# On The Scalability of Sensor Network Routing and Compression Algorithms

John D. Owens

**Abstract**

Recent advances in sensor networks have developed routing algorithms and compression and aggregation schemes that allow these networks to use their limited resources, particularly power, most ef?ciently. As sensor networks mature and sensor nodes become cheaper, deployed sensor networks will feature more nodes. This work examines the scalability behavior of routing and compression algorithms as the number of nodes in a sensor network increases. We demonstrate the longer lifetimes of clustered routing algorithms as they increase in size, the superior spatial distribution of node deaths in hierarchical clustered and high-compression multihop routing algorithms, and the characteristics of optimal cluster head distribution in clustered routing algorithms.

## 1 Introduction

The advent of low-cost and low-power sensors, increasingly programmable and powerful sensor nodes, and maturing radio technologies have led to an explosive growth in the use of *sensor networks* targeting a variety of distributed or remote sensor applications. These sensor networks connect large numbers of independent *sensor nodes* to make measurements of interest over a geographic region.

Because of their small physical size and low cost, sensor nodes have a number of constraints in their operation: they have limited processing ability, limited storage on the node itself, and limited power. In particular, the goal of minimizing power has motivated considerable work in the sensor network community. Recent research in sensor network system and protocol design aims to reduce and optimize the power usage of sensor nodes and, more generally, of the sensor network as a whole.

Current sensor network deployments in the ?eld typically feature tens to hundreds of nodes [1, 13], with more constrained environments allowing larger networks [10]. The continued decrease of the cost of nodes and sensors allows and motivates ever-larger deployments, deployments that must scale to tens or hundreds of thousands, or even millions, of nodes. Networks of such sizes will ?nd applications in many areas; a representative example would be in the ?eld of precision agriculture, which advocates individual control of each plant. Large citrus groves or vineyards under precision agriculture techniques would thus require networks considerably larger than the deployed networks of today.

As more sensor networks are deployed and the sizes and densities of these emerging networks grow, so must the algorithms for collecting information in these networks. In this work, we analyze the scalability of these algorithms in an effort to characterize and identify the most important problems facing the wide-scale deployment of sensor networks.

We concentrate on the following problem: we place a set of sensor nodes in a constrained geographic patch. Each of these nodes collects a measurement at a periodic time interval and must send this measurement over a radio to a distant base station. Speci?cally, in this problem, all nodes are randomly placed in a 50 m by 50 m region and must communicate their measurements to a base station located 100 m from this region[1]. We assume the base station's power is unconstrained, but each node has a ?xed and limited amount of power. Such a situation would be applicable to our agricultural examples, in which a base station may have access to external power (located at a farmhouse or other powered source) but individual nodes do not.

This scenario is well-suited for studying scalability issues. First, dense networks are the most appropriate class of networks with which to begin this study. Scaling sensor networks to many thousands of nodes requires that the nodes have a low cost. The relatively short distances in this scenario allow nodes to operate with inexpensive radios and smaller batteries; the cost of a sensor network in this scenario would not be wholly limited by communication cost. The base station position we have chosen ensures that while the amount of power required for messages to reach the base station is signi?cant, it does not completely

[1]In our coordinate system, nodes are located in the axis-aligned square bounded by (0, 0) and (50, 50), with the base station located at (25, -100).



John D. Owens. "On The Scalability of Sensor Network Routing and Compression Algorithms". Technical Report ECE-CE-2004-1, Computer Engineering Research Laboratory, University of California, Davis, 2004.

dominate the aggregate energy budget of the sensor nodes. Second, as sensor nodes become cheaper, many researchers will mitigate node reliability problems by simply deploying large numbers of nodes in a small area, knowing the redundancy across many nodes will allow them to gain an accurate picture of phenomena in their region of interest. The consequences of deploying large numbers of nodes are precisely what we analyze here.

In this scalability analysis we concentrate on the following aspects of a sensor network: its *routing algorithm*, which decides the communication patterns between nodes, and its method of *data aggregation and compression*. Aggregation allows multiple packets to be combined into a single packet; compression permits the data in a packet to be reduced in size. Per-node aggregation and compression techniques reduce the transmission cost of data by reducing the amount of data at the cost of additional computation. As per-operation power costs decline with advances in technology, this tradeoff is increasingly advantageous.

In particular, we analyze the following classes of routing algorithms (described in more detail in Section 3.2):

**DIRECT** Direct send is the simplest routing protocol: all nodes send their packets directly to the base station.

**MULTIHOP** Multihop methods minimize the aggregate transmission energy necessary to communicate a packet from its source to its destination. Because the cost of sending a packet is proportional to the square of the distance, it costs less power to route a packet over several shorter hops than one large hop.

**CLUSTERED** Clustered schemes designate some of their nodes as "cluster heads"; nodes that are not cluster heads send their measurements to cluster heads, which in turn send them to the base station. These schemes benefit from data aggregation and possibly compression at the cluster heads.

**HIERARCHICAL** Hierarchical clustered schemes differ from their clustered cousins in that they have multiple levels of clustering. While a clustered network has only one level of cluster heads, which then must send data directly to the base station, a hierarchical protocol allows these cluster heads to in turn group into another level of cluster heads, and so on.

We also analyze three classes of compression schemes, no compression, square-root compression, and perfect compression (described in detail in Section 3.3).

The rest of this paper is organized as follows. We describe previous work in the next section, Section 2. Section 3 explains the routing and compression algorithms in more detail. In Section 4, we detail our experimental setup and simulation environment. Our results, and discussion of these results, are detailed in Section 5. We offer suggestions for future work in Section 6 and conclude in Section 7.

## 2 Previous Work

The work presented here builds on the work of other researchers in several categories, including multihop, clustered, and hierarchically clustered routing algorithms; compression and aggregation techniques; and theoretical scalability analyses.

Distributed algorithms for routing in wireless networks is a longstanding and interesting problem. A representative example of early work considering mobile, radio-equipped nodes is the 1981 work of Baker and Ephremides [2]. Multihop methods for routing have been proposed and analyzed by many authors, among them Ettus [5], Meng and Rodoplu [14], Singh et al. [20], and Chang and Tassiulas [4].

LEACH, introduced by Heinzelman et al., was one of the first clustered routing protocols to be described in the context of wireless sensor networks [9]. Their work provides both theoretical and (simulation-based) experimental results to demonstrate the advantages of their clustered routing protocol. In the context of the terminology of this paper (described in Section 3), they compare networks with 100 nodes in 3 scenarios: LEACH and PERFECT; MULTIHOP and NONE; and DIRECT and NONE. This work is described in more detail by the lead author in her dissertation [8].

