Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks

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

Abstract

In this paper, we present a family of adaptive protocols, called SPIN (Sensor Protocols for Information via Negotiation), that efficiently disseminate 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 four specific SPIN protocols: SPIN-PP and SPIN-EC, which are optimized for a point-to-point network, and SPIN-BC and SPIN-RL, which are optimized for a broadcast network. Comparing the SPIN protocols to other possible approaches, we find that the SPIN protocols can deliver 60% more data for a given amount of energy than conventional approaches in a point-to-point network and 80% more data for a given amount of energy in a broadcast network. We also find that, in terms of dissemination rate and energy usage, the SPIN protocols perform close to the theoretical optimum in both point-to-point and broadcast networks.

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 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 wireless sensors have a limited supply of energy, energy-conserving forms of communication and computation are essential to wireless sensor networks.
- **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 designed SPIN to disseminate individual sensor observations to all sensors in a network, treating all

Submitted to *ACM Wireless Networks*; an earlier version of this paper appeared in ACM MOBICOM '99.

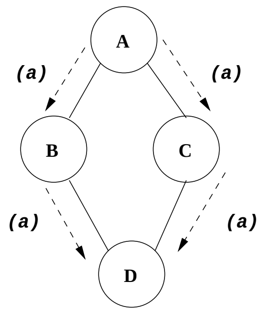

Figure 1: The implosion problem. In this graph, node A starts by flooding its data to all of its neighbors. Two copies of the data eventually arrive at node D. The system wastes energy and bandwidth in one unnecessary send and receive.

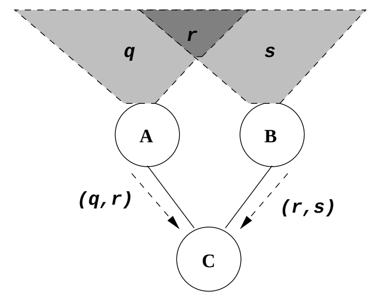

Figure 2: The overlap problem. Two sensors cover an overlapping geographic region. When these sensors flood their data to node C, C receives two copies of the data marked r.

sensors as potential sink nodes. SPIN thus provides a way of replicating complete views of the environment throughout an entire network.

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.

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 performed two studies of the SPIN approach based on two different wireless network models. In the first study, we examined five different protocols and their performance in a simple, point-to-point, wireless network where packets are never dropped and queueing delays never occur. Two of the protocols in this study are SPIN protocols (*SPIN-PP* and *SPIN-EC*). 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 in which each node has global knowledge of the status of all other nodes in the network, yielding the best possible performance. In the second study, we were interested in studying SPIN protocols in a more realistic wireless network model, where radios send packets over a single, unreliable, broadcast channel. *SPIN-BC* and *SPIN-RL* are two SPIN protocols that we designed specifically for such networks, and we compare them to two other protocols, flooding and ideal.

We evaluated each protocol under varying conditions by measuring the amount of data it transmitted and the amount of energy it used. 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-PP uses negotiation to solve the implosion and overlap problems; it reduces energy consumption by a factor of 3.6 compared to flooding, while disseminating data almost as quickly as theoretically possible. SPIN-EC, which additionally incorporates a threshold-based resource-awareness mechanism in addition to negotiation, disseminates 1.4 times 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. In a lossless, broadcast network with queueing delays, SPIN-BC reduces energy consumption by a factor of 1.6 and speeds up data dissemination by a factor of 1.8 compared to flooding. When the network loses packets, SPIN-RL is able to successfully recover from packet-losses, while still using half as much energy per unit data as flooding.

2 SPIN Philosophy and Overview

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. This section presents the individual features that make up the SPIN family of protocols.

2.1 Application-level Control

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., [21]), 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.

Because SPIN is an application-level approach to network communication, we 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.

2.2 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.

2.3 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.

2.4 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.

3 SPIN Protocols

In this section, we present four protocols that follow the SPIN philosophy outlined in the previous section. Two of the protocols, SPIN-PP and SPIN-BC, tackle the basic problem of data transmission under ideal conditions, where energy is plentiful and packets are never lost. SPIN-PP solves this problem for networks using point-to-point transmission media, and SPIN-BC solves this problem for networks using broadcast media. The other two protocols, SPIN-EC and SPIN-RL, are modified versions of the first two protocols, and they are meant to operate in networks that are not ideal. SPIN-EC, an energy-conserving version of SPIN-PP, reduces the number of messages it exchanges when energy in the system is low. SPIN-RL, a reliable version of SPIN-BC, recovers from losses in the network by selectively retransmitting messages.

3.1 SPIN-PP: A 3-Stage Handshake Protocol for Point-to-Point Media

The first SPIN protocol, SPIN-PP, is optimized for a networks using point-to-point transmission media, where it is possible for nodes A and B to communicate exclusively with each other without interfering with other nodes. In such a point-to-point wireless network, the cost of communicating with n neighbors in terms of time and energy is n times the cost of communicating with 1 neighbor. We start our study of SPIN protocols with a point-to-point network because of its relatively simple, linear cost model.

The SPIN-PP protocol works in three stages (ADV-REQ-DATA), with each stage corresponding to one of the messages described above. The protocol starts when a node advertises 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 (1). If not, node B sends an REQ message back to A, listing all of the data that it would like to acquire (2). When node A receives the REQ packet, it retrieves the requested data and sends it back to node B as a DATA message (3). Node B, in turn, sends ADV messages advertising the new data it received from node A to all of its neighbors (4). 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 (4). Second, nodes are not required to respond to every message in the protocol. In this example, one neighbor does not send an REQ packet back to node B (5). This would occur if that node already possessed the data being advertised.

Although 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. Nodes using the protocol make very simple decisions when they receive new data, and they therefore waste 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-PP 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.

