N_{min}, R) + C(N, N_{min})$ where the latter term is the cost of transmitting a packet from

N to N_{min}. $C(N, N_{min})$ can also be a combination function of both the remaining energy levels of N, N_{min} and the distance between these two neighbors.

The intuition for minimizing the estimated cost function $c(N, R)$ is as follows:

- When all nodes have equal energy, this degenerates to the classical greedy geographic forwarding: forwarding the packet to the nearest neighbor to destination.

- When all neighbors are equidistant, this degenerates to load splitting among neighbors. Note that minimizing the energy cost among neighbors is a local approximation to the lowest energy cost path, if it is more expensive to use a node that has less remaining energy. Since GEAR makes forwarding decision only based on local knowledge, an approximation to the global lowest cost path is the best that a local algorithm can achieve.

Now that a node has a learned cost state $h(N, R)$ or a default estimated cost function $c(N, R)$ for each neighbor, we now describe the forwarding actions at node N. As with other geographical routing schemes, there are two cases to consider:

- If at least one neighbor of N is closer to D than N;

- All neighbors are further away from D than N.

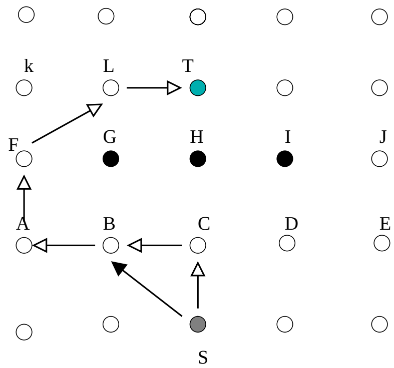

Figure 1: Learning routes around holes

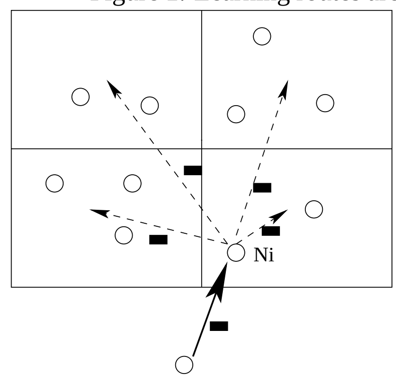

Figure 2: Recursive Geographic Forwarding

3.1.1 Closer Neighbor Exists

As mentioned before, under GEAR, the packet contains a target region field. Therefore, a forwarding node can make locally greedy choice in selecting next-hop node. Whenever a node N receives a packet, it will pick the next hop among the neighbors that are closer to the destination, at the same time, minimizing the learned cost value $h(N_i, R)$. Since it picks a next-hop node from closer neighbors, it will route progressively towards the target region when there are no holes. Without holes, the learned cost is a combination of consumed energy and distance, minimizing the learned cost value is a trade-off between routing towards the next-hop closest to the destination and balancing energy usage.

3.1.2 All Neighbors are Farther Away

In this case, N knows it is in a *hole*. A node's learned cost $h(N, R)$ and its update rule are combined to circumvent holes. Intuitively, when there is no hole in the path towards R, the node's *learned cost* $h(N_i, R)$ is equivalent to the *estimated cost* $c(N_i, R)$. However, when there is a hole in the path towards R, the node's *learned cost* represents "resistence" to following the path towards that hole; "resistence" that the *estimated cost* cannot provide. Next, we illustrate this latter feature of the learned cost.

Figure1 is a grid topology. Suppose the distance between the nearest two neighbors is *1*, and each node can reach its 8 neighbors. The nodes in black, i.e., G, H, I, are energy depleted nodes, thereby can not relay packets. Suppose node S wants to send a packet to region R with centroid at T. For illustration purposes, we use T to denote this region. Again, for simplicity, we illustrate the algorithm using pure geographic routing, i.e., we set α in Equation 1 to be 1, and we use direct distance instead of *normalized distance* mentioned earlier.

Initially, at time 0, at node S, among all neighbors of S, B, C, D are closer to T than S. $h(B, T) = c(B, T) = \sqrt{5}$, $h(C, T) = c(C, T) = 2$, $h(D, T) = c(D, T) = \sqrt{5}$.

Upon receiving a packet destined to T, S will forward it to its lowest cost neighbor, i.e., C. At C, it will find itself in a hole, since all C's neighbors are further away from T than itself. At C, it forwards the packet to the node with the minimum $h(N, T)$. When there are ties, it breaks ties based on some predefined ordering (e.g., node ID). For example, it picks B as the next hop, then update its own $h(C, T) = h(B, T) + C(C, B)$, where $h(B, T) = \sqrt{5}$, and suppose one hop transmission cost from a node to its neighbor is 1, i.e., C(C, B) = 1.