More complex protocols require knowledge of more system state than we assume in this work. For example, Lindsey et al. improve upon LEACH's energy usage with their PEGASIS scheme [11]. In PEGASIS, each node transmits to its nearest neighbor such that all data in a cluster flows along a ring network before finally ending up at the cluster head for long-range transmission. As they note in their paper, their algorithm assumes "all nodes have global knowledge of the network." Another example would be Liu and Lin's clustering method that makes cluster head decisions based on the remaining energy in the nodes [12].

Bandyopadhyay and Coyle summarize much of the previous work in hierarchical clustering algorithms in their recent paper [3], which concentrates on minimizing the overall energy expenditure in the network. Their analysis uses a simpler radio and energy model than the one we use in this paper (Section 3) in which all radios have the same range and power expenditure for any message. Though they do some analysis of the effects of increasing density by increasing node count, the bulk of the paper concentrates on demonstrating the effectiveness of their distributed algorithm for hierarchical clustering, which has better complexity than previous $O(n)$ algorithms and is thus suitable for networks with large numbers of nodes.

Recent research points the way to more powerful and efficient compression in sensor networks. Assuming data with some correlation (non-random data), Pradhan et al. show that correlation between nodes can be leveraged for network compression without collaboration between nodes using distributed source coding [18]. They achieve a 33% reduction in transmission cost in a small example using their DISCUS algorithm. Further gains can be achieved by compressing or aggregating data from multiple nodes, assuming data from those nodes is also correlated. An example of a compression/aggregation method is Petrović et al.'s proposal to reduce the overhead of transmitting IDs for each sensor measurement in aggregated data transmission [16].

Gupta and Kumar present theoretical bounds for the scalability and capacity of multihop ad hoc networks [7]. For networks in which all nodes generate uncorrelated data, and source and destination nodes are selected randomly, they demonstrate that the throughput per node for a network of $n$ nodes is $O(1/\sqrt{n})$. However, Servetto shows that as long as the rate of information per transmitter decreases more quickly than the overall network throughput, scalable communication is achievable [19].

# 3 Algorithms

In this section, we describe our implementation of each of the routing and compression algorithms studied in this paper. We begin with our assumptions about the characteristics of the class of sensor network we consider in this work, then discuss the routing and compression algorithms.

## 3.1 Assumptions

For the purposes of this study, we make the following assumptions about our system (recognizing and describing the assumptions that are difficult or unrealistic in practice):

- Nodes communicate using radios. The cost of transmitting a $k$-bit packet is the sum of the cost of operating the transmitter electronics ($E_{Telec}$) and the cost of radio transmission ($E_{Tamp}$). This latter quantity depends on the distance $d$ between source and destination. The overall cost is

  $$\begin{aligned} E_T(k,d) &= E_{Telec}(k) + E_{Tamp}(k,d) \\ &= E_{elec}k + \varepsilon_{amp}kd^2 \end{aligned}$$

  The cost of receiving a packet is $E_R(k)$ and is given by

  $$E_R(k) = E_{elec}k$$

  Finally, compressing schemes (SQRT and PERFECT) require energy to fuse messages. We assume this cost is proportional to the number of received input bits to each node:

  $$E_F(k) = E_{fuse}k$$

  These radio parameters are the same as those described by Heinzelman et al. [9], and we use their values for each of the constants as well: $E_{elec}$ = 50 nJ/bit, $\varepsilon_{amp}$ = 100 pJ/bit/m$^2$, and $E_{fuse}$ = 5 J/bit.

- Each node produces one measurement per round of a fixed size. In this paper, each measurement is 2000 bits. $N$ packets can always be aggregated into a single packet; there is no limit to the amount of data per packet. A realistic deployment, in contrast, would likely be able to aggregate many packets into a single packet, but would have a maximum size per packet.

- Nodes have finite amounts of energy. In this paper, each node has 0.5 J of energy. The base station is assumed to have unlimited energy.

- Nodes have finite storage and limited computation capability. We thus assume no node can have global knowledge of its environment. In fact, as network sizes scale, it is increasingly unrealistic to assume that even the base station has global knowledge of the environment. Consequently we require in this study that all routing protocols can be implemented using strictly local knowledge: we only assume that each node knows its distance to the base station, can make queries of its immediate neighbors ("what is my nearest cluster head"), and can calculate a distance to a neighbor (through time-of-flight or signal strength estimation). This assumption rules out protocols in this analysis which require global knowledge of the system (for example, PEGASIS [11]).
- Nodes can generate random numbers.
- We do not account for routing setup energy costs. Although this is obviously an unrealistic assumption in practice, particularly with high node densities, this cost is highly dependent on the implementation of the routing protocol. In this work we present feasible routing protocol implementations that meet our assumptions, but recognize that future work must concentrate on these setup costs. Consequently the results presented here are best-case results.
- We do not account for radio contention, meaning we do not consider the effects of packet loss or radio collisions. Again, this is an unrealistic situation in practice, but considerable work has demonstrated the feasibility of allowing multiple nodes using the same frequency bands to efficiently share the bandwidth of that channel (TDMA, CDMA, etc.).
- We place no message count or bandwidth limits on any individual node. In practice, sensor node radios can be limited by input or output message count rate (usually a limit of the processing time to process an individual packet on a node) or by input or output bandwidth (limited by the radio's maximum bandwidth).

### 3.2 Routing Algorithm Detail

Given a source-destination node pair, routing algorithms determine the next-hop destination for the source node's packets. In the context of this problem, implementing different routing algorithms equates to simply implementing different routing tables. We discuss our routing algorithms of interest below. With the exception of HIERARCHICAL, all of our routing tables are equivalent to DAGs (directed acyclic graphs) with the base station as the ultimate destination. We process a new round of messages by maintaining a queue at each node and traversing this DAG in postorder (always processing subtrees of the DAG before their roots).

**DIRECT** In a direct send protocol, each node simply routes directly to the base station. Direct send does not benefit from aggregation because no node ever handles any packets from a different node.

**MULTIHOP** Because the transmission cost of sending a packet is proportional to the square of the distance, multihop methods can decrease the aggregate transmission energy necessary to communicate a packet from its source to its destination by choosing a routing path that uses intermediate hops between the source and destination. Calculating the absolute minimum energy path from source to base station requires global knowledge of the network; instead we approximate this metric with a locally optimal metric by simply sending packets to the closest node that is closer to the base station.

**CLUSTERED** Clustered schemes designate some of their nodes as "cluster heads"; nodes that are not cluster heads send their measurements to cluster heads, which in turn send them to the base station. In this study we implement the LEACH (Low-Energy Adaptive Clustering Hierarchy) protocol [9] as a representative clustering protocol. LEACH specifies a probability $p$ that each node will become a cluster head on each round[2]. Nodes make cluster head decisions based only on a locally-generated random number and do not require any global knowledge of the network; the protocol ensures that a node is a cluster head once and only once every $1/p$ rounds. After cluster head decisions are made, nodes route to their nearest cluster heads, and cluster heads aggregate all incoming messages and route directly to the base station. Because nodes send to their nearest cluster head, all messages processed by a single cluster head are geographically proximate and thus potentially amenable to aggregation.