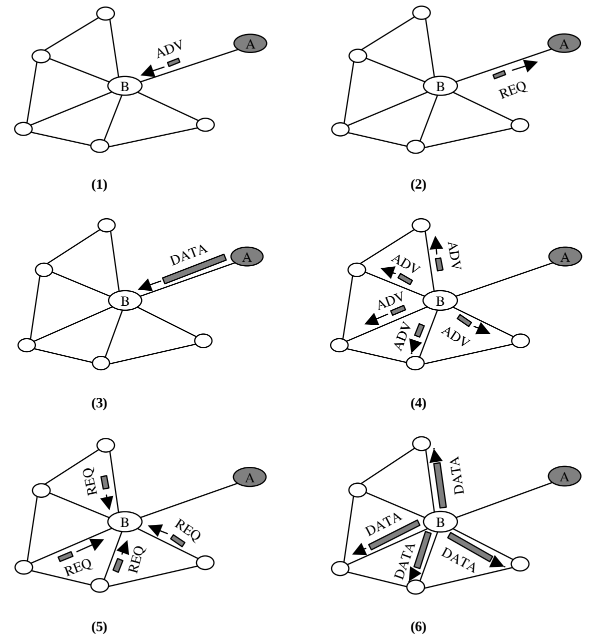

Figure 3: The SPIN-PP Protocol. Node A starts by advertising its data to node B (1). Node B responds by sending a request to node A (2). After receiving the requested data (3), node B then sends out advertisements to its neighbors (4), who in turn send requests back to B (5,6).

3.2 SPIN-EC: SPIN-PP with a Low-Energy Threshold

The SPIN-EC protocol adds a simple energy-conservation heuristic to the SPIN-PP protocol. When energy is plentiful, SPIN-EC nodes communicate using the same 3-stage protocol as SPIN-PP nodes. When a SPIN-EC 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.

3.3 SPIN-BC: A 3-Stage Handshake Protocol for Broadcast Media

In broadcast transmission media, nodes in the network communicate using a single, shared channel. As a result, when a node sends out a message in a broadcast network, it is received by every node within a certain range of the sender[1], regardless of the message's destination. If a node wishes to send a message and senses that the channel is currently in use, it must wait for the channel to become idle before

[1] This *transmission range* is determined by the power with which the sender transmitted the message and the sensitivity of the receiver.

attempting to send the message. The disadvantage of such networks is that whenever a node sends out a message, all nodes within transmission range of that node must pay a price for that transmission, in terms of both time and energy. However, the advantage of such networks is that, when a single node sends a message out to a broadcast address, this node can reach all of its neighbors using only one transmission. One-to-many communication is therefore $1/n$ times cheaper in a broadcast network than in a point-to-point network, where n is the number of neighbors for each node.

SPIN-BC improves upon SPIN-PP for broadcast networks by exclusively using cheap, one-to-many communication. This means that all messages are sent to the broadcast address and thus processed by all nodes that are within transmission range of the sender. We justify this approach by noting that, since broadcast and unicast transmissions use the same amount of network resources in a broadcast network, SPIN-BC does not lose much efficiency by using the broadcast address. Moreover, SPIN-BC nodes can coordinate their resource-conserving efforts more effectively because each node overhears all transactions that occur within its transmission range. For example, if two nodes A and B send requests for a piece of data to node C, C only needs to broadcast the requested data once in order to deliver the data to both A and B. Thus, only one node, either A or B, needs to send a request to C, and all other requests are redundant. If A and B address their requests directly to C, only C will hear the message, though all of the nodes within the transmission range of A and B will pay for two requests. However, if A and B address their requests to the broadcast address, all nodes within range will overhear these requests. Assuming that A and B are not perfectly synchronized, then either A will send its request first or B will. The node who does not send first will overhear the other node's request, realize that its own request is redundant, and suppress its own request. In this example, nodes that use the broadcast address can roughly halve their network resource consumption over nodes that do not. As we will illustrate shortly, this kind of approach, often called *broadcast message-suppression*, can be used to curtail the proliferation of redundant messages in the network.

Like the SPIN-PP protocol, the SPIN-BC protocol has an ADV, REQ, and DATA stage, which serve the same purpose as they do in SPIN-PP. There are three central differences between between SPIN-PP and SPIN-BC. First, as mentioned above, all SPIN-BC nodes send their messages to the broadcast address, so that all nodes within transmission range will receive the messages. Second, SPIN-BC nodes do not immediately send out requests when they hear advertisements for data they need. Upon receiving an ADV, each node checks to see whether it has already received or requested the advertised data. If not, it sets a random timer to expire, uniformly chosen from a predetermined interval. When the timer expires, the node sends an REQ message out to the broadcast address, specifying the original advertiser in the header of the message. When nodes other than the original advertiser receive the REQ, they cancel their own request timers, and prevent themselves from sending out redundant copies of the same request. The final difference between SPIN-PP and SPIN-BC is that a SPIN-PP node will send out the requested data to the broadcast address once and only once, as this is sufficient to get the data to all its neighbors (assuming a lossless network). It will not respond to multiple requests for the same piece of data.

Figure 4 shows an example of the protocol. Upon receiving an ADV packet from node A, A's neighbors check to see whether they have received the advertised data (1). Three of A's neighbors, C, D, and E, do not have A's data, and enter request suppression mode for different, random amounts of time. C's timer expires first, and C broadcasts a request for A's data (2), which in turn suppresses the duplicate request from D. Though several nodes receive the request, only A responds, because it is the originator of the original packet (3). After A sends out its data, E's request is suppressed, and C, D, and E all send out advertisements for their new data (4).

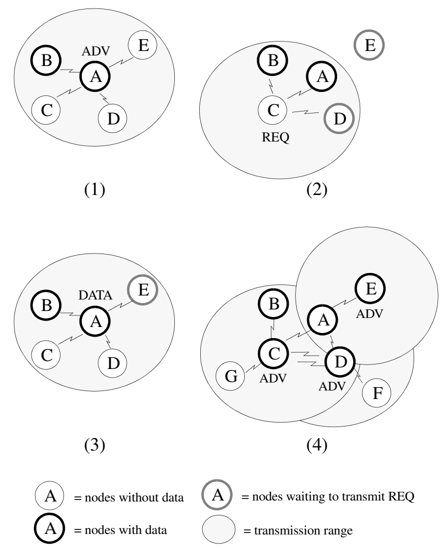

Figure 4: The SPIN-BC Protocol. Node A starts by advertising its data to all of its neighbors (1). Node C responds by broadcasting a request, specifying A as the originator of the advertisement (2), and suppressing the request from D. After receiving the requested data (3), E's request is also suppressed, and C, D, and E send advertisements out to their neighbors for the data that they received from A (4).

