Adaptive Protocols for Information Dissemination in Wireless Sensor Networks

Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan
Massachusetts Institute of Technology
Cambridge, MA 02139
Email:{wendi,jokulik,hari}@mit.edu

Abstract

In this paper, we present a family of adaptive protocols, called SPIN (Sensor Protocols for Information via Negotiation), that efficiently disseminates information among sensors in an energy-constrained wireless sensor network. Nodes running a SPIN communication protocol name their data using high-level data descriptors, called meta-data. They use meta-data negotiations to eliminate the transmission of redundant data throughout the network. In addition, SPIN nodes can base their communication decisions both upon application-specific knowledge of the data and upon knowledge of the resources that are available to them. This allows the sensors to efficiently distribute data given a limited energy supply. We simulate and analyze the performance of two specific SPIN protocols, comparing them to other possible approaches and a theoretically optimal protocol. We find that the SPIN protocols can deliver 60% more data for a given amount of energy than conventional approaches. We also find that, in terms of dissemination rate and energy usage, the SPIN protocols perform close to the theoretical optimum.

1 Introduction

Wireless networks of sensors are likely to be widely deployed in the future because they greatly extend our ability to monitor and control the physical environment from remote locations. Such networks can greatly improve the accuracy of information obtained via collaboration among sensor nodes and online information processing at those nodes.

Wireless sensor networks improve sensing accuracy by providing distributed processing of vast quantities of sensing information (e.g., seismic data, acoustic data, high-resolution images, etc.). When networked, sensors can aggregate such data to provide a rich, multi-dimensional view of the environment. In addition, networked sensors can focus their attention on critical events pointed out by other sensors in the network (e.g., an intruder entering a building). Finally, networked sensors can continue to function accurately in the face of failure of individual sensors; for example, if some sensors in a network lose a piece of crucial information, other sensors may come to the rescue by providing the missing data.

Wireless sensor networks can also improve remote access to sensor data by providing *sink nodes* that connect them to other networks, such as the Internet, using wide-area wireless links. If the sensors share their observations and process these observations so that meaningful and useful information is available at the sink nodes, users can retrieve information from the sink nodes to monitor and control the environment from afar.

We therefore envision a future in which collections of sensor nodes form ad hoc distributed processing networks that produce easily accessible and high-quality information about the physical environment. Each sensor node operates autonomously with no central point of control in the network, and each node bases its decisions on its mission, the information it currently has, and its knowledge of its computing, communication and energy resources. Compared to today's isolated sensors, tomorrow's networked sensors have the potential to perform their responsibilities with more accuracy, robustness and sophistication.

Several obstacles need to be overcome before this vision can become a reality. These obstacles arise from the limited energy, computational power, and communication resources available to the sensors in the network.

- **Energy:** Because networked sensors can use up their limited supply of energy simply performing computations and transmitting information in a wireless environment, energy-conserving forms of communication and computation are essential.
- **Computation:** Sensors have limited computing power and therefore may not be able to run sophisticated network protocols.
- **Communication:** The bandwidth of the wireless links connecting sensor nodes is often limited, on the order of a few hundred Kbps, further constraining inter-sensor communication.

In this paper, we present *SPIN* (Sensor Protocols for Information via Negotiation), a family of negotiation-based information dissemination protocols suitable for wireless sensor networks. We focus on the efficient dissemination of individual sensor observations to all the sensors in a network, treating all sensors as potential sink nodes. There are several benefits to solving this problem. First, it will give us a way of replicating complete views of the environment

Mobicom '99 Seattle Washington USA
Copyright ACM 1999 1-58113-142-9/99/08...$5.00

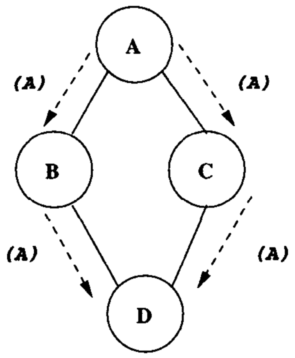

Figure 1: The implosion problem. In this graph, node A starts by flooding its data to all of its neighbors. Two copies of the data eventually arrive at node D. The system wastes energy and bandwidth in one unnecessary send and receive.

across the entire network to enhance the fault-tolerance of the system. Second, it will give us a way of disseminating a critical piece of information (e.g., that intrusion has been detected in a surveillance network) to all the nodes.

The design of SPIN grew out of our analysis of the different strengths and limitations of conventional protocols for disseminating data in a sensor network. Such protocols, which we characterize as *classic flooding,* start with a source node sending its data to all of its neighbors. Upon receiving a piece of data, each node then stores and sends a copy of the data to all of *its* neighbors. This is therefore a straightforward protocol requiring no protocol state at any node, and it disseminates data quickly in a network where bandwidth is not scarce and links are not loss-prone.

Three deficiencies of this simple approach render it inadequate as a protocol for sensor networks:

- *Implosion:* In classic flooding, a node always sends data to its neighbors, regardless of whether or not the neighbor has already received the data from another source. This leads to the implosion problem, illustrated in Figure 1. Here, node A starts out by flooding data to its two neighbors, B and C. These nodes store the data from A and send a copy of it on to their neighbor D. The protocol thus wastes resources by sending two copies of the data to D. It is easy to see that implosion is linear in the degree of any node.
- *Overlap:* Sensor nodes often cover overlapping geographic areas, and nodes often gather overlapping pieces of sensor data. Figure 2 illustrates what happens when two nodes (A and B) gather such overlapping data and then flood the data to their common neighbor (C). Again, the algorithm wastes energy and bandwidth sending two copies of a piece of data to the same node. Overlap is a harder problem to solve than the implosion problem—implosion is a function only of network topology, whereas overlap is a function of both topology and the mapping of observed data to sensor nodes.
- *Resource blindness:* In classic flooding, nodes do not modify their activities based on the amount of energy available to them at a given time. A network of embedded sensors can be "resource-aware" and adapt its communication and computation to the state of its energy resources.

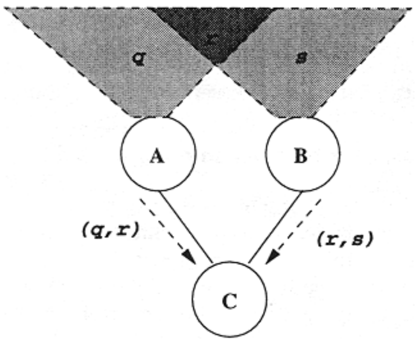

Figure 2: The overlap problem. Two sensors cover an overlapping geographic region. When these sensors flood their data to node C, C receives two copies of the data marked r.

The SPIN family of protocols incorporates two key innovations that overcome these deficiencies: *negotiation* and *resource-adaptation.*

To overcome the problems of implosion and overlap, SPIN nodes negotiate with each other before transmitting data. Negotiation helps ensure that only useful information will be transferred. To negotiate successfully, however, nodes must be able to describe or name the data they observe. We refer to the descriptors used in SPIN negotiations as *meta-data.*

In SPIN, nodes poll their resources before data transmission. Each sensor node has its own *resource manager* that keeps track of resource consumption; applications probe the manager before transmitting or processing data. This allows sensors to cut back on certain activities when energy is low, e.g., by being more prudent in forwarding third-party data.

Together, these features overcome the three deficiencies of classic flooding. The negotiation process that precedes actual data transmission eliminates implosion because it eliminates transmission of redundant data messages. The use of meta-data descriptors eliminates the possibility of overlap because it allows nodes to name the portion of the data that they are interested in obtaining. Being aware of local energy resources allows sensors to cut back on activities whenever their energy resources are low, thereby extending longevity.