Later on, for example, at time 2, node S receives a packet destined to the same region T, the h values of its neighbors become:

$$h(B, T) = \sqrt{5},\ h(C, T) = \sqrt{5} + 1,\ h(D, T) = \sqrt{5}.$$

Thus, this time node S will forward the packet directly to B instead of C to circumvent the hole. The actual forwarding action of the algorithm at node S will oscillate between B and C several rounds before the learned cost converges to favoring B as a next hop neighbor. After the first packet reaches the destination, the correct learned cost value will be propagated one-hop back. Every time a packet is delivered, the correct learned cost value will be propagated one-hop away. Therefore, suppose the path length from S to T is n, the learned cost will converge after the node delivers n packets to the same target T. Note that the convergence of learned cost does not affect successfully routing a packet out of holes, it only affects how efficient is the hole routing path. Propagating the learned cost values further upstream through the update rule will enable the packet to have an earlier chance to avoid holes (i.e., more effectively circumnavigate holes), and at the same time avoid depleting the nodes surrounding the holes.

In summary, the learned cost together with its update rule help to learn the route around holes. Intuitively, the learned cost is set to the current best choice available.

This learned cost is inspired by the Learning Real Time A* algorithm [14], which is a well known heuristic search technique. In [14], Korf has proved the completeness of the LRTA*, i.e., if there is a path, LRTA* will find it, and there will not be infinite loop. Since our learned cost based hole routing algorithm is very similar to LRTA*, we conjecture that this result applies to our scheme as well.

3.1.3 Discussion

The estimated cost $c(N, R)$ is a combination of the normalized distance from a neighbor to the destination and its normalized remaining energy level. In equation 1, α can be adjusted to emphasize minimizing path length to the destination or balancing energy consumption. We tried several variants of this estimated cost function. For example, we tried different energy cost functions and different normalization denominators. The simulation results show that the algorithm performance is not very sensitive to the particular estimated cost function. Our explanation is that it is the comparison (relative value among all the neighbors), not the absolute estimated cost value that matters, since it is used to make a local selection among all the neighbors.

For computing $c(N, R)$, each node needs to know neighbors' energy levels and locations. A node also needs neighbors' learned cost to make forwarding decisions. Various techniques are possible: e.g., piggybacking these on data traffic, requesting this information on demand, advertising the information only when its value changes significantly, or a combination of the above. In our simulation, we implement a threshold advertising scheme.

We have verified through simulation (section 4.4) that the threshold can be set in a manner that the overhead is insignificant without adversely affecting performance.

3.2 Recursive Geographic Forwarding

Before a packet reaches the target region R, we use the forwarding rules described in the previous section. Once the packet is inside the target region, a simple flooding with duplicate suppression scheme can be used to flood the packet inside region R. However, flooding is expensive in terms of energy consumption, due to the fact that in this simple flooding scheme, every node has to broadcast once, and all its neighbors receive this broadcast message. This is especially expensive in high-density networks, which is the case for some proposed sensor net applications where nodes are densely, and redundantly deployed for robustness. This demonstrates the necessity to use an energy efficient routing algorithm in place of flooding in disseminating the packet inside the target region, or any kind of flooding in sensor networks.

Therefore, we use a Recursive Geographic Forwarding approach to disseminate the packet inside target region R. As shown in Figure2, suppose the target region R is the big rectangle in Figure2, and now node N_i receives a packet P for region R, and finds itself inside R. In this case, N_i creates four new copies of P bound to 4 sub-regions(as shown by 4 small rectangles in Figure2) of region R. Repeat this recursive splitting and forwarding procedure until the stop condition for recursive splitting and forwarding are satisfied.

The recursive splitting terminates if the current node is the only one inside this sub-region. The criteria to determine this is when the farthest point of the region is within a node's transmission range, but none of its neighbors are inside the region. When no node is inside the sub-region, the packet is dropped altogether. The rule to determine a sub-region being empty is the same as the previous criteria.

We choose to divide region R into 4 subregions for convenience. Other ways of recursively visiting subregions are also possible, though we do not consider them in this paper.

3.2.1 Pathologies

When network density is low recursive geographic forwarding is subject to two pathologies: inefficient transmissions and non-termination.