**HIERARCHICAL** Hierarchical clustered schemes are motivated by the realization that if a clustering algorithm such as LEACH gains benefits by clustering its nodes, we should continue to gain benefits by clustering the cluster heads themselves. While LEACH's cluster heads send directly to the base station, implying only a single layer of clustering, hierarchical protocols allow multiple layers. We designate the lowest level of clustering as level 0; nodes that are not cluster heads

[2] As an example, Heinzelman et al. consider the same sensor region and base station placement we assume in this study. They demonstrate that a random node distribution of 100 nodes over this region, routed with LEACH and assuming perfect aggregation and compression, has a maximum network lifetime when the cluster head percentage $p$ is 0.05.

route to level 0 cluster heads. Level 0 cluster heads route to level 1 cluster heads, which route to level 2 cluster heads, and so on. Finally, the highest level of cluster heads route directly to the base station.

We made several decisions in implementing this protocol:

**How are cluster heads chosen?** LEACH chooses cluster heads using a random number generated at each node. We make cluster head decisions in the same way, but extend this implementation by generating a random number for each level of cluster heads. For example, if we have both level 0 and level 1 cluster heads, we generate two random numbers, one for each level. Per-node decisions for each level of cluster head are made independently, so it is possible that any single node might be a cluster head at zero, one, many, or all cluster-head levels at any time.

**How many levels of cluster heads are necessary?** We argue that the network should look the same from any level of the network. If nodes that are not cluster heads expect a ratio of 20 nodes that are not cluster heads to one level-0 cluster head, then level-0 cluster heads should expect a ratio of 20 level-0 cluster heads to one level-1 cluster head. In practice, this means that LEACH's probability of becoming a cluster head $p$ becomes $p^{l-1}$, where $l$ is the level of cluster head. Given this exponential structure, we add levels of clustering until the point that the expected number of cluster heads at any level is less than 1.

**How is information passed between levels?** We mandate that any node at level $n$ can only send information to a node at level $n+1$; this disallows a node that is a cluster head at levels 1 and 3, for instance, from passing its data directly from level 1 to level 3 without going through a level 2 intermediate cluster head. (Nodes that are cluster heads at two adjacent levels, however, may pass information directly between the levels without incurring a send or receive cost.) This decision likely results in slightly poorer lifetime overall, but better locality for aggregation and compression purposes.

**How often are clustering decisions made?** For the simulations in this paper, unless otherwise noted, new routing decisions are made every round. In general, more frequent rerouting provides more randomization and hence better performance, so this decision gives a best-case estimate.

## 3.3 Compression Algorithm Detail

We analyze the routing algorithms above with three compression schemes, ranging from no compression (NONE) to intermediate compression (SQRT) to perfect compression (PERFECT). We assume that all three schemes can *aggregate* packets: if multiple packets arrive at a node in a given round, those packets can be combined into a single packet. Comparison between schemes is facilitated by normalizing packet sizes such that a packet sent from a leaf node of the routing tree is always 2000 bits[3].

**NONE** No compression is performed when packets are aggregated at a single node. In other words, $m$ packets arriving at a single node with size $N$ produce a single packet of size $mN$. This class of compression schemes represents scenarios in which every measurement must be delivered to the base station and those measurements are uncorrelated and incompressible.

**SQRT** Each packet keeps track of its uncompressed (raw) size; data can be compressed to the square root of its uncompressed size. Packets can be aggregated together, and the resulting packet has the size of the square root of the total uncompressed size of its constituent packets. The packet's compressed size is used in all energy calculations. For example, a packet with an uncompressed size of $N$ bits is compressed to $\sqrt{N}$ bits when sent. Aggregating $m$ $N$-bit packets together results in a packet of compressed size $\sqrt{mN}$. Implementing SQRT may require a fairly aggressive or lossy compression method; Servetto has demonstrated that this amount of compression results in a scalable (but barely scalable) network [19].

**PERFECT** The PERFECT scheme requires perfect compression/aggregation of all packets; any number of packets of size $N$ arriving a node can be combined into a single packet of size $N$. Such a scheme would be applicable in tasks that need to count measurement events but not record the measurements. This scheme is scalable; both LEACH and PEGASIS use this compression method in their respective papers [9, 11].

It is important to note at this point that we are *not* proposing compression/aggregation implementations that deliver the performance of SQRT or PERFECT; instead we demonstrate the scalability of the routing protocols under consideration when such compression/aggregation schemes *are* implemented.

[3] In practice this only impacts SQRT; generated packets using this compression scheme have a compressed size of 2000 bits and an uncompressed size of 40,000 bits.

# 4 Experimental Setup and Simulation Environment

All results in this paper were obtained in simulation from a custom simulator written in C++. The simulator has two major functions: routing and message generation/propagation. Rerouting is triggered in two circumstances: when a node dies or when a timer expires (for example, the clustering protocols can be set to reroute every $n$ rounds). After the routing tables are constructed, messages are generated and propagated. On each round, each node generates one message. Starting at the leaves of the routing tree(s), messages are processed and propagated to succeeding nodes until the base station is reached.

Particularly with simulations that reroute frequently, the nearest-neighbor calculations (to find the nearest node in the proper direction for MULTIHOP or to find the nearest cluster head in the clustering protocols) are the most time-consuming. However, precomputing sorted nearest-neighbor lists per node required too much memory for the large simulations. Constructing kd-trees for nearest-neighbor search [6] proved to be the best balance between efficiency and memory usage. We used the publicly-available ANN library [15], configured for exact nearest-neighbor search, for our kd-tree implementation.

All simulation was performed on a Apple Macintosh 1 GHz PowerBook G4 and compiled with gcc 3.3. As an example, the runtime of a simulation to completion for three networks with hierarchical routing and perfect compression (the most computationally demanding configuration) with 100, 1000, and 10,000 nodes took 2.4, 51.6, and 687.4 seconds of user time respectively[4].

# 5 Results and Discussion

In analyzing the scalability of these routing and compression algorithms, we want to answer four questions. First, how does scalability impact network lifetime (Section 5.1)? Next, how do scalability and the choice of routing and compression algorithm affect the spatial pattern of node deaths (Section 5.2)? How do we configure the clustered protocols as the number of nodes increase for maximum lifetime (Section 5.3)? And finally, how does the energy consumption of the network change with the routing algorithm, the compression algorithm, and the network size (Section 5.4)?