To assess the efficiency of information dissemination via SPIN, we perform a simulation-based study of five dissemination protocols. Two of the protocols are SPIN protocols (which we call *SPIN-1* and *SPIN-2*); these are the experimental protocols in our study. The other three protocols function as comparison protocols: (i) *flooding,* which we outlined above; (ii) *gossiping,* a variant on flooding that sends messages to random sets of neighboring nodes; and (iii) *ideal,* an idealized routing protocol that assumes perfect knowledge and has the best possible performance.

We evaluate these protocols by measuring both the amount of data they disseminate over time and the amount of energy they dissipate. The SPIN protocols disseminate information with low latency and conserve energy at the same time. Our results highlight the advantages of using meta-data to name data and negotiate data transmissions. SPIN-1 uses negotiation to solve the implosion and overlap problems; it reduces energy consumption by a factor of 3.5 compared to flooding, while disseminating data almost as quickly as theoretically possible. SPIN-2, which additionally incorporates a threshold-based resource-awareness mechanism in addition to negotiation, disseminates 60% more data per unit energy than flooding and in fact comes very close to the ideal amount of data that can be disseminated per unit energy.

2 SPIN: Sensor Protocol for Information via Negotiation

The SPIN family of protocols rests upon two basic ideas. First, to operate efficiently and to conserve energy, sensor applications need to communicate with each other about the data that they already have and the data they still need to obtain. Exchanging sensor data may be an expensive network operation, but exchanging data *about* sensor data need not be. Second, nodes in a network must monitor and adapt to changes in their own energy resources to extend the operating lifetime of the system.

Our design of the SPIN protocols is motivated in part by the principle of Application Level Framing (ALF) [4]. With ALF, network protocols must choose transmission units that are meaningful to applications, i.e., packetization is best done in terms of Application Data Units (ADUs). One of the important components of ALF-based protocols is the common data naming between the transmission protocol and application (e.g., [20]), which we follow in the design of our meta-data. We take ALF-like ideas one step further by arguing that *routing* decisions are also best made in application-controlled and application-specific ways, using knowledge of not just network topology but application data layout and the state of resources at each node. We believe that such integrated approaches to naming and routing are attractive to a large range of network situations, especially in mobile and wireless networks of devices and sensors.

This section presents the individual elements that make up the SPIN family of protocols and presents two SPIN protocols that we have designed, SPIN-1 and SPIN-2.

2.1 Meta-Data

Sensors use meta-data to succinctly and completely describe the data that they collect. If x is the meta-data descriptor for sensor data X, then the size of x in bytes must be shorter than the size of X, for SPIN to be beneficial. If two pieces of actual data are distinguishable, then their corresponding meta-data should be distinguishable. Likewise, two pieces of indistinguishable data should share the same meta-data representation.

SPIN does not specify a format for meta-data; this format is application-specific. Sensors that cover disjoint geographic regions may simply use their own unique IDs as meta-data. The meta-data x would then stand for "all the data gathered by sensor x". A camera sensor, in contrast, might use (x, y, ϕ) as meta-data, where (x, y) is a geographic coordinate and ϕ is an orientation. Because each application's meta-data format may be different, SPIN relies on each application to interpret and synthesize its own meta-data. There are costs associated with the storage, retrieval, and general management of meta-data, but the benefit of having a succinct representation for large data messages in SPIN far outweighs these costs.

2.2 SPIN Messages

SPIN nodes use three types of messages to communicate:

- ADV – new data advertisement. When a SPIN node has data to share, it can advertise this fact by transmitting an ADV message containing meta-data.
- REQ – request for data. A SPIN node sends an REQ message when it wishes to receive some actual data.
- DATA – data message. DATA messages contain actual sensor data with a meta-data header.

Because ADV and REQ messages contain only meta-data, they are smaller, and cheaper to send and receive, than their corresponding DATA messages.

2.3 SPIN Resource Management

SPIN applications are resource-aware and resource-adaptive. They can poll their system resources to find out how much energy is available to them. They can also calculate the cost, in terms of energy, of performing computations and sending and receiving data over the network. With this information, SPIN nodes can make informed decisions about using their resources effectively. SPIN does not specify a particular energy management policy for its protocols. Rather, it specifies an interface that applications can use to probe their available resources.

2.4 SPIN Implementation

SPIN is an application-level approach to network communication. We therefore intend to implement SPIN as middleware application libraries with a well defined API. These libraries will implement the basic SPIN message types, message handling routines, and resource-management functions. Sensor applications can then use these libraries to construct their own SPIN protocols.

2.5 SPIN-1: A 3-Stage Handshake Protocol

The SPIN-1 protocol is a simple handshake protocol for disseminating data through a lossless network. It works in three stages (ADV-REQ-DATA), with each stage corresponding to one of the messages described above. The protocol starts when a node obtains new data that it is willing to disseminate. It does this by sending an ADV message to its neighbors, naming the new data (ADV stage). Upon receiving an ADV, the neighboring node checks to see whether it has already received or requested the advertised data. If not, it responds by sending an REQ message for the missing data back to the sender (REQ stage). The protocol completes when the initiator of the protocol responds to the REQ with a DATA message, containing the missing data (DATA stage).

Figure 3 shows an example of the protocol. Upon receiving an ADV packet from node A, node B checks to see whether it possesses all of the advertised data (a). If not, node B sends an REQ message back to A, listing all of the data that it would like to acquire (b). When node A receives the REQ packet, it retrieves the requested data and sends it back to node B as a DATA message (c). Node B, in turn, sends ADV messages advertising the new data it received from node A to all of its neighbors (d). It does not send an advertisement back to node A, because it knows that node A already has the data. These nodes then send advertisements of the new data to all of their neighbors, and the protocol continues.

There are several important things to note about this example. First, if node B had its own data, it could aggregate this with the data of node A and send advertisements of the aggregated data to all of its neighbors (d). Second, nodes are not required to respond to every message in the

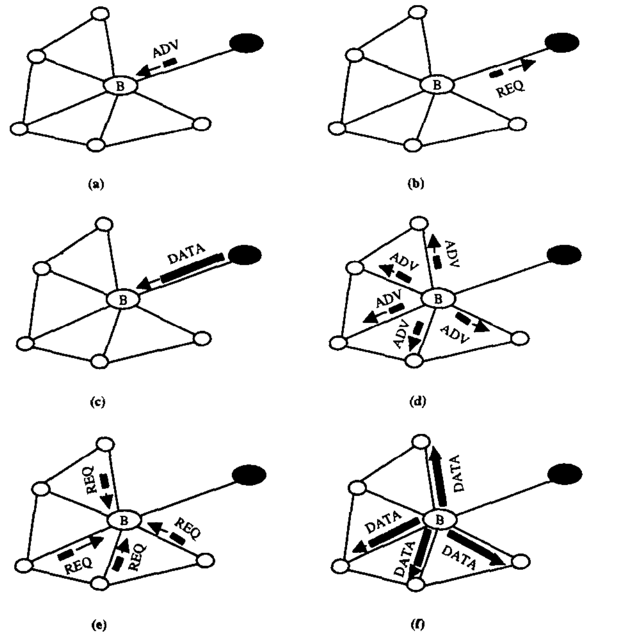

Figure 3: The SPIN-1 Protocol. Node A starts by advertising its data to node B (a). Node B responds by sending a request to node A (b). After receiving the requested data (c), node B then sends out advertisements to its neighbors (d), who in turn send requests back to B (e,f).

protocol. In this example, one neighbor does not send an REQ packet back to node B (e). This would occur if that node already possessed the data being advertised.

Though this protocol has been designed for lossless networks, it can easily be adapted to work in lossy or mobile networks. Here, nodes could compensate for lost ADV messages by re-advertising these messages periodically. Nodes can compensate for lost REQ and DATA messages by re-requesting data items that do not arrive within a fixed time period. For mobile networks, changes in the local topology can trigger updates to a node's neighbor list. If a node notices that its neighbor list has changed, it can spontaneously re-advertise all of its data.