3.4 SPIN-RL: SPIN-BC for Lossy Networks

SPIN-RL, a reliable version of SPIN-BC, can disseminate data efficiently through a broadcast network, even if the network loses packets. The SPIN-RL protocol incorporates two adjustments to SPIN-BC to achieve reliability. First, each SPIN-RL node keeps track of which advertisements it hears from which nodes, and if it doesn't receive the data within a reasonable period of time following a request, the node re-requests the data. It fills out the originating-advertiser field in the header of the REQ message with a destination, randomly picked from the list of neighbors that had advertised that specific piece of data. Second, SPIN-RL nodes limit the frequency with which they will resend data. If a SPIN-RL node sends out a DATA message corresponding to a specific piece of data, it will wait a predetermined amount of time before responding to any more requests for that piece of data.

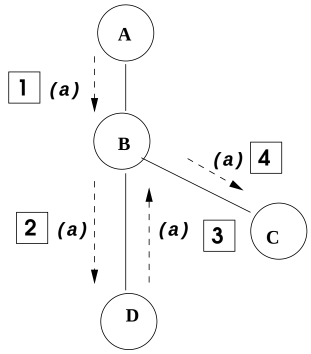

Figure 5: Gossiping. At every step, each node only forwards data on to one neighbor, which it selects randomly. After node D receives the data, it must forward the data *back* to the sender (B), otherwise the data would never reach node C.

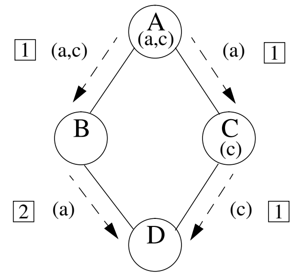

Figure 6: Ideal dissemination of observed data a and c. Each node in the figure is marked with its initial data, and boxed numbers represent the order in which data is disseminated in the network. In ideal dissemination, both implosion, caused by B and C's common neighbor, and overlap, caused by A and C's overlapping initial data item, c, do not occur.

4 Other Data Dissemination Algorithms

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

4.1 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-PP, it does not solve either the implosion or the overlap problem.

4.2 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 5 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.

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

4.3 Ideal Dissemination

Figure 6 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. For point-to-point networks, it turns out that we can simulate the results of an ideal dissemination protocol using a modified version of SPIN-PP. We arrive at this simulation approach by noticing that if we trace the message history of the SPIN-PP protocol in a network, the DATA messages in the network would match the history of an ideal dissemination protocol. Therefore, to simulate an

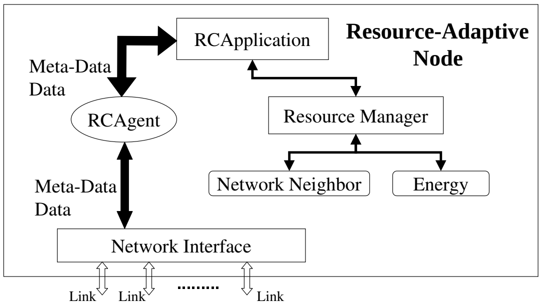

Figure 7: Block diagram of a Resource-Adaptive Node.

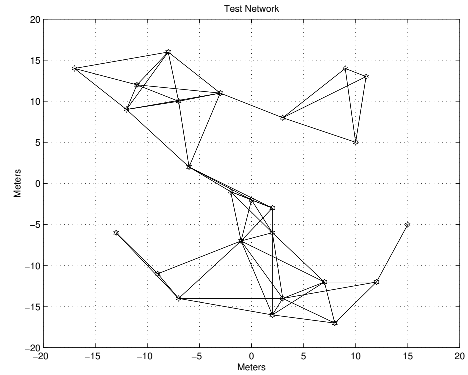

Figure 8: Topology of the 25-node, wireless test network. The edges shown here signify communicating neighbors in a point-to-point wireless medium.

ideal dissemination protocol for point-to-point networks, we run the SPIN-PP protocol and eliminate any time and energy costs that ADV and REQ messages incur. Defining an ideal protocol for broadcast networks is more tricky. We approximate an ideal dissemination protocol for broadcast networks by running the SPIN-BC protocol on a lossless network and eliminating any time and energy costs that ADV and REQ messages would incur.

5 Point-to-Point Media Simulations

In order to study the SPIN-PP and SPIN-EC 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-PP and SPIN-EC with classic flooding and gossiping and the ideal data distribution protocol. We found that SPIN-PP provides higher throughput than gossiping and the same order of throughput as flooding, while at the same time it uses substantially less energy than both these protocols. SPIN-EC is able to deliver even more data per unit energy than SPIN-PP 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.

5.1 `ns` Implementation

`ns` [16] is an event-driven network simulator with extensive support for simulation of TCP, routing, and multicast protocols. To implement the SPIN-PP and SPIN-EC 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 7. 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. For each point-to-point link that would exist between neighboring nodes in a wireless network, we created a wired link using `ns`'s built-in link support. We made these wired links appear to be wireless by forcing them to consume the same amount of time and energy that would accompany real, wireless link communications.

5.2 Simulation Testbed

For our simulations, we used the 25-node network shown in Figure 8. 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[2].

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 set of results, we ran the simulation 10 times and averaged the data distribution times and energy usage to account for the random processing delay. The results of these simulations are presented in the following sections.

5.3 Unlimited Energy Simulations

For the first set of simulations, we gave all the nodes a virtually infinite supply of energy and simulated each data distribution protocol until it converged. Since energy is not limited, SPIN-PP and SPIN-EC are identical protocols. There-

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

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	$3x10^8$ 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.

fore, the results in this section only compare SPIN-PP with flooding, gossiping, and the ideal data distribution protocol.

5.3.1 Data Acquired Over Time

Figure 9 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. Because 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 9 shows that SPIN-PP takes 80 ms longer to converge than flooding, whereas flooding takes only 10 ms longer to converge than ideal. Although it appears that SPIN-PP 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-PP protocol will be negligible.

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, $n_i(d)$. 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.