## 5.1 Network Lifetime

The first result we analyze is the lifetime of the network under different routing and compression algorithms. The first important question to answer is what, exactly, we mean by "lifetime." Lifetime is traditionally defined as the amount of time between the start of dataflow in the network and the time a certain percentage of nodes have run out of energy. Choosing that percentage is a tricky task, however, because the pattern of how nodes die varies between different algorithms. For the purposes of this study, we define lifetime as the length of time until half the nodes run out of energy.

Figure 1 shows node lifetimes at each of 3 different network sizes with each of 3 different compression schemes. Each plot represents the pattern of node deaths for the given combination of network size, compression scheme, and routing algorithm. The clustering schemes were measured with their optimal lifetimes (we chose the value of $p$ for each of the clustering schemes that maximized the lifetime of the network at that data point). From this data we can draw a number of conclusions.

The most important conclusion is that *for less aggressive compression schemes, the clustered routing algorithms substantially outperform* MULTIHOP*; for* PERFECT *compression,* MULTIHOP *leads the clustered protocols for small network sizes, but as the network size grows, the clustered protocols look increasingly attractive*. The clustering algorithms scale well as network sizes grow, particularly HIERARCHICAL, which continues to add levels of clustering with increasing node counts. MULTIHOP's advantage with perfect compression in small networks lies in the very small number of nodes that communicate with the base station in any MULTIHOP protocol, coupled with the perfect compression at those nodes. As network sizes increase, the clustering protocols have a smaller and smaller percentage of their nodes communicating directly with the base station, so they gain the same benefit as MULTIHOP.

We can also conclude that *increasing the size of the network also increases the lifetime of the network* (with the exception of the DIRECT protocol). The increasing opportunities for aggregation in larger networks are responsible for this increase in lifetime;

[4] Runtime is not linear with network size for several reasons: the cost of constructing the kd-tree is greater than $O(n)$; the cost of accessing the kd-tree per node is greater than $O(1)$; cache behavior is poorer as the total memory footprint increases; and large networks typically have longer lifetimes than smaller ones.

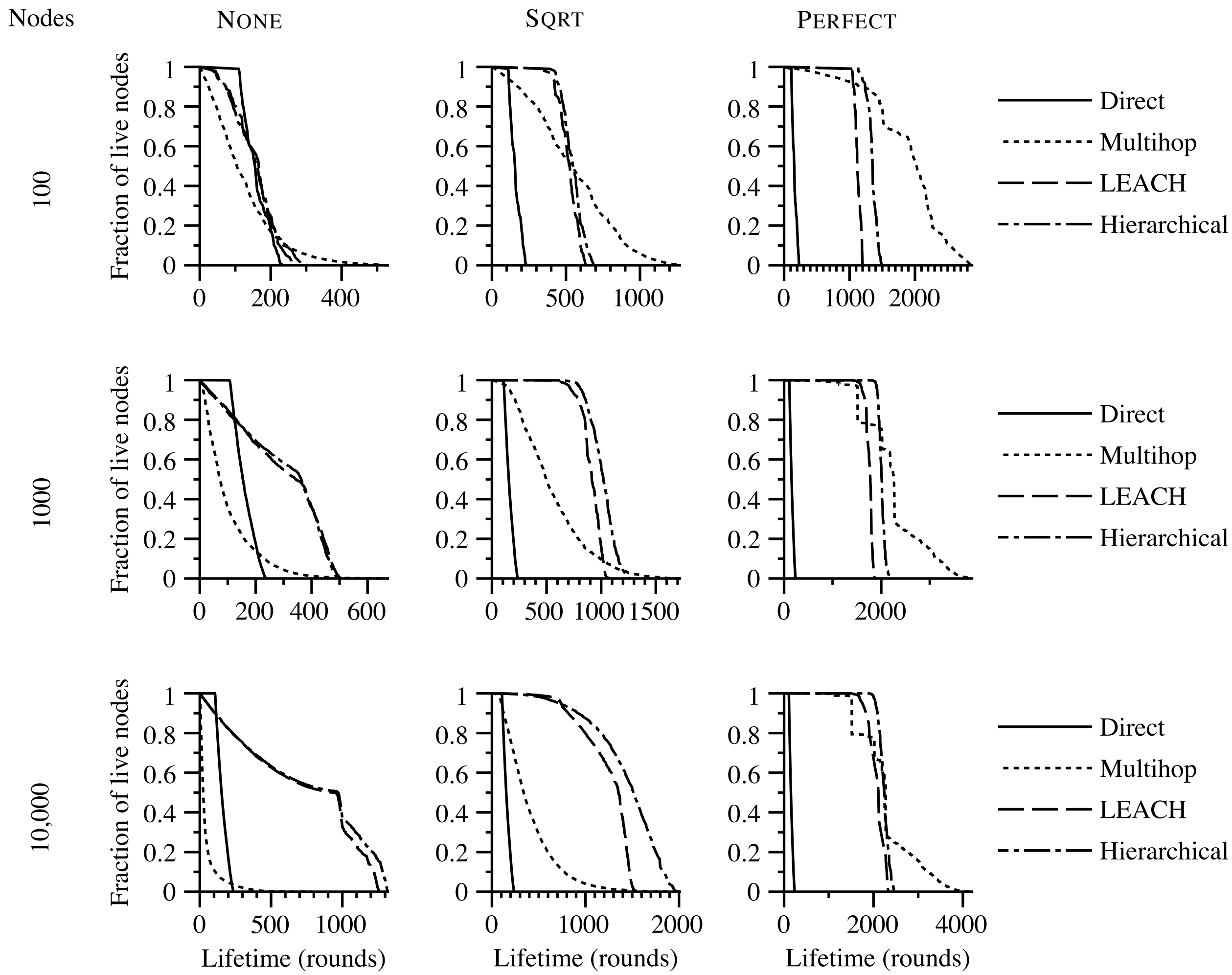



Figure 1: Node lifetime as a function of sensor network size (rows) and compression scheme (columns). In each graph, the curve representing each {size, routing, compression} experiment represents the percentage of live nodes on the y axis plotted against the time step on the x axis.

a smaller fraction of nodes is necessary to send data to the base station. However, we also note that continuing to increase the size of the network produces diminishing gains in lifetime gains as network sizes become larger.

For the clustered protocols, *increasing the degree of compression increases the lifetime of the network, with the lifetime increasing as the network size increases.* Compression implies that the aggregate amount of data transmitted to the base station, normalized by the number of nodes, decreases as the network size increases. Even NONE, which aggregates but does not compress, increases overall network lifetime because one node sending *n* packets is more efficient than *n* nodes sending one packet each.

*The performance of* DIRECT *is unaffected by the compression algorithm or the network size.* With DIRECT, each node behaves independently, since all messages are sent directly to the base station. And because aggregation does not occur with a direct-send routing algorithm (because there are no intermediate nodes), the aggregation algorithm is also immaterial.