This protocol's strength is its simplicity. Each node in the network performs little decision making when it receives new data, and therefore wastes little energy in computation. Furthermore, each node only needs to know about its single-hop network neighbors. The fact that no other topology information is required to run the algorithm has some important consequences. First, SPIN-1 can be run in a completely unconfigured network with a small, startup cost to determine nearest neighbors. Second, if the topology of the network changes frequently, these changes only have to travel one hop before the nodes can continue running the algorithm.

2.6 SPIN-2: SPIN-1 with a Low-Energy Threshold

The SPIN-2 protocol adds a simple energy-conservation heuristic to the SPIN-1 protocol. When energy is plentiful, SPIN-2 nodes communicate using the same 3-stage protocol as SPIN-1 nodes. When a SPIN-2 node observes that its energy is approaching a low-energy threshold, it adapts by reducing its participation in the protocol. In general, a node will only participate in a stage of the protocol if it believes that it can complete all the other stages of the protocol without going below the low-energy threshold. This conservative approach implies that if a node receives some new data, it only initiates the three-stage protocol if it believes it has enough energy to participate in the full protocol with all of its neighbors. Similarly, if a node receives an advertisement, it does not send out a request if it does not have enough energy to transmit the request and receive the corresponding data. This approach does not prevent a node from receiving, and therefore expending energy on, ADV or REQ messages below its low-energy threshold. It does, however, prevent the node from ever handling a DATA message below this threshold.

3 Other Data Dissemination Algorithms

In this section, we describe the three dissemination algorithms against which we will compare the performance of SPIN.

3.1 Classic Flooding

In classic flooding, a node wishing to disseminate a piece of data across the network starts by sending a copy of this data to all of its neighbors. Whenever a node receives new data, it makes copies of the data and sends the data to all of its neighbors, except the node from which it just received the data. The amount of time it takes a group of nodes to receive some data and then forward that data on to their neighbors is called a *round*. The algorithm finishes, or *converges*, when all the nodes in the network have received a copy of the data. Flooding converges in $O(d)$ rounds, where d is the diameter of the network, because it takes at most d rounds for a piece of data to travel from one end of the network to the other.

Although flooding exhibits the same appealing simplicity as SPIN-1, it does not solve either the implosion or the overlap problem.

3.2 Gossiping

Gossiping [9] is an alternative to the classic flooding approach that uses *randomization* to conserve energy. Instead of indiscriminately forwarding data to all its neighbors, a gossiping node only forwards data on to one randomly selected neighbor. If a gossiping node receives data from a given neighbor, it can forward data back to that neighbor if it randomly selects that neighbor. Figure 4 illustrates the reason that gossiping nodes forward data back to the sender. If node D never forwarded the data back to node B, node C would never receive the data.

Whenever data travels to a node with high degree in a classic flooding network, more copies of the data start floating around the network. At some point, however, these copies may end up imploding. Gossiping avoids such implosion because it only makes one copy of each message at any node. The fewer copies made, the lower the likelihood that any of these copies will ever implode.

While gossiping distributes information slowly, it dissipates energy at a slow rate as well. Consider the case where a single data source disseminates data using gossiping. Since the source sends to only one of its neighbors, and that neighbor sends to only one of its neighbors, the fastest rate at which gossiping distributes data is 1 node/round. Thus, if there are c data sources in the network, gossiping's fastest possible distribution rate is c nodes/round.

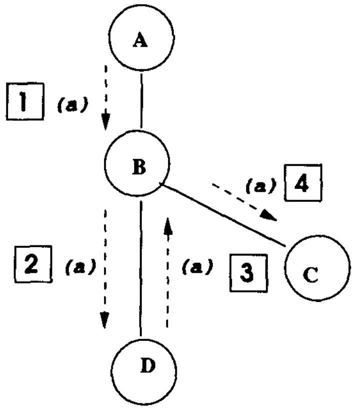

Figure 4: Gossiping. At every step, each node only forwards data on to one neighbor, which it selects randomly. After node D receives the data, it must forward the data *back* to the sender (B), otherwise the data would never reach node C.

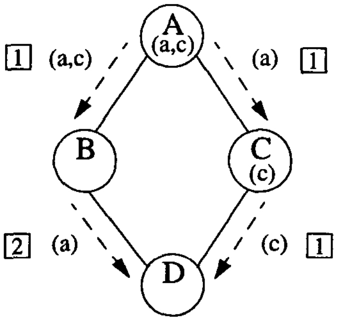

Figure 5: Ideal dissemination of observed data a and c. Each node in the figure is marked with its initial data, and boxed numbers represent the order in which data is disseminated in the network. In ideal dissemination, both implosion, caused by B and C's common neighbor, and overlap, caused by A and C's overlapping initial data item, c, do not occur.

Finally, we note that, although gossiping largely avoids implosion, it does not solve the overlap problem.

3.3 Ideal Dissemination

Figure 5 depicts an example network where every node sends observed data along a shortest-path route and every node receives each piece of distinct data only once. We call this *ideal dissemination* because observed data a and c arrive at each node in the shortest possible amount of time. No energy is ever wasted transmitting and receiving useless data.

Current networking solutions offer several possible approaches for dissemination using shortest-paths. One such approach is network-level multicast, such as IP multicast [5]. In this approach, the nodes in the network build and maintain distributed source-specific shortest-path trees and themselves act as multicast routers. To disseminate a new piece of data to all the other nodes in the network, a source would send the data to the network multicast group, thus ensuring that the data would reach all of the participants along shortest-path routes. In order to handle losses, the dissemination protocol would be modified to use reliable multicast. Unfortunately, multicast and particularly reliable multicast both rely upon complicated protocol machinery, much of which may be unnecessary for solving the specific problem of data dissemination in a sensor network. In many respects, SPIN may in fact be viewed as a form of *application-level multicasting*, where information about both the topology and data layout are incorporated into the distributed multicast trees.

Since most existing approaches to shortest-path distribution trees would have to be modified to achieve ideal dissemination, we will concentrate on comparing SPIN to the results of an ideal dissemination protocol, rather than its implementation. It turns out that we can simulate the results of an ideal dissemination protocol using a modified version of SPIN-1. We arrive at this simulation approach by noticing that if we trace the message history of the SPIN-1 protocol in a network, the DATA messages in the network would match the history of an ideal dissemination protocol. Therefore, to simulate an ideal dissemination protocol, we run the SPIN-1 protocol and eliminate any time and energy costs that ADV and REQ messages incur.

4 Sensor Network Simulations

In order to compare the different communication approaches discussed in the previous sections, we developed a sensor network simulator by extending the functionality of the ns software package. Using this simulation framework, we compared SPIN-1 and SPIN-2 with classic flooding and gossiping and the ideal data distribution protocol. We found that SPIN-1 provides higher throughput than gossiping and the same order of throughput as flooding, while at the same time uses substantially less energy than both these protocols. SPIN-2 is able to deliver even more data per unit energy than SPIN-1 and close to the ideal amount of data per unit energy by adapting to the limited energy of the network. We found that in all of our simulations, nodes with a higher degree tended to dissipate more energy than nodes with a lower degree, creating potential weak points in a battery-operated network.

4.1 ns Implementation

ns [15] is an event-driven network simulator with extensive support for simulation of TCP, routing, and multicast protocols. To implement the SPIN family of data distribution protocols, we added several features to the ns simulator. The ns Node class was extended to create a Resource-Adaptive Node, as shown in Figure 6. The major components of a Resource-Adaptive Node are the Resources, the Resource Manager, the Resource-Constrained Application (RCApplication), the Resource-Constrained Agent (RCAgent) and the Network Interface. The Resource Manager provides a common interface between the application and the individual resources. The RCApplication, a subclass of ns's Application class, is responsible for updating the status of the node's resources through the Resource Manager. In addition, the RCApplication implements the SPIN communication protocol and the resource-adaptive decision-making algorithms. The RCAgent packetizes the data generated by the RCApplication and sends the packets to the Node's Network Interface for transmission to one of the node's neighbors.