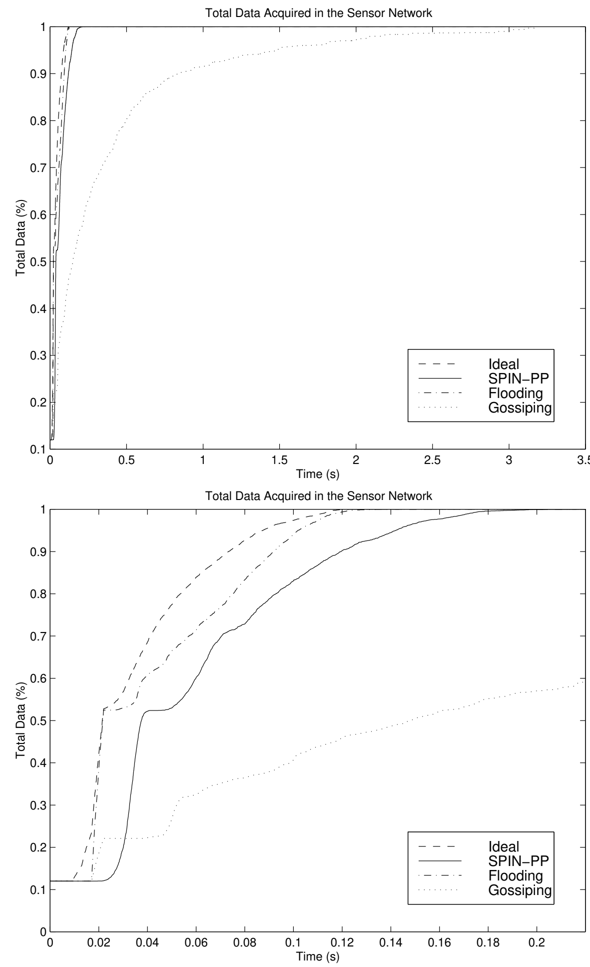

Figure 9: Percent of total data acquired in the system over time for each protocol. (a) shows the entire time scale until all the protocols converge. (b) shows a blow-up of the first 0.22 seconds.

5.3.2 Energy Dissipated Over Time

For the previous set of simulations, we also measured the energy dissipated by the network over time, as shown in Figure 10.

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 10 also shows that SPIN-PP uses approximately a factor of 3.5 less energy than flooding. Thus, by sacrificing a small, constant offset in convergence time, SPIN-PP achieves a dramatic reduction in system energy. SPIN-PP 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-PP protocol by looking at the message profiles for the different protocols, shown in Figure 11. 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

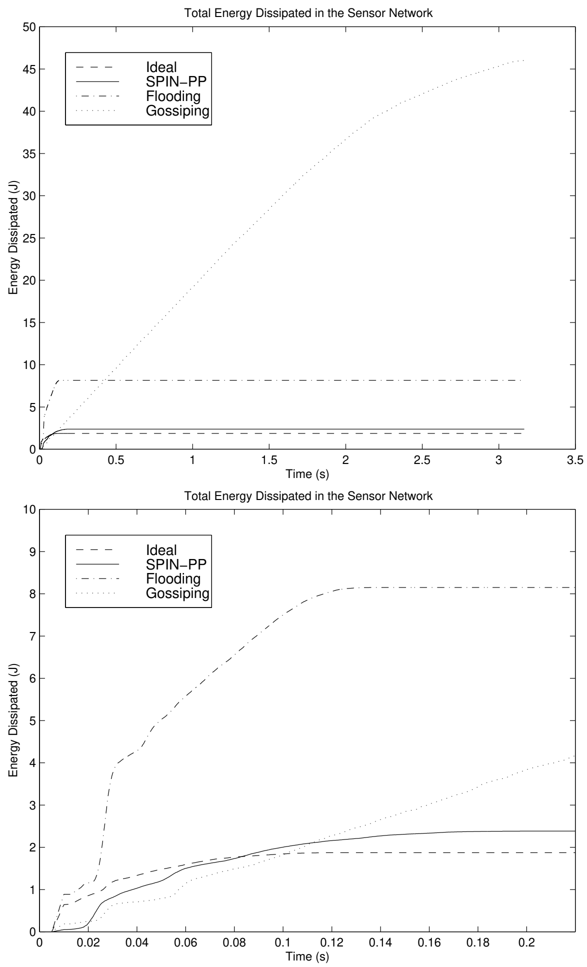

Figure 10: Total amount of energy dissipated in the system for each protocol. (a) shows the entire time scale until all the protocols converge. (b) shows a blow-up of the first 0.22 seconds.

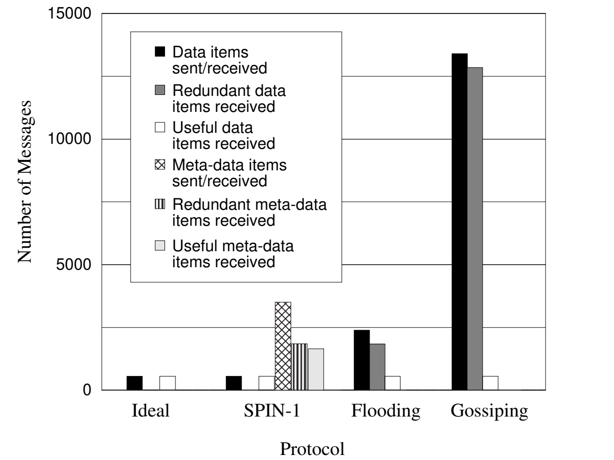

Figure 11: Message profiles for the unlimited energy simulations. Notice that SPIN-PP does not send any redundant data messages.

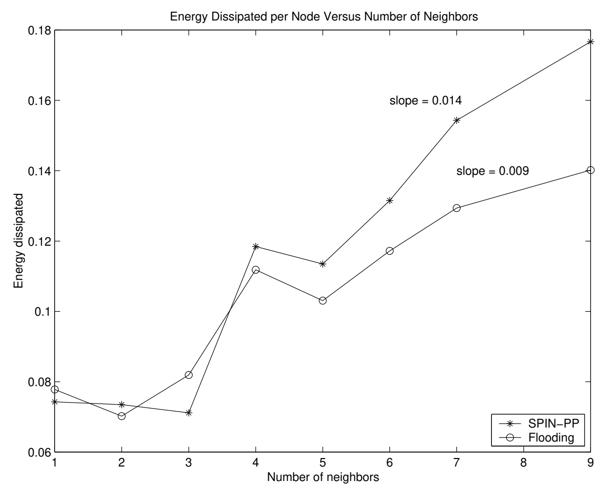

Figure 12: Energy dissipation versus node degree for unlimited energy simulations.

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-PP protocol is three times the number of useful data transmissions, since each data transmission in the SPIN-PP protocol requires three messages with meta-data.