*The performance of* MULTIHOP *is most markedly affected by the choice of compression algorithm.* Because MULTIHOP has the highest number of average network hops per message, it has the greatest opportunity to aggregate and compress packets at intermediate nodes. That aggregation performance is the leading factor in determining the lifetime of a MULTIHOP-based network.

*Between the clustering schemes,* HIERARCHICAL *has a slight but measurable advantage in network lifetime over* LEACH. The difference between these routing algorithms is not a large one, but is present in both SQRT and PERFECT; the performance for NONE is equivalent between the two. We will see in the following sections that HIERARCHICAL has other advantages over

LEACH as well.

With the compression schemes SQRT and PERFECT, *both* LEACH *and* HIERARCHICAL *have a much smaller time gap between the first node death and the last node death when compared to* MULTIHOP. Stated another way, the standard deviation across the time of all node deaths is much smaller for the clustering schemes. We can quantify this characteristic by dividing the difference between the times of node deaths that result in 90% and 10% live nodes and the network lifetime (which we have defined at the 50% point). As an example, we choose a 1000 node network and SQRT aggregation. For MULTIHOP, this ratio is 1.57 (meaning the difference between the 90% point and the 10% point is 1.57 times as large as the 50% lifetime); for LEACH, 0.28; for HIERARCHICAL, 0.29. Sensor network applications with different needs may make a design decision based on this property. For example, an application most interested in maintaining at least some coverage for as long as possible may lean toward MULTIHOP, while a need for a critical mass of sensors for as long as possible may prefer the clustered protocols.

## 5.2 Spatial Patterns of Node Death

Network lifetime is not the only metric of interest, however. Many sensor network applications are not only interested in protocols that allow nodes in the system to last as long as possible but also in the spatial distribution of node death. These applications wish to guarantee at least some degree of coverage of all areas of the geographic region of interest as nodes die. In other words, they prefer that node death be uncorrelated with geographic position.

Figure 2 shows the pattern of node death for 12 simulations of 1000 nodes at the point where half the nodes are alive and half are dead. The figure depicts the results of testing each of the four routing algorithms against each of the three compression schemes.

For each simulation we mark the nodes that are alive and dead as well as calculate the centroid (average) of the alive and dead nodes. We are interested in the distance between the alive centroid and the dead centroid. If node death is uncorrelated with position, the two centroids will be in the same place. Any correlation with position will manifest itself as geographically separated centroids.

From these results we can draw the following conclusions:

*With any compression scheme, a* DIRECT *routing protocol results in a strong correlation between node liveness and position.* Specifically, the nodes that are alive at the halfway point are those closest to the base station. Such behavior is expected, as every node incurs the same amount of network traffic, and distant nodes must expend more energy to send packets to the base station.

*While* MULTIHOP *with no compression suffers from strong correlation between node position and liveness, increasing the amount of compression mitigates this effect.* MULTIHOP with no compression has the opposite correlation from DIRECT; the closest nodes to the base station, which are the only nodes that communicate with the base station, are the first to die. As the amount of compression increases, however, the cost of sending to the base station lessens, and the dominant power factor shifts toward the next-hop distance and the number of upstream nodes in the MULTIHOP routing tree, two characteristics that are largely uncorrelated with location. Close inspection of the MULTIHOP-PERFECT simulation result reveals that all the nodes at the bottom of the plot have died from the cost of the send to the base station. However, the onset of substantial node death curve is much more delayed for PERFECT than for the other aggregation schemes (Figure 1), so in a sense, those nodes closest to the base station can "protect" the other nodes from a long-distance communication for a much longer time period.

*Between the clustered protocols,* HIERARCHICAL *has less correlation than* LEACH *between node death and node position.* Multiple layers of clustering in a large network such as this one essentially do a better job of randomizing the high-energy costs of aggregation and long-distance transmission than does the single layer of LEACH.

## 5.3 Selection of Cluster Head Percentage

We have previously described our method for choosing cluster heads in both LEACH and HIERARCHICAL networks. Both are parameterized by a fraction $p$ of nodes serving as cluster heads in each round of communication. The selection of $p$ that maximizes network lifetime depends on several factors, however, including network size and density, routing algorithm, and compression algorithm. This section describes simulation results that help explain the impact of these factors.