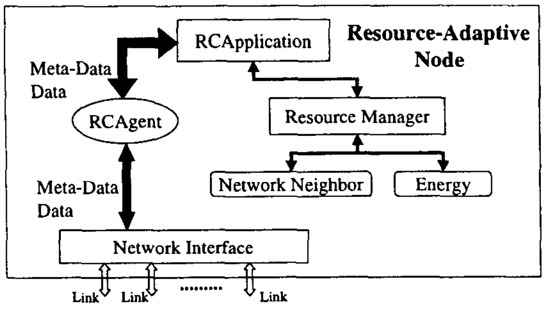

Figure 6: Block diagram of a Resource-Adaptive Node.

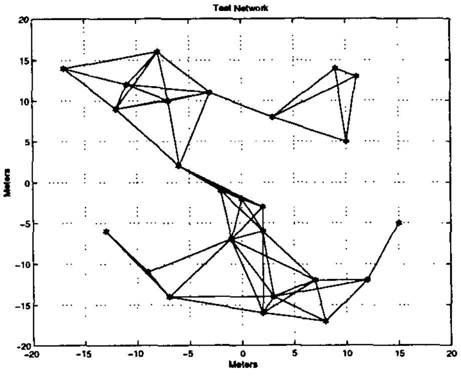

Figure 7: Topology of the 25-node, wireless test network. The edges shown here signify communicating neighbors.

4.2 Simulation Testbed

For our experiments, we created the 25-node network shown in Figure 7. This network, which was randomly generated with the constraint that the graph be fully connected, has 59 edges, a degree of 4.7, a hop diameter of 8, and an average shortest path of 3.2 hops. The power of the sensor radio transmitter is set so that any node within a 10 meter radius is within communication range and is called a neighbor of the sensor. The radio speed (1 Mbps) and the power dissipation (600 mW in transmit mode, 200 mW in receive mode) were chosen based on data from currently available radios. The processing delay for transmitting a message is randomly chosen between 5 ms and 10 ms[1]. We initialized each node with 3 data items, chosen randomly from a set of 25 possible data items. This means there is overlap in the initial data of different sensors, as often occurs in sensor networks. The size of each data item was set to 500 bytes, and we gave each item a distinct, 16 byte, meta-data name. Our test network assumes no network losses and no queuing delays. Table 1 summarizes these network characteristics.

Using this network configuration, we ran each protocol and tracked its progress in terms of the rate of data distribution and energy usage. For each experiment, we ran the protocols 10 times and averaged the data distribution times and energy usage to account for the random processing delay. The results of these experiments are presented in the following sections.

Nodes	25
Edges	59
Average degree	4.7 neighbors
Diameter	8 hops
Average shortest path	3.2 hops
Antenna reach	10 m
Radio propagation delay	$3\text{x}10^8$ m/s
Processing delay	5-10 ms
Radio speed	1 Mbps
Transmit cost	600 mW
Receive cost	200 mW
Data size	500 bytes
Meta-data size	16 bytes
Network losses	None
Queuing delays	None

Table 1: Characteristics of the 25-node wireless test network.

4.3 Unlimited Energy Simulations

For the first experiment, we gave all the nodes a virtually infinite supply of energy and ran each data distribution protocol until it converged. Since energy is not limited, SPIN-1 and SPIN-2 are identical protocols. Therefore, the results in this section only compare SPIN-1 with flooding, gossiping, and the ideal data distribution protocol.

4.3.1 Data Acquired Over Time

Figure 8 shows the amount of data acquired by the network over time for each of the protocols. These graphs clearly show that gossiping has the slowest rate of convergence. However, it is interesting to note that using gossiping, the system has acquired over 85% of the total data in a small amount of time; the majority of the time is spent distributing the last 15% of the data to the nodes. This is because a gossiping node sends all of the data it has to a randomly chosen neighbor. As the nodes obtain a large amount of data, this transmission will be costly, and, since it is very likely that the neighbor already has a large proportion of the data which is being transmitted, it will also be very wasteful. A gossiping protocol which kept some per-neighbor state, such as having each node keep track of the data it has already sent to each of its neighbors, would perform much better by reducing the amount of wasteful transmissions.

Figure 8 shows that SPIN-1 takes 80 ms longer to converge than flooding, whereas flooding takes only 10 ms longer to converge than ideal. Although it appears that SPIN-1 performs much worse than flooding in convergence time, this increase is actually a constant amount, regardless of the length of the simulation. Thus for longer simulations, the increase in convergence time for the SPIN-1 protocol will be negligible. The reasons for this behavior will be discussed in detail in Section 4.5.

Our experimental results showed that the data distribution curves were convex for all four protocols. We therefore speculated that these curves might generally be convex, regardless of the network topology. If we could predict the shape of these curves, we might be able to gain some intuition about the behavior of the protocols for different network topologies. To do this, we noted that the amount of data received by a node i at each round d depends only on the number of neighbors d hops away from this node, $n_i(d)$.

[1] Note that these simulations do not account for any delay caused by accessing, comparing, and managing meta-data.

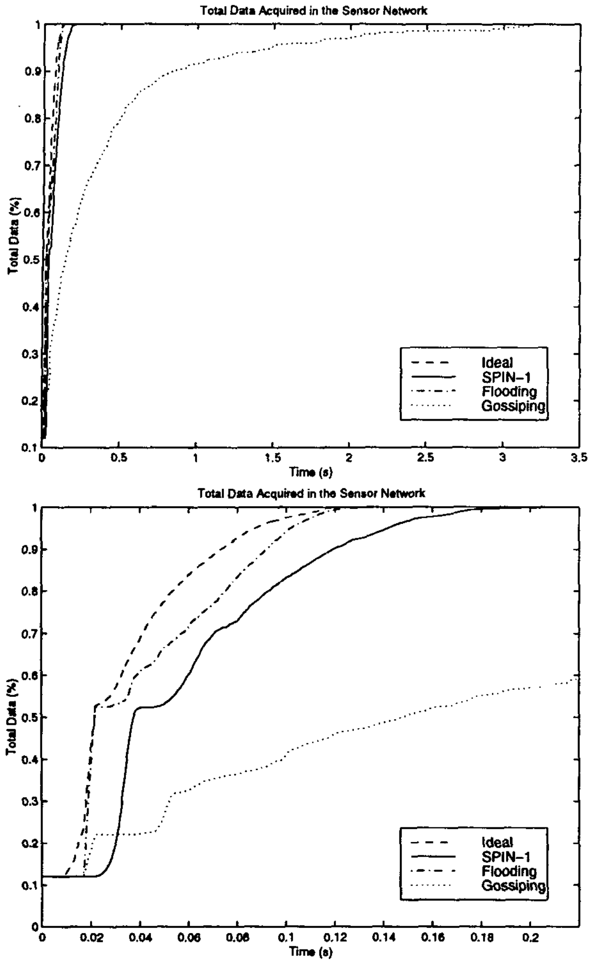

Figure 8: Percent of total data acquired in the system over time for each protocol. (a) shows the entire time scale until all the protocols converge. (b) shows a blow-up of the first 0.22 seconds.