Inefficient transmissions In recursive geographic forwarding, in its recursive splitting process, to reach 4 sub-regions, a unicast packet is sent to its neighbors multiple times, and it is received only by the intended receivers. In contrast, restricted flooding exploits the broadcast medium of the wireless channel, only sends one broadcast message to all its neighbors, but every node in its transmission range receives this broadcast message whether it is an intended receiver or not. Which approach is more energy efficient depends on the density of the target region. In low density scenarios, it is more energy efficient to use restricted flooding than recursive geographic forwarding.

For example, in Figure3, the rectangle is the target region R, there are 4 nodes, i.e., D, B, C, and E, in the target region, and the transmission range of a node is about the distance from D to B. Suppose the average number of neighbors of a node is 4. The first node that the packet has reached in region R, i.e., C, finds out that R is within its transmission range (i.e., all other nodes in R are its neighbors). Then the node can use one-hop flooding instead of sending 3 copies of unicast message separately to the other 3 nodes in region R as in recursive geographic forwarding case. Suppose sending or receiving a packet consumes 1 unit of energy, then the best that recursive geographic forwarding can achieve is to send 3

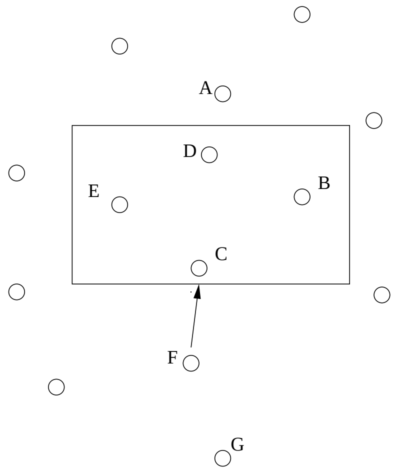

Figure 3: Recursive Geographic Forwarding vs. restricted flooding

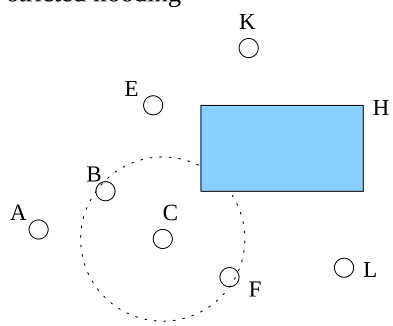

Figure 4: Non-termination at empty region in low density case

unicast messages to the other 3 nodes in R, thus consuming 6 units of energy in total. However, one-hop flooding consumes 1 unit energy for sending, and 4 units of energy for receiving at C's neighbors (i.e., B, D, E, F), thus 5 units of energy in total. This is just a simple example to show the difference between two schemes. Extrapolated to a larger region, the differences can be significant.

Non-termination In the recursive geographic forwarding protocol, packet forwarding terminates when *the target sub-region is empty*. The following heuristic is used to determine if *the target sub-region is empty*: *if the farthest point of the region is within its transmission range, but none of its neighbors are inside the region, the region is considered empty.*

However, this strategy does not work when the network density is low compared to the (sub)target region size. When density is low the probability of the target region being empty is high, and the transmission range is small compared to the target region size so that the node that is closest to the target region can not reach the other end of the target region. As a result, the nodes outside the target region have no indication that the region is empty, hence the packet still searches for routes to get into the empty region. The search will not stop until the number of hops traversed exceeds the packet's time-to-live. This failed search process can heavily drain the nodes around the target region.

For example, in Figure4, suppose the blue (gray) rectangle is one subregion of the target region R, namely R_s, toward which the packet is heading. Each node has equal transmission range, and the dotted circle indicates the transmission range of C. In this example, region R_s is empty, however, for each individual node bordering R_s, i.e., C, E, K, L, its transmission range is unable to cover the whole region R_s. For example, at C, its transmission range can not reach the far end of region R_s (i.e., H). Therefore, the packet can not tell that region R_s is empty. Hence, the packet will repeatedly traverse the nodes around region R_s until its hop count passes some bound.

We propose to use restricted flooding to deal with these pathologies in low density scenarios. Node degree is used as a criteria to differentiate low density scenarios from high density scenarios. Once the packet reaches the first node J in the target region, whether to use recursive geographic forwarding or restricted flooding will be based on the number of neighbors of node J. If this number is below a threshold, then the packet is flooded inside the region, otherwise recursive geographic forwarding will be triggered. Simulation results show that this approach helps to avoid the pathological case mentioned before, thereby improving the performance dramatically(more than an order of magnitude improvement) in low density scenarios, and exhibits the same performance as before in high density case, since restricted flooding is not triggered in high density scenarios. However, we will explore a better alternative solution to this problem as future work.