Flooding and gossiping nodes send out many more data items than SPIN-PP 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 come at the high cost of 500 bytes each. SPIN-PP 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-PP 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 12. 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.

5.4 Limited Energy Simulations

For this set of simulations, we limited the total energy in the system to 1.6 Joules to determine how effectively each protocol uses its available energy. Figure 13 shows the data acquisition rate for the SPIN-PP, SPIN-EC, flooding, gossiping, and ideal protocols. This figure shows that SPIN-

Performance *Relative to Ideal	Protocol		
	SPIN-PP	Flooding	Gossiping
Increase in Energy Dissipation*	1.25x	4.5x	25.5x
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-PP, flooding, and gossiping protocols compared with the ideal data distribution protocol.

EC puts its available energy to best use and comes close to distributing the same amount of data as the ideal protocol. SPIN-EC is able to distribute 73% of the total data as compared with the ideal protocol which distributes 85%. We note that SPIN-PP distributes 68%, flooding distributes 53%, and gossiping distributes only 38%.

Figure 14 shows the rate of energy dissipation for this set of simulations. This plot shows that flooding uses all its energy very quickly, whereas gossiping, SPIN-PP, and SPIN-EC use the energy at a slower rate and thus are able to remain operational for a longer period of time.

Figure 15 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-EC over SPIN-PP, 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-EC is able to distribute 10% more data per unit energy than SPIN-PP and 60% more data per unit energy than flooding.

6 Broadcast Media Simulations

For our second study, we examined the use of SPIN protocols in a single, shared-media channel. The nodes in this model use the 802.11 MAC layer protocol to gain access to the channel. Packets may be queued at the nodes themselves or may be lost due to transmission errors or channel collisions. We used this framework to compare the performance of SPIN-BC, SPIN-RL, flooding, and an ideal data distribution protocol. We found that SPIN-RL is able to use meta-data to successfully recover from packet losses, while acquiring twice as much data per unit energy as flooding. Because flooding does not have any built-in mechanisms for providing reliability, it can not recover from packet losses and never converges.

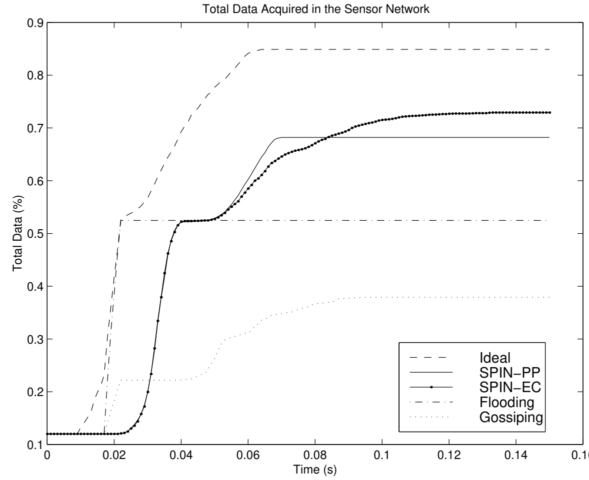

Figure 13: Percent of total data acquired in the system for each protocol when the total system energy is limited to 1.6 Joules.

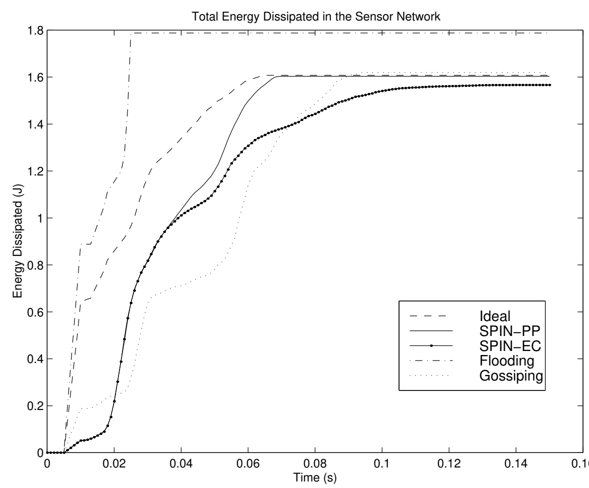

Figure 14: Energy dissipated in the system for each protocol when the total system energy is limited to 1.6 Joules.

6.1 Simulation Implementation and Setup

We used `monarch`, a variant of the `ns` simulator for all the simulations in this study. `monarch` [14] extends the functionality of `ns` to enable the simulation of realistic wireless communication. These extensions include a radio propagation model and a detailed simulation of the IEEE 802.11 DCF MAC protocol. We extended `monarch`'s MobileNode class to create wireless Resource-Adaptive Nodes. The only difference between these Resource-Adaptive Nodes and those described in Section 5 is that we replaced the wired Network Interface shown in Figure 7 with a wireless 802.11 MAC interface. We also made several modifications to `monarch`'s built-in 802.11 MAC implementation in order to perform our simulations. First, we modified the MAC implementation to appropriately subtract energy from a node's Energy Resource whenever it sends and receives a packet. Second, we added a switch to the MAC layer that turns off packet collisions and losses.

The simulation testbed that we used in our second study is the same as the testbed used in our first study. We used

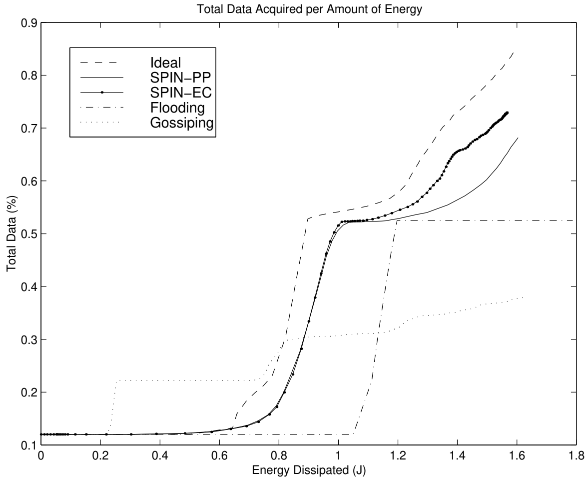

Figure 15: Data acquired for a given amount of energy. SPIN-EC distributes 10% more data per unit energy than SPIN-PP and 60% more data per unit energy than flooding.

the same topology and radio characteristics as those given in Figure 8 and in Table 1. The only differences between these two studies are that packets in this study may experience queueing delays and, depending upon the test configuration, may also be lost due to multi-path fading or packet collisions.