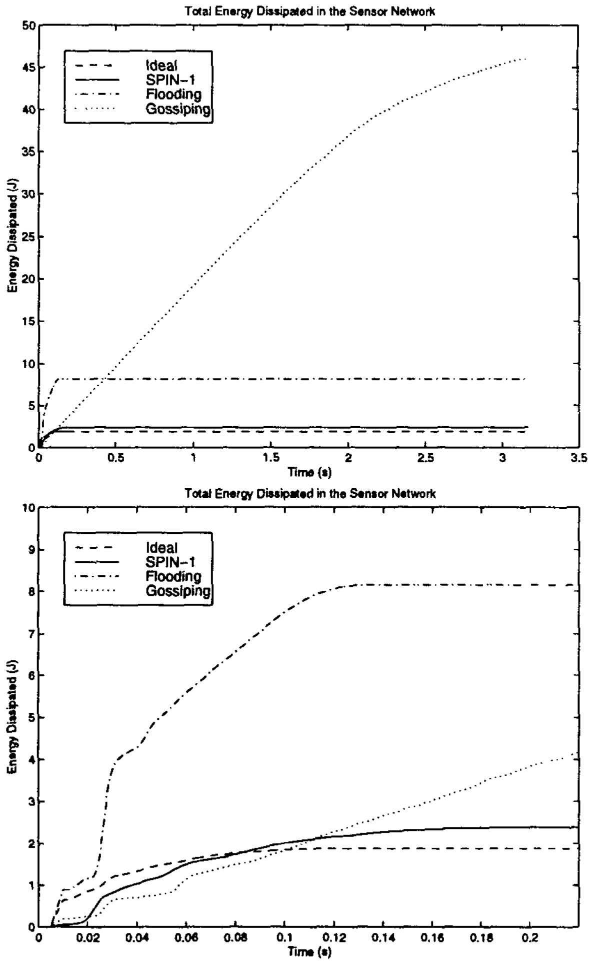

Figure 9: Total amount of energy dissipated in the system for each protocol. (a) shows the entire time scale until all the protocols converge. (b) shows a blow-up of the first 0.22 seconds.

However, since $n_i(d)$ is different for each node i and each distance d and is entirely dependent on the specific topology, we found that, in fact, no general conclusions can be drawn about the shape of these curves.

4.3.2 Energy Dissipated Over Time

For the previous experiment, we also measured the energy dissipated by the network over time, as shown in Figure 9.

These graphs show that gossiping again is the most costly protocol; it requires much more energy than the other two protocols to accomplish the same task. As stated before, adding a small amount of state to the gossiping protocol will dramatically reduce the total system energy usage.

Figure 9 also shows that SPIN-1 uses approximately a factor of 3.5 less energy than flooding. Thus, by sacrificing a small, constant offset in convergence time, SPIN-1 achieves a dramatic reduction in system energy. SPIN-1 is able to achieve this large reduction in energy since there is no wasted transmission of the large 500-byte data items.

We can see this advantage of the SPIN-1 protocol by looking at the message profiles for the different protocols, shown in Figure 10. The first three bars for each protocol show the number of data items transmitted throughout the network, the number of these data items that are redundant and thus represent wasteful transmission, and the number of data items that are useful. The number of useful data transmissions is the same for each protocol since the data distribution is complete once every node has all the data. The last three bars for each protocol show the number of meta-data items transmitted and the number of these items that are redundant and useful. These bars have a height zero for ideal, flooding, and gossiping, since these protocols do not use meta-data transmissions. Note that the number of useful meta-data transmissions for the SPIN-1 protocol is three times the number of useful data transmissions, since each data transmission in the SPIN-1 protocol requires three messages with meta-data.

Flooding and gossiping nodes send out many more data items than SPIN-1 nodes. Furthermore, 77% of these data items are redundant for flooding and 96% of the data items are redundant for gossiping, and these redundant messages

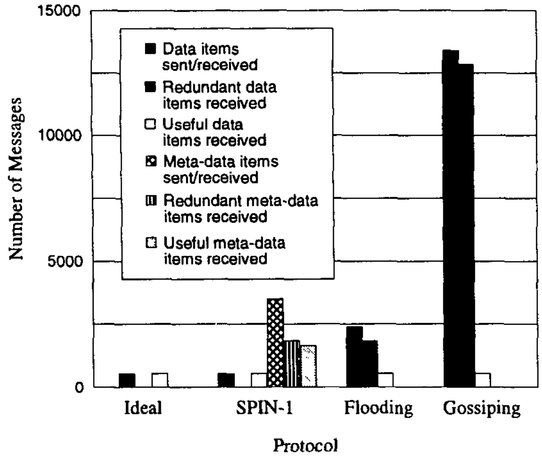

Figure 10: Message profiles for the simulations. Notice that SPIN-1 does not send any redundant data messages.

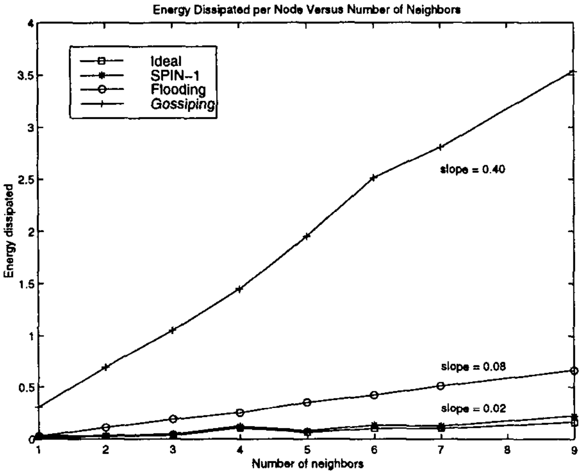

Figure 11: Energy dissipation versus node degree.

Performance *Relative to Ideal	Protocol		
	SPIN-1	Flooding	Gossiping
Increase in Energy Dissipation*	0.45 J	6.3 J	44.1 J
Increase in Convergence Time*	90 ms	10 ms	3025 ms
Slope of Energy Dissipation vs. Node Degree Correlation Line*	1.25x	5x	25x
% of Total Data Messages that are Redundant	0	77%	96%

Table 2: Key results of the unlimited energy simulations for the SPIN-1, flooding, and gossiping protocols compared with the ideal data distribution protocol.

come at the high cost of 500 bytes each. SPIN-1 nodes also send out a large number of redundant messages (53%); however, these redundant messages are meta-data messages. Meta-data messages come at a relatively low cost and come with an important benefit: *meta-data negotiation keeps SPIN-1 nodes from sending out even a single redundant data-item.*

We plotted the average energy dissipated for each node of a certain degree, as shown in Figure 11. This figure shows that for all the protocols, the energy dissipated at each node depends upon its degree. The repercussions of this finding is that if a high-degree node happens to lie upon a critical path in the network, it may die out before other nodes and partition the network. We believe that handling such situations is an important area for improvement in all four protocols.

The key results from these unlimited energy simulations are summarized in Table 2.

4.4 Limited Energy Simulations

For this experiment, we limited the total energy in the system to 1.6 Joules to determine how effectively each protocol uses its available energy. Figure 12 shows the data acquisition rate for the SPIN-1, SPIN-2, flooding, gossiping, and ideal protocols. This figure shows that SPIN-2 puts its available energy to best use and comes close to distributing the same amount of data as the ideal protocol. SPIN-2 is able to distribute 73% of the total data as compared with the ideal protocol which distributes 85%. We note that SPIN-1 distributes 68%, flooding distributes 53%, and gossiping distributes only 38%.

Figure 13 shows the rate of energy dissipation for this experiment. This plot shows that flooding uses all its energy very quickly, whereas gossiping, SPIN-1, and SPIN-2 use the energy at a slower rate and thus are able to remain operational for a longer period of time.

Figure 14 shows the number of data items acquired per unit energy for each of the protocols. If the system energy is limited to below 0.2 Joules, none of the protocols has enough energy to distribute any data. With 0.2 Joules, the gossiping protocol is able to distribute a small amount of data; with 0.5 Joules, the SPIN protocols begins to distribute data; and with 1.1 Joules, the flooding protocol begins to distribute the data. This shows that if the energy is very limited, the gossiping protocol can accomplish the most data distribution. However, if there is enough energy to get the flooding or one of the SPIN protocols started, these protocols deliver much more data per unit energy than gossiping. This graph also shows the advantage of SPIN-2 over SPIN-1, which doesn't base any decisions on the current level of its resources. By making the communication decisions based on the current level of the energy available to each node, SPIN-2 is able to distribute 10% more data per unit energy than SPIN-1 and 60% more data per unit energy than flooding.