3.3 Adaptive energy aware behavior of our algorithm

We use an energy aware metric in the estimated cost function to balance energy consumption, and achieve energy efficiency for the whole network. However, under some circumstances, taking the geographically direct path is more energy efficient, and consequently prolongs network lifetime. Under the following conditions, the pure geographic mode is used instead of adding the energy aware metric.

1. After the number of hops traversed crosses some threshold, pure geographical routing is used instead of energy aware routing. The motivation behind this is to get to the target directly if the packet has already traveled a long way. The threshold is tunable, however, the algorithm is not very sensitive to this parameter which is confirmed by the simulation results. For example, it can be set to a large enough number.

2. After a packet reaches a node whose neighbors are heavily depleted (which indicates a neighborhood where nodes are heavily depleted), the packet will switch to pure geographic mode to avoid taking an alternative longer path, and consequently consuming more energy than the direct path.

3. When nodes are near the target region, pure geographic mode is used. Since the packet is supposed to be disseminated to all the nodes in the target region, the nodes bordering the target region are the bottleneck. Pure geographic mode is used near the target region to avoid taking a longer path and thereby avoid burning out more bottleneck nodes more quickly.

4 Simulation and comparison

Geographic routing is designed to improve efficiency relative to flooding data. Our primary performance measure is increased network lifetime due to energy awareness. Moreover, we are interested in how this comparative measure scales with network size. Sensor networks may involve thousands of nodes, thus scaling to large size network is essential for a routing protocol to be applicable to sensor networks.

We study and compare GEAR with and without energy awareness. In addition, we evaluate how our geographic routing algorithm compares with other geographic routing algorithm, such as GPSR where a pro-active and deterministic hole routing scheme is used. Since the original GPSR does not handle routing to a region, we have augmented GPSR with our recursive geographic forwarding protocol to route packets to a region. Idealized multicast and flooding are simulated for comparison. They represent two extremes, idealized multicast requires global information to compute shortest paths to every node in the target region, while flooding uses no information, but is far less efficient.

Testing the scalability of our protocol to *large* networks was our primary goal, and we achieved this through simulation. We are also completing a prototype implementation to validate our results on moderate size testbed.

Next, we describe and justify some major features of our simulator.

- We use a discrete event-driven simulator. It simulates routing packets among different nodes in the network. We are not interested in the processing details of each packet in the network per se, therefore we chose to leave out packet level details, such as transmission delay, or queuing delay etc. Leaving out irrelevant details enables us to simulate large size network.

- We use a simple energy model in which every node starts with the same initial energy budget, and transmitting or receiving a packet consumes one unit of energy. We do not count energy consumed by control packets. As we show in section 4.4, control overhead is relatively small in GEAR. Therefore, our comparison is fair considering that periodic updating in other protocols may consume much more energy than threshold triggered update in our approach.

- MAC assumptions: We assume a MAC layer in which each node consumes negligible energy when it is not sending or receiving packets. In a MAC layer which can not turn itself off when it is not actively transmitting or receiving, the energy consumed by listening or idle state will dominate the total energy consumption of a node, and make different routing protocols exhibit roughly the same energy consumption characteristics. As a result, sensor net applications demand a MAC layer where a node can put itself into sleep mode when it is not involved in active communication [20]. On the other hand, our routing protocol does not depend on any particular type of MAC protocol as long as it can resolve medium access control issue, therefore, we use a MAC layer that assumes no collision. Note that our goal is performance comparison between different protocols. *The effect of this idealization is common to all simulated protocols*, therefore, the relative comparison between different protocols should not change much under a different MAC layer, even though a MAC layer with more realistic details may change a routing protocol's performance.

We adjust the following parameters in the simulation: network size, density, size of target region, and traffic description.

In this paper, we focus on varying network size, keeping density and other parameters constant. Our goal is to design a protocol that can scale to thousands of nodes in future sensor networks, therefore we focus our study on how the protocol scales with networks of different sizes. However, we also conducted some scalability studies with other input parameters. The simulation results show that the GEAR algorithm also scales well along other input parameters.

4.1 Simulation Scenarios

We do not know how future sensor net traffic will be, thus we choose two simple models: uniform and non-uniform traffic.

Uniform Traffic Randomly distributed source and destination pairs. The source and target regions are randomly distributed throughout the network. This represents independently and uniformly distributed traffic.