6.2 Simulations without Packet Losses

For the first set of simulations, we gave all the nodes a virtually infinite supply of energy, turned off MAC layer losses, and ran each data distribution protocol until it converged.

6.2.1 Data Acquired Over Time

Figure 16 shows the amount of data acquired by the network over time for each of the protocols. These graphs show that SPIN-BC converges faster than flooding, and almost as quickly as the ideal protocol. The difference in convergence times between SPIN-BC and flooding can be explained by queueing delays in the network. Recall that in a broadcast network, each node must wait for the channel to become free in order to send out a packet. When many nodes in a small area have packets to send, these nodes queue up their packets while waiting for access to the channel. If some of these packets are redundant, than they cause other, useful packets in the network to wait needlessly in queues. Flooding does not provide any mechanisms to circumvent implosion and overlap and therefore sends out many useless packets, as shown in Figure 17. These packets therefore cause unnecessary delays in the running time of the flooding algorithm.

6.2.2 Energy Dissipated Over Time

For the previous set of simulations, we also measured the energy dissipated by the network over time, as shown in Figure 18. These figures show that SPIN-BC reduces energy consumption by a factor of 1.6 over flooding. We can see the advantage of the SPIN-BC protocol by examining the message profiles for each protocol given in Figure 17. Because these protocols all use broadcast, some redundant data-transmissions are unavoidable, as illustrated by the ideal protocol's message profile. What this figure illustrates is that, by sacrificing small amounts of energy sending meta-data messages, SPIN-BC achieves a dramatic reduction in wasted data messages and a corresponding reduction in system energy and convergence time. Figure 20 further reinforces these results, showing that SPIN-BC nodes acquire 2 times more data per unit energy expended than flooding. The key results from these simulations are summarized in Table 3.

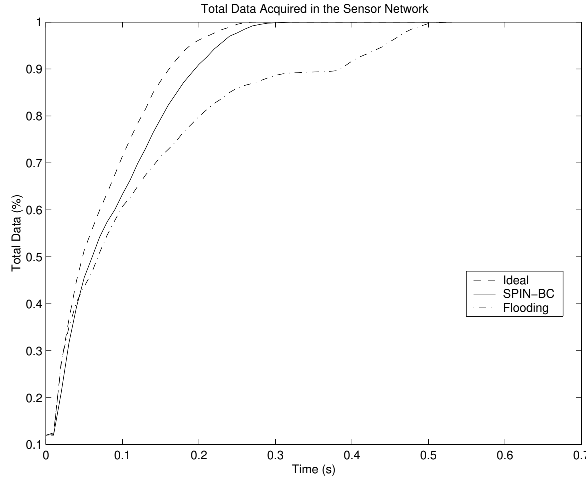

Figure 16: Percent of total data acquired in the system over time in a lossless broadcast-network.

Performance relative to Ideal	Protocol *no losses* SPIN-BC	Flooding	*losses* SPIN-RL
Increase in Energy Dissipation	1.6x	2.4x	1.6x
Increase in Convergence Time	1.1x	2x	5x
Slope of Energy Dissipation vs. Node Degree Correlation Line	.11x	1.67x	1.6x
Total Data Messages received	1x	2.2x	.89x
% of Total Data Messages that are Redundant	1.1x	1.8x	.96x

Table 3: Key results of the broadcast network simulations compared with the ideal data distribution protocol.

6.3 Simulations with Packet Losses

For the second set of simulations, we gave all the nodes a virtually infinite supply of energy and allowed the MAC layer to lose packets due to collisions and transmission errors. We compare SPIN-RL, our reliable protocol, to SPIN-BC and flooding. As a point of reference, we also compare SPIN-RL to the ideal protocol, run in a lossless network. We ran each protocol until it either converged or ceased to make any progress towards converging.

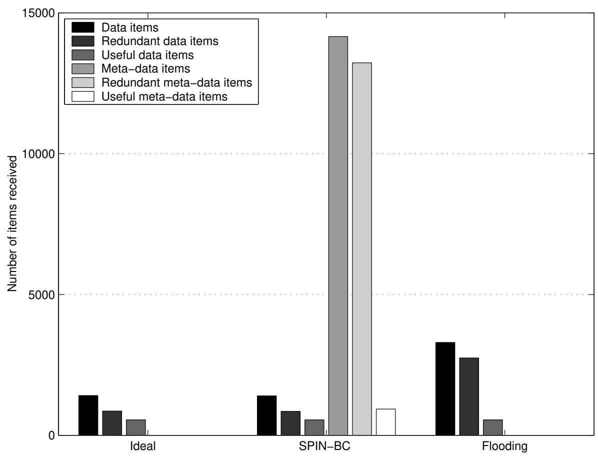

Figure 17: Message profiles for each protocol in a lossless broadcast-network.

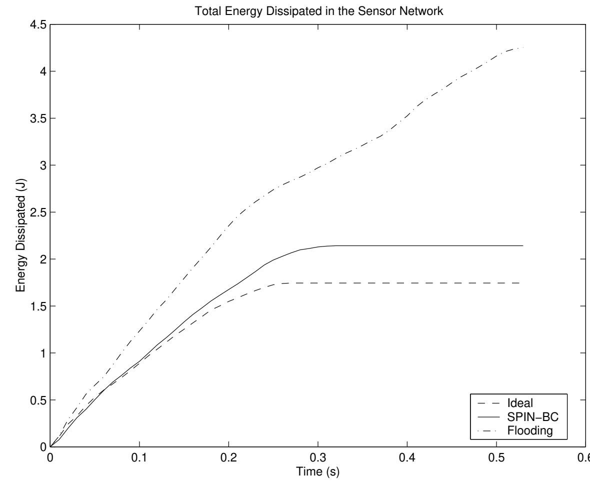

Figure 18: Total amount of energy dissipated in the system for each protocol in a lossless broadcast-network.