4.5 Best-Case Convergence Times

In many cases, we are less concerned with the behavior of the protocols over time than the overall time at which the protocols converge. To study this behavior, we set up a series of experiments where we measured the effects of various network parameters on the convergence times of the protocols. As with the previous experiments, these experiments and the ensuing analysis do not account for queuing delays or network losses and are thus the best-case scenarios for real networks.

Figures 15 - 17 show the change in convergence time

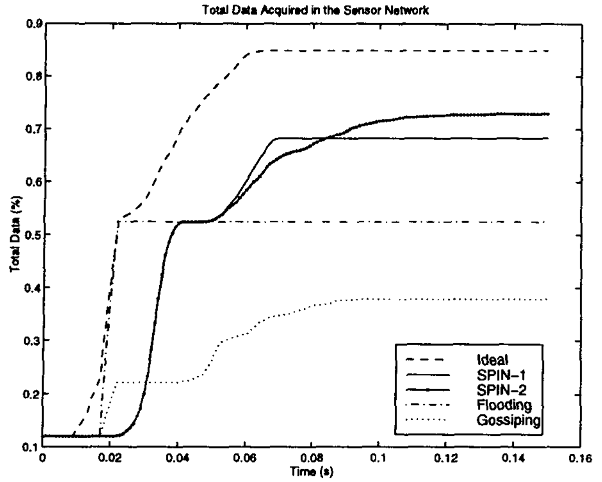

Figure 12: Percent of total data acquired in the system for each protocol when the total system energy is limited to 1.6 Joules.

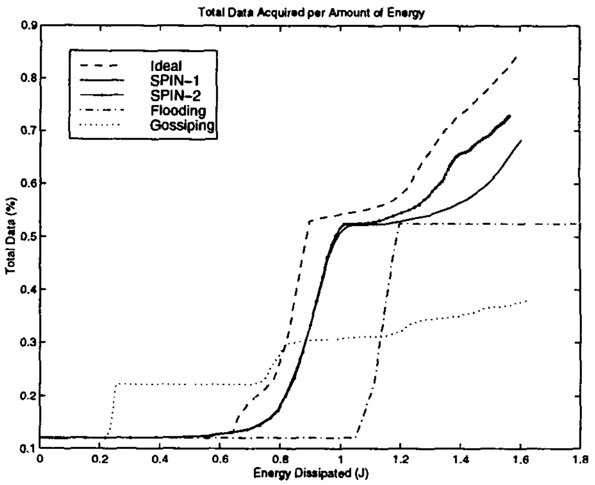

Figure 14: Data acquired for a given amount of energy. SPIN-2 distributes 10% more data per unit energy than SPIN-1 and 60% more data per unit energy than flooding.

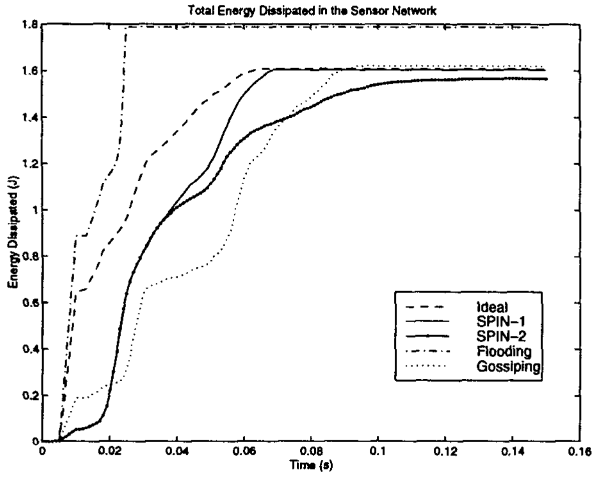

Figure 13: Energy dissipated in the system for each protocol when the total system energy is limited to 1.6 Joules.

for flooding, SPIN-1, and ideal as the parameters b (link bandwidth), d (fixed processing delay), and s (data size) are varied for the scenarios: (1) each sensor begins with a single unique data item and (2) each sensor begins with three pieces of overlapping data. The circles on the top graphs and the stars on the bottom graphs denote the conditions used in all our previous experiments (b = 1 Mbps, d = 5 ms, s = 500 bytes).

The convergence time for ideal and flooding are the same when there is no overlap in the initial data. Note that in the non-overlapping case, there is no set of parameters that gives SPIN-1 a smaller convergence time than flooding. However, for the overlapping initial data case, there are cross-overs as the bandwidth of the link and the size of each data item are varied.

To understand these results, we develop equations that predict the convergence time of each of these protocols. For all three protocols, the longest path any piece of data will need to traverse is the maximum shortest path of the network, or the network diameter, l_d. The transmission time over a single link of bandwidth b bits per second for a data message of size s bytes is $8s/b$. The transmission time for ADV and REQ messages is negligible compared with the transmission time for the DATA messages and will be ignored here. In addition, the network imposes a fixed d ms and a random [0-r] ms processing delay before any message (e.g., ADV, REQ, or DATA) is transmitted. This means that the convergence time for the ideal and flooding protocols are:

$$l_d(d+\frac{8s}{b}) \le C_{Ideal}, C_{Flood} \le l_d(d+r+\frac{8s}{b}) \qquad (1)$$

The minimum convergence time would occur if the random delay was always zero and the maximum convergence time would occur if the random delay was always the maximum possible value. A typical convergence time would be in the middle of these two bounds.

A similar analysis can be done for the SPIN-1 protocol. Once again, the longest path any piece of data will need to traverse is l_d. However, the delay incurred to get the data from one node to the next will be $3(d+r)+8s/b$, since each message (ADV, REQ, and DATA) incurs a processing delay of $(d+r)$ ms. This means SPIN-1 has the convergence bounds:

$$l_d(3d+\frac{8s}{b}) \le C_{SPIN-1} \le l_d(3(d+r)+\frac{8s}{b}) \qquad (2)$$

Therefore, there will always be an offset of between $2l_d d$ and $2l_d(d+r)$ between the convergence time of SPIN-1 and flooding (or ideal) for the case when there is no overlap in the initial data of each node and there are no queuing delays; there is no choice of network parameters for which SPIN-1 will converge before flooding for this scenario. However, the difference between convergence times will be a constant and thus be negligible for long simulations.

The analysis changes slightly for the case where there is overlap in the initial data and each node begins with $k > 1$ pieces of data. To begin with, the length of the longest path which a piece of data must traverse in this scenario is not necessarily the maximum shortest path of the network. Rather, this length l_{lp} will depend on the layout of the network and the initial distribution of the data. In addition, the size of each data message being transmitted can range

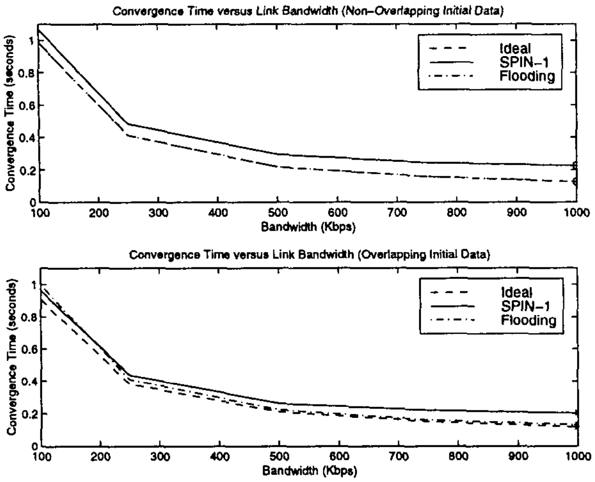

Figure 15: Convergence time as the link bandwidth is varied between 100 Kbps and 1 Mbps. The fixed processing delay is set to 5 ms and the data size is set to 500 bytes. (a) Each node begins with a single piece of unique data. (b) Each node begins with 3 pieces of non-unique data.