Non-uniform Clustered traffic For clustered source and destination pairs, the sources and destinations are randomly selected, but the sources from different traffic pairs, or the destinations from different traffic pairs are geographically close to each other. This represents non-uniform traffic distribution, which is not uncommon in proposed sensor net applications where detected physical events or triggered communication are not completely independent from each other, and sometimes highly correlated.

As mentioned earlier, in this paper, we vary network size, fix density, transmission range, and other parameters. More specifically, the simulation results shown here include networks of size ranging from 400, to 4800 nodes. For a 600 nodes network, its geometric area is 1200×1200 square. We adjust network geometric area proportionally to the number of nodes in the network to keep network density fixed. We fix a node's transmission range to be 100 units across all simulations. The target region is a circle with radius 50 units.

10 traffic source and target pairs are randomly selected either from a uniform distribution or clustered together. A node's initial energy level is 1 joule, transmiting or receiving a packet consumes 0.001 joule. Each data point in the graphs presented in this paper is averaged over at least 100 simulation runs. For the results presented in this section, we set α in Equation 1 to 0.5.

4.2 Performance Metrics

It is difficult to precisely define metrics to study the impact of energy awareness on network lifetime. However, we have developed two metrics that approximately capture notions of network lifetime:

Packets before partition Number of data packets sent and successfully delivered before network partition. Network is considered partitioned if all the given sources are partitioned from their respective target regions. In some sense, this metric indicates the network lifetime.

Connectivity after partition Fraction of pairs still connected after partition. This metric indicates how the given traffic affects the rest of the network. Because we lack of benchmark traffic model, we concern the residual energy after the given traffic pairs are partitioned, and how much further communication it can support.

4.3 Simulation Results

Uniform Traffic With random/uniform traffic distribution, our algorithm successfully delivers 0.25 to 0.35 times more packets than GPSR. GEAR is also more energy efficient, which is measured in terms of the "connectivity after partition" metric mentioned earlier. In both uniform traffic distribution and non-uniform traffic distribution (which is shown in Figure 8), GEAR delivers significantly (40 to 100 times) more packets than flooding.

Figure 5 shows the mean and 95% confidence interval of the number of packets sent and successfully delivered before network partition. Note that some confidence interval is too tight to be visible in the graph. All later graphs show the mean and 95% confidence interval of the y-axis value, except Figure 9 and Figure10, where we omit the confidence interval to make the graph easier to read.

As shown in Figure 5, GEAR successfully delivers 25 - 35% more packets than GPSR. The energy aware version delivers comparable number of packets compared to its pure geographical variant. In complete random/ uniform traffic distribution, the energy consumption is already balanced due to randomly distributed traffic. Extra energy balancing efforts will not help much in terms of contributing to delivering more packets before network partition. Therefore, under uniform traffic, the energy aware version and the pure geographical version deliver similar number of packets before network partition. The reason that GEAR prolongs network lifetime compared to GPSR, is that GPSR tends to concentrate traffic on the perimeter when it routes around holes, thus burning out the nodes on the perimeter sooner. This is confirmed by the fact that under GPSR, after network partition, there are more nodes depleted along the direct paths