6.3.1 Data Acquired Over Time

Figure 21 shows the amount of data acquired by the network over time for each of the protocols. Only three of the protocols, namely SPIN-BC, SPIN-RL, and flooding were run on a lossy network. The ideal protocol was run on a lossless network, and is provided as a best-case reference point. Of all three of the protocols run on the lossy network, SPIN-RL is the only protocol that will retransmit lost packets, and therefore is the only protocol that converges. It is interesting to note that, although SPIN-BC outperformed flooding in the lossless network, it does not perform as well as flooding in a lossy network. We can account for SPIN-BC's poor performance by the fact that SPIN-BC nodes must successfully send and receive three messages in order to move a piece of data over a hop in the network, whereas flooding nodes only have to send one. SPIN-BC's protocol is therefore three times more vulnerable to network losses than flooding, which explains the difference in behavior we see between Figures 16 and 21.

6.3.2 Energy Dissipated Over Time

For the previous set of simulations, we also measured the energy dissipated by the network over time, as shown in Figure 22. These figures show that, of all the protocols, SPIN-RL expends the most energy, only slightly more than flooding. We can account for the relative energy expenditure of each protocol by examining the message profiles, given in Figure 25. Of all the protocols, SPIN-RL nodes receive the most data messages, as well as the most meta-data messages. This extra expenditure is well justified, however, if we look at how it is put to use. Figure 24 shows the amount of data acquired per unit energy for each protocol. Using almost the same amount of energy, SPIN-RL is able to acquire twice the amount of data as flooding. The key results from these simulations are summarized in Table 3.

7 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 [15]) periodically disseminate their view of the network topology to their neighbors, as discussed in [10, 25]. 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 [17]. 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, 18, 20, 24]. 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, 23] 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 SPIN 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 [19], 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].

Close 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.

There has been a lot of recent interest in using IP multicast [5] as the underlying infrastructure to efficiently and

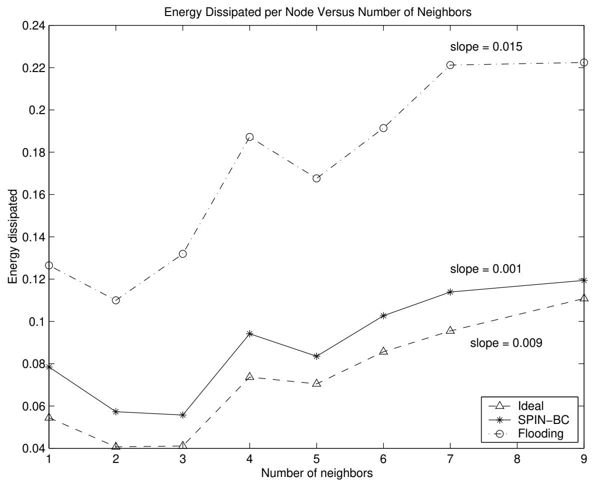

Figure 19: Energy dissipation versus node degree in a lossless broadcast-network.

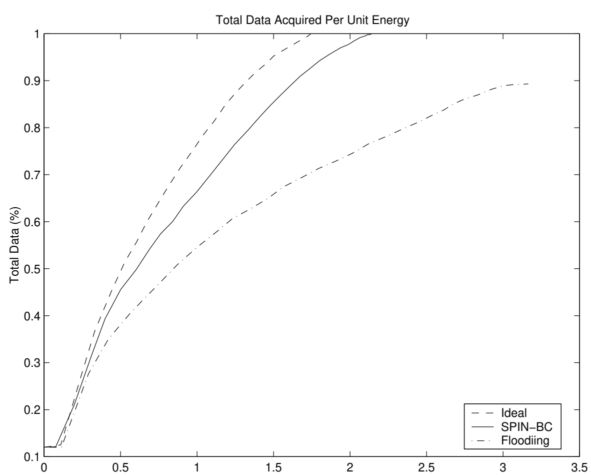

Figure 20: Energy dissipated versus data acquired in a lossless broadcast-network.

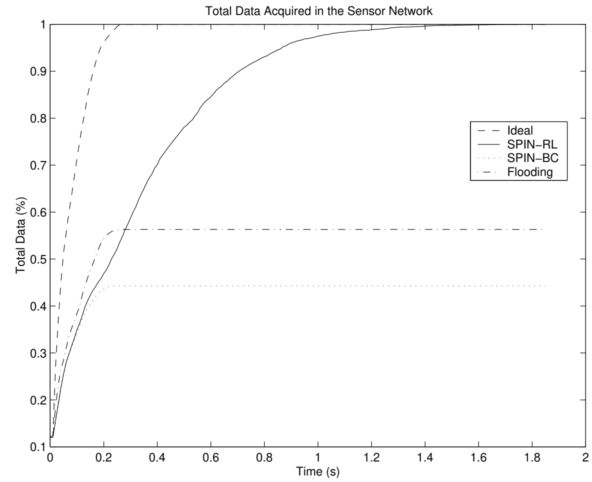

Figure 21: Percent of total data acquired in the system over time for each protocol in a lossy broadcast-network.

reliably disseminate data from a source to many receivers [22] on the Internet. However, for the reasons described in Section 4, we believe that enabling applications to control routing decisions is a less complex and better approach for wireless sensor networks.

8 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. Because they are 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 four specific SPIN protocols, SPIN-PP and SPIN-EC for point-to-point networks, and SPIN-BC and SPIN-RL for broadcast networks. SPIN-PP is a 3-stage handshake protocol for disseminating data, and SPIN-EC is a version of SPIN-PP that backs off from communication at a low-energy threshold. SPIN-BC is a variant of SPIN-PP that takes advantage of cheap, MAC-layer broadcast, and SPIN-RL is a reliable version of SPIN-BC. Finally, we compared the SPIN-PP, SPIN-EC, SPIN-BC, and SPIN-RL 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.

- The SPIN protocols are simple and efficiently disseminate data, while maintaining only local information about their nearest neighbors. These protocols are well-suited for an environment where the sensors are

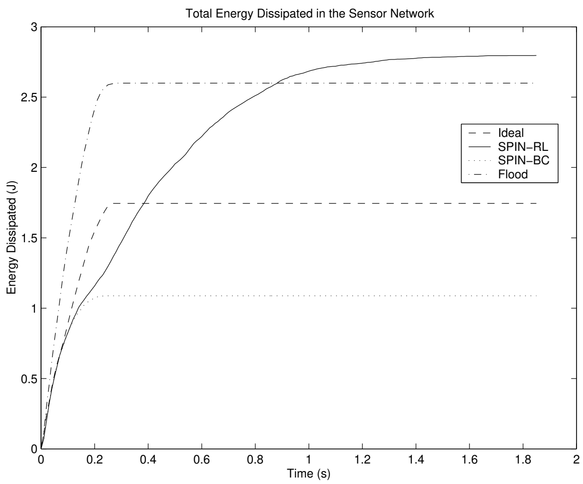

Figure 22: Total amount of energy dissipated in the system for each protocol in a lossy broadcast-network.

mobile because they base their forwarding decisions on local neighborhood information.