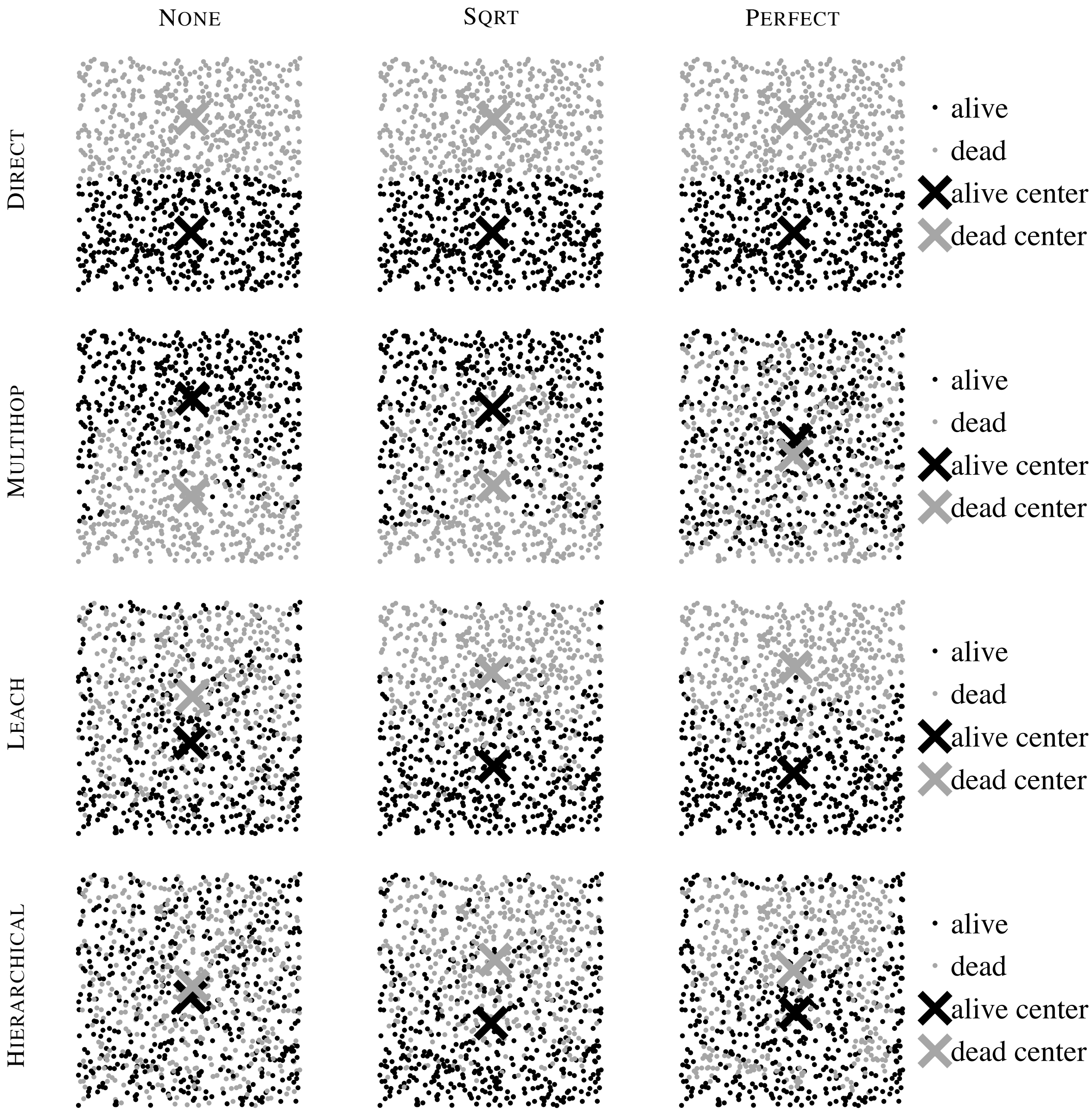



Figure 2: Each plot represents the point in the 1000-node simulation at which half the nodes are alive and half are dead. All measurements are taken with $p$ = 0.05. Each row represents a routing algorithm (DIRECT, MULTIHOP, LEACH, and HIERARCHICAL); each column represents a compression scheme (NONE, SQRT, and PERFECT). For each plot, the base station is located 100 m below each sensor patch. Live nodes are marked with black dots, dead nodes with grey dots. The centroids of the alive and dead nodes are marked with black and grey X's, respectively. The distance between the alive centroid and the dead centroid is a measure of the dependence of node death on node position. One interesting point to note is the survival of perimeter nodes in HIERARCHICAL, particularly with higher compression; perimeter nodes have fewer neighbors on average than nodes in the center, hence they aggregate data from fewer nodes when they are cluster heads and consequently have a longer average survival time. Not shown in the graph are results from rerouting LEACH and HIERARCHICAL every 20th round instead of every round; less rerouting increases randomness, and the alive and dead centroids for all compression schemes are closer together when rerouting is less frequent.

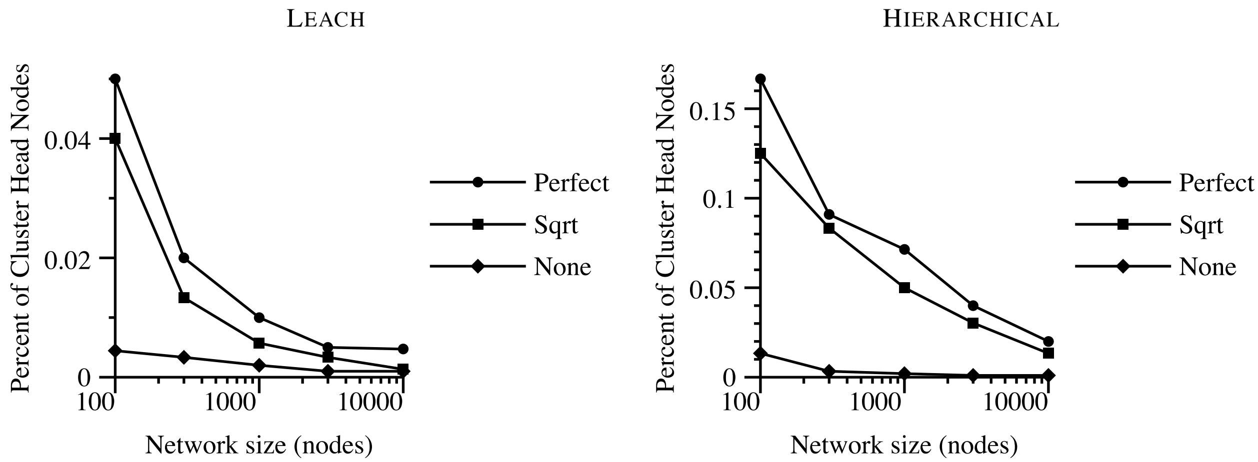



Figure 3: Optimal percentage of cluster heads ($p$) plotted against network size for LEACH- and HIERARCHICAL-routed networks at each of three compression schemes (NONE, SQRT, and PERFECT).

Figure 3 shows the optimal percentage of cluster heads as a function of network size and routing algorithm. We concentrate primarily on SQRT and PERFECT because we have already seen (Figure 1) that clustering makes little difference under the NONE compression scheme.

Three major trends are evident:

First, *as network sizes increase and the network becomes more dense, the optimal percentage of cluster heads ($p$) decreases*. For LEACH, changing the percentage of cluster heads is a tradeoff. More cluster heads means a smaller average distance for each node's transmission to the cluster head. Fewer cluster heads gives more opportunity for aggregation at each cluster head and fewer nodes incurring the cost of sending to the base station. In the context of this paper, as we increase the network size, we also increase the density of nodes. We can see in Figure 3 that the overall density of cluster heads continues to rise with network size even as the percentage of them decreases. Consequently, if we follow the optimal curve, as the network size increases, the average distance to a cluster head decreases. The argument is similar for HIERARCHICAL networks, but in addition, because of the larger network sizes, the smaller percentage of cluster heads are balanced against more cluster heads overall and also deeper layers of cluster heads.

This trend is also interesting from the point of view of networks with fixed cluster heads (unlike these experiments) as it indicates that as the network grows, the number of fixed (and possibly more expensive) cluster heads grows more slowly than the number of nodes.

Second, *more aggressive compression protocols are more efficient with more cluster heads (larger $p$) than less aggressive protocols*. Having many cluster heads means many nodes incur the expensive cost of sending to the base station. More aggressive compression techniques lower this cost when compared to less aggressive ones.

Third, *the multi-layer* HIERARCHICAL *requires a higher percentage of cluster heads than the single-layer* LEACH. To take advantage of multiple layers of clustering and its associated benefits in aggregation, HIERARCHICAL benefits from more cluster heads.