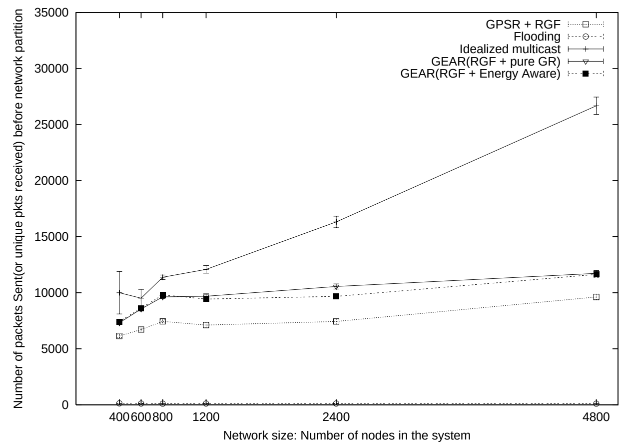

Figure 5: Comparison for uniform traffic

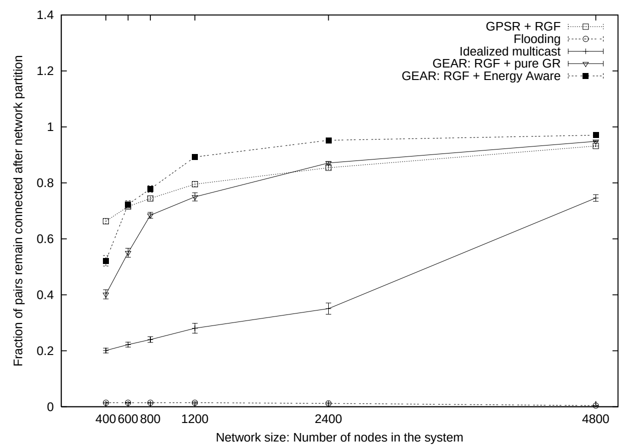

Figure 6: Fraction of pairs remain connected after partition for uniform traffic

between the source and the target compared to the GEAR case. As expected, the idealized multicast protocol delivers the largest number of packets before network partition in this uniform traffic case since we did not count energy consumption for control packets in the simulation. However, in reality, it is prohibitively expensive to collect the required global information for idealized multicast routing.

As shown in Figure 6, after the given traffic pairs are partitioned, there are more source and destination pairs remaining connected in GEAR than other protocols. This is an energy-efficient feature in the sense that after the given traffic pairs are partitioned, the rest of the network can still communicate. However, this metric alone can not represent the energy efficiency of a protocol, since an inefficient protocol may deliver far fewer packets than other protocols before network partition, thereby having lots of energy left in the network, and consequently having more pairs connected after partition. For example, as shown in Figure 5, GPSR delivers between 25% and 35% fewer packets before network partition than GEAR, but its curve is very close to GEAR's in Figure 6. To better differentiate between these two protocols in terms of their energy efficiency, we introduce the metric *resource expended per packet delivered*, and we count *resources expended* in terms of the number of connected pairs that are broken down because of nodes being depleted. Precisely, it is defined as:

$$\frac{N_b - N_e}{total\ number\ of\ delivered\ packets} \tag{2}$$

where N_b is total number of connected pairs in the beginning, N_e is the is total number of connected pairs after network partition. It can be considered as a normalized version of the metric presented in Figure 6.

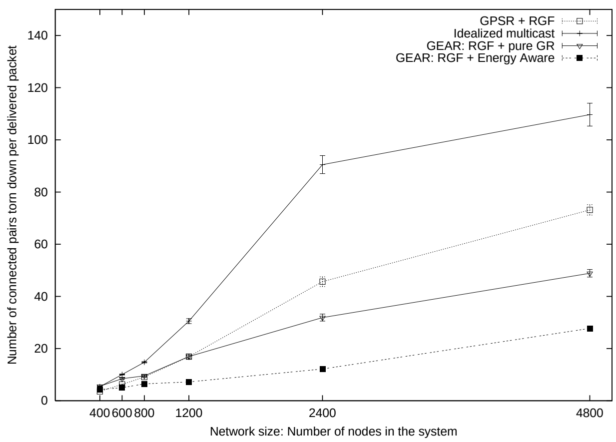

Figure 7: Number of broken down pairs per delivered data packet

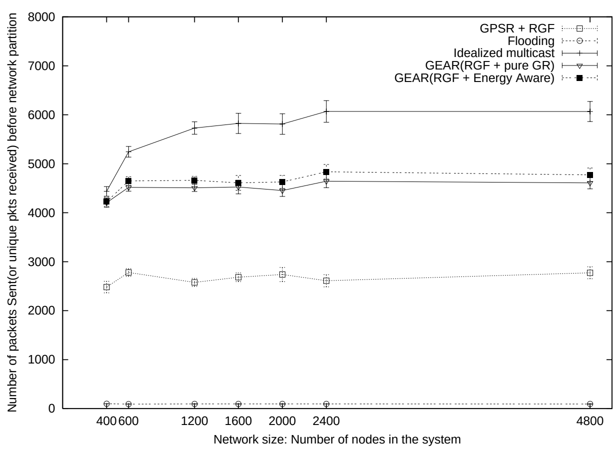

Figure 8: Comparison for non-uniform traffic

Figure 7 shows the number of pairs disconnected per delivered packet with different protocols. Note that the lower is the curve, the more energy efficient is the protocol. As shown in the graph, GEAR is much better than all other protocols, including the pure geographical variant of the same algorithm. Specifically, GEAR is approximately 50% more efficient than its pure geographic variant, approximately 70% more efficient than GPSR, and approximately 80% more efficient than idealized multicast. The value for Flooding is too large to be shown in this graph. This may suggest that the GEAR protocol successfully achieves energy efficiency. Note that the metric in Figure 7 magnifies the difference between different protocols in terms of energy efficiency compared to the metric in Figure 6.

Non-uniform Traffic Clustered source and destination pairs represent non-uniform traffic distribution in sensor net applications.