- In terms of time, SPIN-PP achieves comparable results to classic flooding protocols, and in some cases outperforms classic flooding. In terms of energy, SPIN-PP uses only about 25% as much energy as a classic flooding protocol. SPIN-EC is able to distribute 60% more data per unit energy than flooding. In all of our experiments, SPIN-PP and SPIN-EC outperformed gossiping. They also come close to an ideal dissemination protocol in terms of both time and energy under some conditions.

- Perhaps surprisingly, SPIN-BC and SPIN-RL are able to use one-to-many communications exclusively, while still acquiring data faster than flooding using less energy. Not only can SPIN-RL converge in the presence of network packet losses, it is able to dissipate twice the amount of data per unit energy as flooding.

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 work to be done in this area. Though we have discussed energy-conservation in terms of point-to-point media and reliability in terms of broadcast media, we would like to explore methods for combining these techniques for both kinds of networks, and we do not believe this would be difficult to accomplish. We would also like to study SPIN protocols in a mobile wireless network model. We expect that these networks would challenge the speed and adaptiveness of SPIN protocols in a way that stationary networks do not. 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.

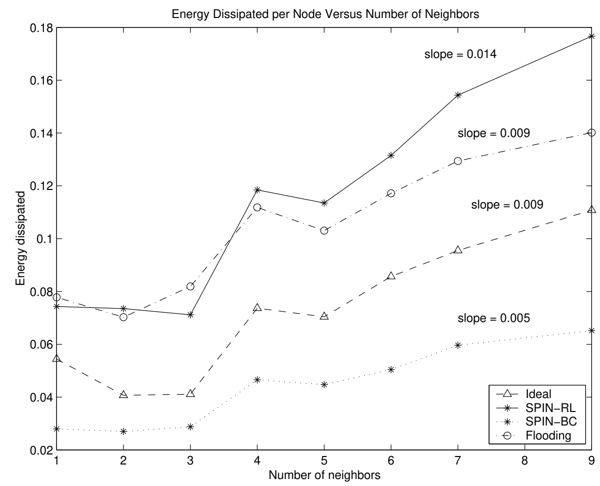

Figure 23: Energy dissipation versus node degree for each protocol in a lossy broadcast-network..

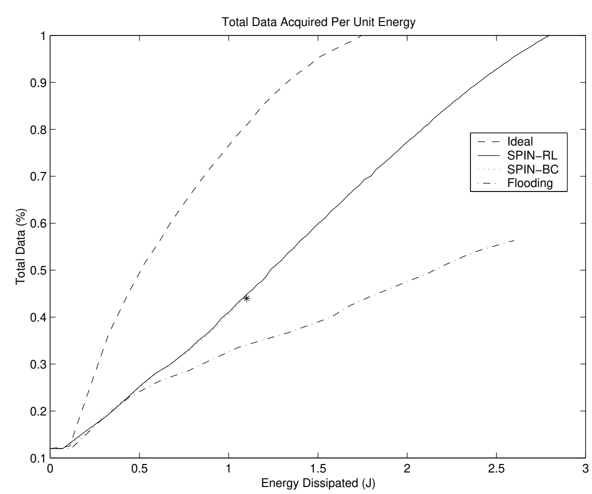

Figure 24: Energy dissipated versus data acquired for each protocol in a lossy broadcast-network. The $*$ symbol in the second graph highlights the last data-point of the SPIN-BC line.

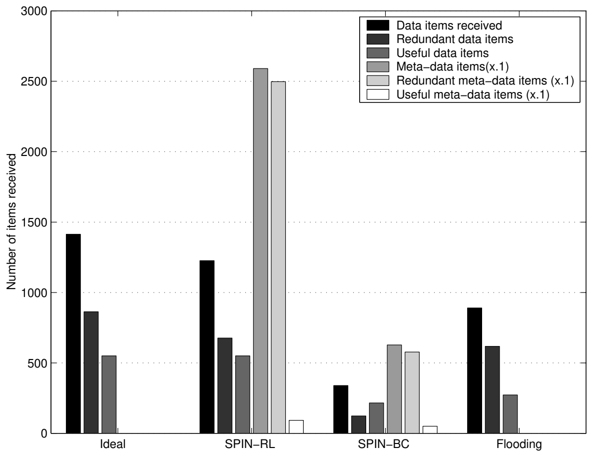

Figure 25: Message profiles for the each protocol in a lossy broadcast-network.

Acknowledgments

We thank Wei Shi, who participated in the initial design and evaluation of some of the work in this paper. We thank Anantha Chandrakasan for his helpful comments and suggestions throughout this work. We also thank Suchitra Raman and John Wroclawski for several useful comments and suggestions on earlier versions of this paper. This research was supported in part by a research grant from the NTT Corporation and in part by DARPA contract DAAN02-98-K-0003. Wendi 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] Monarch Extensions to the ns-2 Network Simulator. http://www.monarch.cs.cmu.edu/cmu-ns.html, 1999.

[15] Moy, J. OSPF Version 2, 1991. RFC 1583.

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

[17] Pahlavan, K., and Levesque, A. *Wireless Information Networks.* John Wiley & Sons, Inc., 1995.

[18] Park, V., and Corson, S. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Netowrks. In *Proc. INFOCOM'97* (Apr. 1997).

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

[20] 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).

[21] Raman, S., and McCanne, S. Scalable Data Naming for Application Level Framing in Reliable Multicast. In *Proc. ACM Multimedia* (Sept. 1998).

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

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

[24] Sinha, P., Sivakumar, R., and Bharghavan, V. CEDAR: A Core-Extraction Distributed ad hoc Routing Algorithm. In *Proc. IEEE INFOCOM* (Mar. 1999).

[25] Steenstrup, M. *Routing in Communication Networks.* Prentice Hall, 1995.