Another important question is the sensitivity of the network lifetime to the cluster head percentage $p$. Figure 4 shows data for a representative network size of 1000 nodes. We note two important trends:

*The lifetime of a* LEACH*-routed network is much more sensitive to the percentage of cluster heads $p$ than one routed with* HIERARCHICAL. In other words, the peak of LEACH's lifetime-vs.-$p$ curve is considerably sharper than HIERARCHICAL's. If too few nodes are cluster heads, non-cluster-head nodes must send each message longer distances. Too many cluster heads, on the other hand, leads to too many nodes broadcasting directly to the base station. LEACH's sharper curve has implications in sensor network deployment: designers of a sensor network using LEACH must be particularly careful in choosing their network's percentage of cluster heads, because miscalculating this network parameter has a significant effect on the lifetime of the network.

Though it is not shown in Figure 4, *as network sizes increase, routing protocols become increasingly sensitive to the cluster*

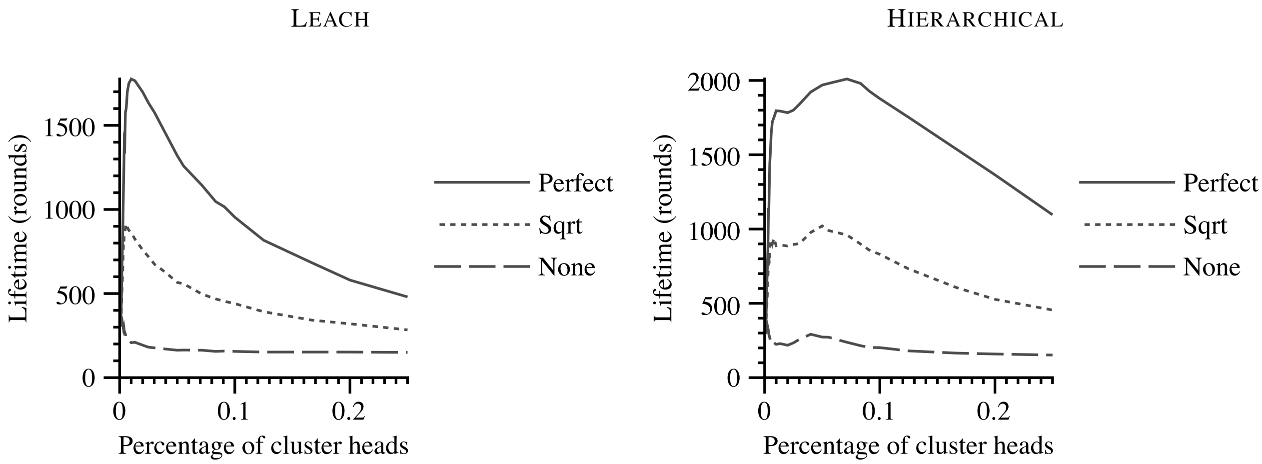



Figure 4: Network lifetime vs. percentage of cluster heads ($p$) for LEACH- and HIERARCHICAL-routed networks with 1000 nodes at each of three compression schemes (NONE, SQRT, and PERFECT).

*head percentage*. The peak where cluster head percentage is optimal becomes narrower as network sizes increase for both LEACH and HIERARCHICAL.

### 5.4 Energy Consumption

Our final analysis looks at the distribution of energy consumption across network sizes, routing algorithms, and compression algorithms. Figure 5 shows the breakdown of energy consumption between the cost of sending a packet ($E_T$), receiving a packet ($E_R$), and fusing multiple packets into a single packet ($E_F$). The case of DIRECT routing does not vary with node size or compression algorithm, but the other routing algorithms deserve more attention.

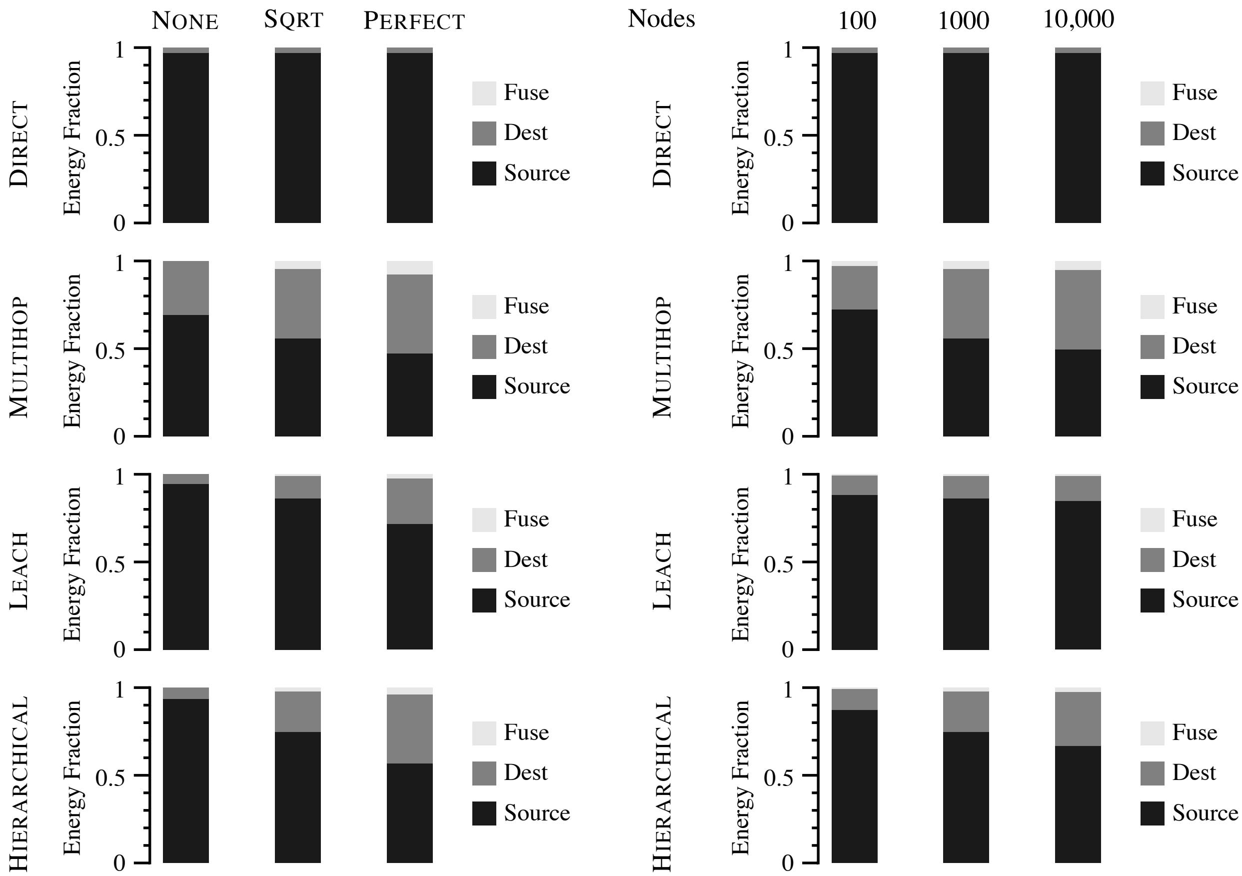
We find that *more aggressive compression decreases the fraction of energy devoted to sending messages in favor of energy devoted to receiving and fusing messages*. Higher compression rates result in smaller packets, so the cost of sending packets decreases. We see that the lowest fraction of energy for sending is for PERFECT compression and either MULTIHOP or HIERARCHICAL routing. These two routing algorithms have the smallest average hop distance, so the sending costs are correspondingly less than the other algorithms. We can also see that *the fraction of energy devoted to sending packets decreases as network size and density increase*. This decrease is a direct consequence of the decrease in the average hop distance. Denser networks mean a node's distance to its nearest nearest neighbor is shorter; larger networks mean proportionately fewer nodes must communicate with the base station (Figure 3).