Figure 8 shows that GEAR delivers 70 - 80% more packets than GPSR and 4 - 15% fewer packets than idealized multicast. Hence, GEAR exhibits more gain in non-uniform traffic scenarios than uniform traffic scenarios. Our explanation is that when traffic sources are clustered together, GEAR's energy balancing efforts pay off most.

Under non-uniform energy distribution(i.e., the nodes in the target region have higher initial energy level than others), the gains for GEAR are even much higher. Simulation results show that GEAR delivers 125% more packets than GPSR and 25% more packets than idealized multicast. That GEAR performs better than idealized multicast seems surprising. However, our energy aware routing scheme balances energy consumption, while the idealized multicast keeps using the shortest path, which in turn will burn out the nodes along those paths quickly. Note that non-uniform energy distribution may map to real world sensor networks where some nodes that have sensing capability may have different energy resource than others.

4.4 Trade-off between neighbor information update and performance

The results shown in this section are for 600 nodes network, traffic are uniformly distributed, a node's initial energy level is 1 joule, transmitting or receiving a packet consumes 0.001 joule. As mentioned before, neighbor information update can be triggered by a pre-defined threshold. As shown in Figure 9 and Figure 10, it is not necessary to update neighbor information for every packet. With increasing update threshold, the protocol performance degrades gracefully, but the number of control packets generated drops dramatically. For example, in Figure 10, for the point with *learned cost h-value update threshold = 5*, and *remaining energy update threshold = 0.05*, the average number of control packets sent per node for the whole simulation run is only 32 packets, i.e., 3.2% of a node's total energy budget. However, as shown in Figrure 9, the performance for this particular case only degrades approximately 6% compared to the no update delay case, which corresponds to the black square in Figrure 9.

5 Discussion and Future work

Sensitivity to location error: We also studied how sensitive the protocol is to the location error. In the real world, location error can be caused by either imprecise measurement from GPS or localization system, or nodes moving, but failing to update location information timely. To investigate the sensitivity of our scheme to location error, we introduce a random location error for every node in the network. Simulation results show that with moderate location error, the

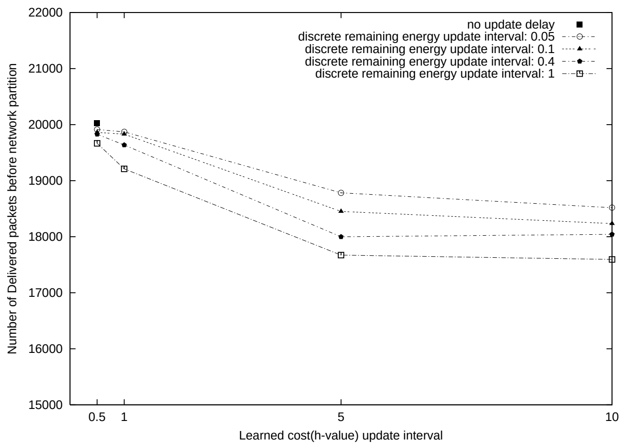

Figure 9: Number of delivered packets with different neighbor information update threshold

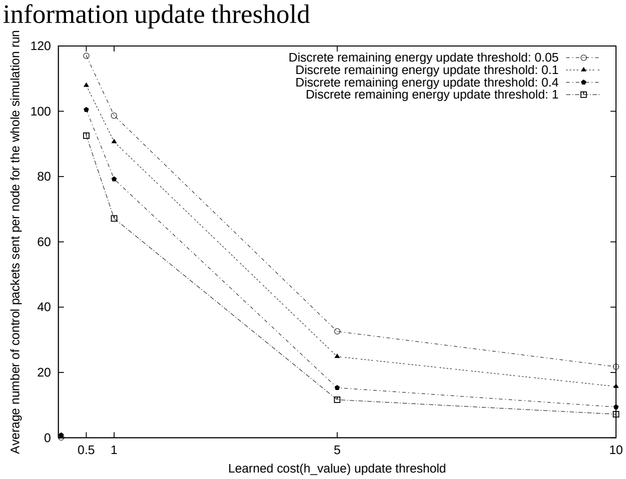

Figure 10: Number of update control packets with different neighbor information update threshold

protocol still achieves satisfying performance. For example, for 1200 x 1200 square, and 600 nodes network, each node's transmission range is 100 units, when we introduce random error in the range *[-6, 6]* for each node's x- and y- coordinates, the simulation results show no statistical performance degradation. If we introduce randomized error *[-60, 60]* for each node's x- and y- coordinates, the simulation results show only 9% performance degradation.