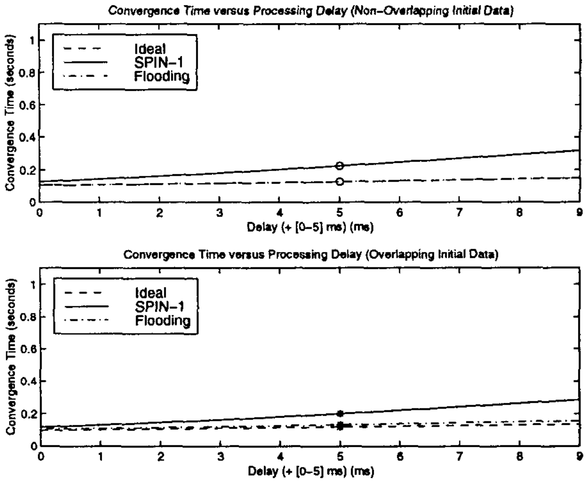

Figure 16: Convergence time as the fixed portion of the processing delay is varied between 1 ms and 9 ms. The link bandwidth is set to 1 Mbps and the data size is set to 500 bytes. (a) Each node begins with a single piece of unique data. (b) Each node begins with 3 pieces of non-unique data.

from s to ks bytes. For example, initially a node A could send all k pieces of its data to its neighbor B. These messages will be ks bytes long. However, the k pieces of data node B receives from A might not all be new; therefore node B will only transmit $k - o$ of these data pieces to its neighbors, where $0 \leq o \leq k$ is the number of data items that A sent to B which B already had and thus has already transmitted to its neighbors. Therefore, the time to transmit a data message is between $8s/b$ and $k8s/b$, depending on the number of data items in the message, so the convergence bounds for flooding and ideal become:

$$l_{lp}(d + \frac{8s}{b}) \leq C'_{Ideal}, C'_{Flood} \leq l_{lp}(d + r + k\frac{8s}{b}) \quad (3)$$

Similarly, the convergence bounds for SPIN-1 become:

$$l_{lp}(3d + \frac{8s}{b}) \leq C'_{SPIN-1} \leq l_{lp}(3(d + r) + k\frac{8s}{b}) \quad (4)$$

However, SPIN-1 and ideal nodes will be much more likely to only send a small number of data items, since these nodes never send wasteful data. Therefore, the convergence time for the SPIN-1 and ideal protocols will most often be between the upper and lower bounds, whereas the convergence time for flooding will most likely be near the upper bound. If the lower bound of convergence for SPIN-1 is much less than the upper bound of convergence for flooding, there is a nonzero probability that SPIN-1 will converge before flooding. This occurs when:

$$l_{lp}(3d + \frac{8s}{b}) \ll l_{lp}(d + r + k\frac{8s}{b}) \quad (5)$$

$$d \ll (k-1)\frac{4s}{b} + \frac{r}{2}$$

This means that when there is a large amount of initial overlapping data, it is possible for SPIN-1 to converge before flooding since SPIN-1 will more often send smaller (and less costly) data messages than flooding.

Network diameter (hops)	l_d	8
Shortest path for overlapping initial data (hops)	l_{lp}	7
Fixed processing delay (s)	d	5x10^{-3}
Random processing delay (s)	r	5x10^{-3}
Number of initial overlapping data items	k	3
Data size (bytes)	s	500
Link bandwidth (bps)	b	1e6

Table 3: Network parameters used to calculate convergence bounds for flooding, SPIN-1, and ideal.

In summary, if each node begins with more than one piece of non-unique data, it is possible for SPIN-1 to converge before flooding. However, if the initial data is unique, SPIN-1 will never converge before flooding[2].

Our testbed network has the parameters shown in Table 3. Plugging these parameters into Eqns. 3 and 4 give the following convergence bounds for our network:

$$0.063 \leq C'_{Ideal}, C'_{Flood} \leq 0.154 \quad (6)$$

$$0.133 \leq C'_{SPIN-1} \leq 0.294 \quad (7)$$

The experimental results show that, on average, flooding converges in 135 ms, SPIN-1 converges in 215 ms, and ideal converges in 125 ms. Notice that the flooding convergence

[2] If each node begins with k pieces of data but the data are unique, it is the same as considering each node starting with one piece of unique data that is k times as large as a single piece of data and SPIN-1 will never converge before flooding. Similarly, if each node begins with one piece of non-unique data, there will never be a case where either protocol reduces the data message size and again SPIN-1 will never converge before flooding.

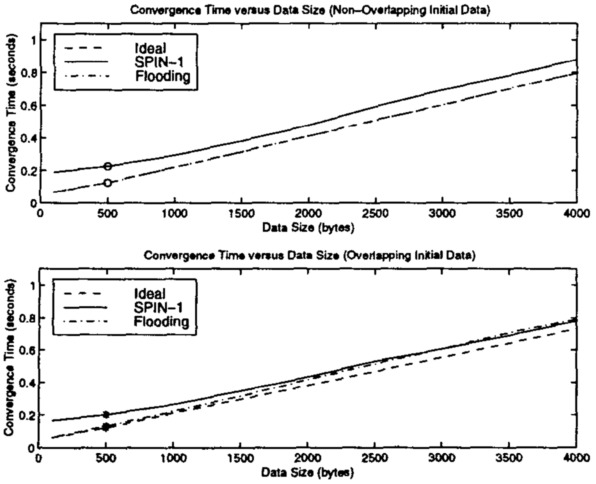

Figure 17: Convergence time as the size of a piece of data is varied between 100 bytes and 4000 bytes. The link bandwidth is set to 1 Mbps and the fixed processing delay is set to 5 ms. (a) Each node begins with a single piece of unique data. (b) Each node begins with 3 pieces of non-unique data.

time is close to the upper bound, whereas the SPIN-1 convergence time is in the middle of the two bounds, as agrees with our intuition that SPIN-1 sends less than $k = 3$ data items per message more often than flooding. As stated before, this increase in convergence time is constant for a given topology and will become negligible for longer simulations.

Once queuing delays are incorporated into our network testbed, the convergence time for flooding will be worse than the convergence time for ideal. In addition, we expect the convergence time for flooding to be worse than the convergence time for SPIN-1, even in the unique initial data case, due to the extraneous transmissions causing queuing delays in a flooding node that are not a problem in a SPIN-1 node.

5 Related Work

Perhaps the most fundamental use of dissemination protocols in networking is in the context of routing table dissemination. For example, nodes in link-state protocols (such as OSPF [14]) periodically disseminate their view of the network topology to their neighbors, as discussed in [10, 24]. Such protocols closely mimic the classic flooding protocol we described earlier.

There are generally two types of topologies used in wireless networks: centralized control and peer-to-peer communications [16]. The latter style is better suited for wireless sensor networks than the former, given the ad hoc, decentralized nature of such networks. Recently, mobile ad hoc routing protocols have become an active area of research [3, 11, 17, 19, 23]. While these protocols solve important problems, they are a different class of problems from the ones that arise in wireless sensor networks. In particular, we believe that sensor networks will benefit from application-controlled negotiation-based dissemination protocols, such as SPIN.

Routing protocols based on minimum-energy routing [12, 22] and other power-friendly algorithms have been proposed in the literature [13]. We believe that such protocols will be useful in wireless sensor networks, complementing SPIN and enabling better resource adaptation. Recent advances in operating system design [7] have made application-level approaches to resource adaptation, such as these, a viable alternative to more traditional approaches.

Using gossiping and broadcasting algorithms to disseminate information in distributed systems has been extensively explored in the literature, often as *epidemic algorithms* [6]. In [1, 6], gossiping is used to maintain database consistency, while in [18], gossiping is used as a mechanism to achieve fault tolerance. A theoretical analysis of gossiping is presented in [9]. Recently, such techniques have also been used for resource discovery in networks [8].