## 6 Future Work

In this study we have made many assumptions which could be investigated in future work. Among them are the location of the base station and the area of the region in which the sensor nodes are placed; a radio model that could be made more realistic (including an energy-distance dependence with a $r^4$ term as well as the $r^2$ term we use; most radios also have a maximum range); a radio model that only allows discrete power levels instead of the continuous one we assume; a realistic radio contention model; a realistic energy cost model for routing setup; nodes with heterogeneous capabilities; and more options in the compression-vs.-power tradeoff. Many of these assumptions cannot be considered in isolation; as an example, having a maximum broadcast distance for some nodes decreases the amount of long-range transmissions, which might dramatically affect radio contention.

We chose to support more nodes by increasing density, but an equally interesting study could keep the same density and increase the coverage area. We also would like to look at uneven distributions of node energy across nodes and protocols that take node energies into account when routing and compressing. Another area that bears further investigation is sensor network communication patterns that do not simply send all data to a single base station.

As we noted in Section 3, our model does not assume radio upper bounds on either message count or total bandwidth per




(a) Energy consumption in 1000-node sensor networks, compared against routing and compression algorithms.

(b) Energy consumption in sensor networks using Sqrt compression, compared against varying network sizes and routing algorithms.

Figure 5: Energy consumption of sensor networks across different network sizes and different routing and compression algorithms. All measurements are taken with $p = 0.05$. The left figure shows the energy breakdown for 1000-node networks while varying the routing and compression algorithms; the right figure uses Sqrt compression and varies network size and routing algorithm. “Source” refers to energy spent sending a packet; “Dest” refers to the energy spent receiving a packet; and “Fuse” refers to the energy spent compressing and aggregating multiple packets into a single packet.

time on a node. Real radios, however, have real limits on these two quantities. We believe that the Hierarchical protocol, properly configured to account for per-node message limits per unit time, will be particularly effective at meeting this challenge.

Although theoretical justifications for the conclusions here are not a focus of this work, many of them will be well-suited for future study.

Finally, and perhaps most importantly, these simulations must be validated against real results with real sensor nodes and radios. We must learn techniques to confirm these results for large networks without necessarily deploying large networks: how can we extend conclusions from experimental small networks to the large networks we will ultimately deploy in the field? It is only when the conclusions we draw here are useful in real network deployments that we can consider them a success.

# 7 Conclusion

In this work we have analyzed the behavior of sensor network routing and compression algorithms as the number and density of sensor nodes increases. As these networks become large and dense, the clustered routing protocols, Leach and Hierarchical, become increasingly attractive in terms of node lifetime. We also demonstrated that Hierarchical has an advantage over Leach in the distribution of node deaths. Finally, we found the fraction of cluster heads necessary for maximum network lifetime in the clustered protocols decreases as the network grows and increases as compression becomes more aggressive.

Kris Pister has estimated that in the year 2010, radios will cost ten cents [17]. Coupling these radios with inexpensive processors results in large sensor networks with remarkable measurement capability at a low cost. In the coming years, researchers will thus have the opportunity to deploy larger and denser sensor networks of thousands or even millions of nodes. In so doing they will encounter a variety of scalability challenges in their algorithms and protocols. The work we have presented here will hopefully stimulate more research toward designing and constructing these sensor networks of the future.

# References


[1] Michael Allen, Deborah Estrin, Michael Hamilton, John Rotenberry, Phil Rundel, Mani Srivastava, and Stefano Soatto. Habitat sensing investigations: In-situ microclimate and organism monitoring. Technical report, UCLA Center for Embedded Networked Sensing, 2003.

[2] Dennis J. Baker and Anthony Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11):1694–1701, November 1981.

[3] Seema Bandyopadhyay and Edward J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of IEEE INFOCOM 2003*, volume 3, pages 1713–1723, March/April 2003.

[4] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM 2000*, volume 1, pages 22–31, March 2000.

[5] Matthew Ettus. System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks. In *Proceedings of the 1998 IEEE Radio and Wireless Conference (RAWCON 98)*, pages 55–58, August 1998.

[6] Jerome H. Freidman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[7] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, March 2000.

[8] Wendi Beth Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts Institute of Technology, June 2000.

[9] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocols for wireless microsensor networks. In *Proceedings of the Hawaiian International Conference on Systems Science*, pages 3005–3014, January 2000.

[10] David Lammers. Embedded projects take a share of Intel's research dollars. *EE Times*, 29 August 2001. `http://www.eetimes.com/story/OEG20010828S0090`.

[11] Stephanie Lindsey, Cauligi Raghavendra, and Krishna M. Sivalingam. Data gathering algorithms in sensor networks using energy metrics. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):924–935, September 2002.

[12] Jain-Shing Liu and Chun-Hung Richard Lin. Power-efficiency clustering method with power-limit constraint for sensor networks. In *Proceedings of the Performance, Computing, and Communications Conference*, pages 129–136, April 2003.

[13] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pages 88–97, Atlanta, GA, September 2002.

[14] Teresa H. Meng and Volkan Rodoplu. Distributed network protocols for wireless communication. In *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS '98)*, volume 4, pages 600–603, May/June 1998.

[15] David Mount and Sunil Arya. ANN: Library for approximate nearest neighbor searching. `http://www.cs.umd.edu/~mount/ANN/`, 1998.

[16] Dragan Petrović, Rahul C. Shah, Kannan Ramchandran, and Jan Rabaey. Data funneling: Routing with aggregation and compression for wireless sensor networks. *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 156–162, 2003.

[17] Kris Pister. Smart dust in 2010. `http://robotics.eecs.berkeley.edu/~pister/SmartDust/in2010`.

[18] S. Sandeep Pradhan, Julius Kusuma, and Kannan Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, 19(2):51–60, March 2002.

[19] Sergio D. Servetto. On the feasibility of large-scale wireless sensor networks. In *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, Urbana, IL, October 2002.

[20] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98)*, pages 181–190, October 1998.