Impact on average path length: GEAR achieves energy balancing by taking alternative path, therefore, it is not surprising that our energy balancing strategy increases path length by 25% to 45% over all packets delivered. However, GEAR delivers more packets than GPSR. When we check the initial same number of delivered packets as in GPSR, the average path length for those packets in GEAR is only 10% longer than GPSR. After some nodes along the path are depleted, the packet has to take alternative longer path to avoid holes, thus the average path length for later communication is expected to be longer than those earlier on.

Implementation and Porting to ns-2 We have already implemented a pure geographic variant of GEAR protocol using our data dissemination diffusion testbed [10]. As a next step, we plan to implement the full-fledged version of GEAR. We plan to investigate how the details that our simulator abstracted out affect the protocol performance, specially how the real world MAC affects the protocol performance in the implementation. We also plan to study how the GEAR algorithm interacts with the directed diffusion implementation.

We also plan to port the GEAR algorithm to ns-2 [1], so that people can study how geographical and energy aware routing interacts with other protocols, such as directed diffusion in sensor networks, with the presence of a more detailed lower layer but still in a controlled simulation environment.

6 Conclusion

Motivated by future sensor network applications, we studied the problem of forwarding a packet to nodes in a geographic region of an ad-hoc wireless sensor network. The proposed Geographic and Energy Aware Routing (GEAR) protocol uses energy aware and geographically informed neighbor selection to route a packet towards the target region. This strategy attempts to balance energy consumption and thereby increase network lifetime. Within a region, it uses a recursive geographic forwarding technique to disseminate the packet. The simulation results show that for an uneven traffic distributions, GEAR delivers 70% to 80% more packets than GPSR. For uniform traffic pairs, GEAR delivers 25 - 35% more packets than GPSR. Moreover, in both cases, GEAR performs better in terms of *connectivity after initial partition*. We are currently implementing a prototype of the GEAR protocol in a moderate size testbed. We plan to investigate how the details of a real implementation affect the protocol performance.

References

[1] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Kannan Varadhan, Ya Xu, Haobo Yu, and Daniel Zappala. Improving simulation for network research. Technical Report 99-702b, University of Southern California, March 1999. revised September 1999, to appear in IEEE Computer.

[2] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[3] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. *INFOCOM*, 2000.

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

[5] J. Gao. Energy efficient routing for wireless sensor networks. *Ph.D. Dissertation, UCLA*, 2000.

[6] J. Gao, K. Sohrabi, V. Ailawadhi, and G. Pottie. An energy conscious self-organizing wireless sensor network. *IEEE PCI Magazine*, October 2000.

[7] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Maui, Hawaii, October 2001.

[8] W. Rabiner Heinzelman, A. Chandrakasan, and Hari Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proceedings of the 33rd International Conference on System Sciences (HICSS '00)*, 2000.

[9] Tomasz Imielinski and Samir Goel. Dataspace - querying and monitoring deeply networked collections in physical space. *IEEE Personal Communications Magazine*, August 2000.

[10] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. ACM Mobicom*, Boston MA, 2000.

[11] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad-hoc Wireless Networks. In T. Imielinksi and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[12] Brad Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proc. ACM Mobicom*, Boston, MA, 2000.

[13] Yong-Bae Ko and Nitin H. Vaidya. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'98)*, Dallas, TX, 1998.

[14] Richard Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2-3):189–211, March 1990.

[15] 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. ACM Mobicom*, Boston, MA, 2000.

[16] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proceedings of INFOCOM 97*, pages 1405–1413, April 1997.

[17] Charles Perkins. Ad-Hoc On Demand Distance Vector Routing (AODV). Internet-Draft, November 1997. draft-ietf-manet-aodv-00.txt.

[18] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proc. ACM Mobicom*, Boston, MA, 2000.

[19] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in adhoc networks of sensors. In *Proc. ACM Mobicom*, 2001.

[20] S. Singh and C.S. Raghavendra. Pamas: Power aware multi-access protocol with signalling for ad hoc networks. *ACM CCR*, July 1998.

[21] Mark Stemm and Randy H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE (Institute of Electronics, Information and Communication Engineers) Transactions on Communications, Special Issue on Mobile Computing*.

[22] Andy Ward, Alan Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5), October 1997.

[23] Mike Woo, Suresh Singh, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. *ACM/IEEE MOBICOM*, 1998.

[24] Seung-Chul Woo and Suresh Singh. Scalable routing protocol for ad hoc networks. *accepted for publication in Journal of Wireless Networks (WINET)*.