Perhaps closest in philosophy to the negotiation-based approach of SPIN is the popular Network News Transfer Protocol (NNTP) for Usenet news distribution on the Internet [2]. Here, news servers form neighborhoods and disseminate new information between each other, using names and timestamps as meta-data to negotiate data dissemination.

We also note that there has been a lot of recent interest in using IP multicast [5] as the underlying infrastructure to efficiently and reliably disseminate data from a source to many receivers [21] on the Internet. However, for the reasons described in Section 3, we believe that enabling applications to control routing decisions is a less complex and better approach for wireless sensor networks.

6 Conclusions

In this paper, we introduced SPIN (Sensor Protocols for Information via Negotiation), a family of data dissemination protocols for wireless sensor networks. SPIN uses meta-data negotiation and resource-adaptation to overcome several deficiencies in traditional dissemination approaches. Using meta-data names, nodes negotiate with each other about the data they possess. These negotiations ensure that nodes only transmit data when necessary and never waste energy on useless transmissions. Being resource-aware, nodes are able to cut back on their activities whenever their resources are low to increase their longevity.

We have discussed the details of two specific SPIN protocols, SPIN-1 and SPIN-2. SPIN-1 is a 3-stage handshake protocol for disseminating data, and SPIN-2 is a version of SPIN-1 that backs off from communication at a low-energy threshold. Finally, we compared the SPIN-1 and SPIN-2 protocols to flooding, gossiping, and ideal dissemination protocols using the *ns* simulation tool.

After examining SPIN in this paper, both qualitatively and quantitatively, we arrive at the following conclusions:

- Naming data using meta-data descriptors and negotiating data transmissions using meta-data successfully solve the implosion and overlap problems described in Section 1.
- SPIN-1 and SPIN-2 are simple protocols that efficiently disseminate data, while maintaining no per-neighbor state. These protocols are well-suited for an environment where the sensors are mobile because they base their forwarding decisions on local neighborhood information.
- In terms of time, SPIN-1 achieves comparable results to classic flooding protocols, and in some cases outperforms classic flooding. In terms of energy, SPIN-1 uses only about 25% as much energy as a classic flooding

protocol. SPIN-2 is able to distribute 60% more data per unit energy than flooding.

- In all of our experiments, SPIN-1 and SPIN-2 outperformed gossiping. They also come close to an ideal dissemination protocol in terms of both time and energy under some conditions.

In summary, SPIN protocols hold the promise of achieving high performance at a low cost in terms of complexity, energy, computation, and communication.

Although our initial work and results are promising, there is still a great deal of work to be done in this area. First and foremost, we would like to study SPIN protocols using more realistic wireless models. The loss-prone nature of wireless channels needs to be incorporated and experimented with in our framework, and we believe that this will not be difficult. Furthermore, SPIN-1 and SPIN-2 are currently targeted for a MAC-layer that does not support wireless broadcast. Such protocols, most notably the popular 802.11 MAC-layer protocol, do exist, and we would like to examine how SPIN protocols may be improved to take advantage of MAC-level broadcast. Finally, we would like to develop more sophisticated resource-adaptation protocols to use available energy well. In particular, we are interested in designing protocols that make adaptive decisions based not only on the cost of communicating data, but also the cost of synthesizing it. Such resource-adaptive approaches may hold the key to making compute-intensive sensor applications a reality in the future.

Acknowledgments

We are grateful to Wei Shi, who participated in the initial design and evaluation of some of the work in this paper, for his contributions. We thank Anantha Chandrakasan for his vision that pointed us in the direction of low-energy sensor networks and for his helpful comments and suggestions throughout this work. We also thank Suchitra Raman and John Wroclawski for several comments and suggestions that greatly improved the quality of this paper. This research was supported in part by a research grant from NTT Corporation and in part by the Advanced Research Projects Agency under contract DAAN02-98-K-0003. W. Heinzelman is supported by a Kodak Fellowship.

References

[1] AGRAWAL, D., ABBADI, A., AND STEINKE, R. Epidemic Algorithms in Replicated Databases. In *Proc. 16th ACM Principles of Database Systems* (May 1997).

[2] BORMANN, C. Network News Transport Protocol. Internet Draft, IETF, Nov. 1998. Work in progress.

[3] BROCH, J., MALTZ, D., JOHNSON, D., HU, Y., AND JETCHEVA, J. A Performance Comparison of Multi-Hop Wireless Ad Hoc Netowrk Routing Protocols. In *Proc. 4th ACM International Conference on Mobile Computing and Networking (Mobicom'98)* (Oct. 1998).

[4] CLARK, D., AND TENNENHOUSE, D. Architectural Consideration for a New Generation of Protocols. In *Proc. ACM SIGCOMM* (September 1990).

[5] DEERING, S., AND CHERITON, D. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems 8*, 2 (May 1990).

[6] DEMERS, A., GREENE, D., HAUSER, C., IRISH, W., AND LARSON, J. Epidemic Algorithms for Replicated Database Maintenance. In *ACM Principles of Distributed Computing* (Aug. 1987).

[7] ENGLER, D. R., KAASHOEK, M. F., AND O'TOOLE JR., J. Exokernel: An operating system architecture for application-level resource management. In *Proc. of the 15th ACM Symposium on Operating Systems Principles* (December 1995).

[8] HARCHOL-BALTER, M., LEIGHTON, T., AND LEWIN, D. Resource Discovery in Distributed Networks. In *ACM Symposium on Principles of Distributed Computing* (May 1999).

[9] HEDETNIEMI, S., HEDETNIEMI, S., AND LIESTMAN, A. A Survey of Gossiping and Broadcasting in Communication Networks. *Networks 18* (1988).

[10] HUITEMA, C. *Routing in the Internet.* Prentice Hall, 1996.

[11] JOHNSON, D. Routing in Ad Hoc Networks of Mobile Hosts. In *Proc. IEEE Workshop on Mobile Computing Systems and Applications* (Dec. 1994).

[12] KARN, P. Spectral Efficiency Considerations for Packet Radio. In *ARRL 10th Computer Networking Conf.* (1991).

[13] MENG, T., AND VOLKAN, R. Distributed Network Protocols for Wireless Communication. In *Proc. IEEEE ISCAS* (May 1998).

[14] MOY, J. OSPF Version 2, 1991. RFC 1583.

[15] ns-2 Network Simulator. http://www-mash.cs.berkeley.edu/ns/, 1998.

[16] PAHLAVAN, K., AND LEVESQUE, A. *Wireless Information Networks.* John Wiley & Sons, Inc., 1995.

[17] PARK, V., AND CORSON, S. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Netowrks. In *Proc. INFOCOM'97* (Apr. 1997).

[18] PELC, A. Fault-tolerant Broadcasting and Gossiping in Communication. *Networks 28*, 3 (Oct. 1996).

[19] PERKINS, C., AND BHAGWAT, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proc. Sigcomm '94 Conference on Communications Architectures, Protocols, and Applications* (Aug. 1994).

[20] RAMAN, S., AND MCCANNE, S. Scalable Data Naming for Application Level Framing in Reliable Multicast. In *Proc. ACM Multimedia* (Sept. 1998).

[21] Reliable Multicast Research Group. http://www.east.isi.edu/RMRG/, 1998.

[22] SHEPARD, T. A Channel Access Scheme for Large Dense Packet Radio Networks. In *Proc. ACM SIGCOMM* (Aug. 1998).

[23] SINHA, P., SIVAKUMAR, R., AND BHARGHAVAN, V. CEDAR: A Core-Extraction Distributed ad hoc Routing Algorithm. In *Proc. IEEE INFOCOM* (Mar. 1999).

[24] STEENSTRUP, M. *Routing in Communication Networks.* Prentice Hall, 1995.