

Data-Centric Protocols for Wireless Sensor Networks

IVAN STOJMENVIĆ

University of Ottawa, Ontario, Canada

STEPHAN OLARIU

Old Dominion University, Norfolk, Virginia

This chapter reviews a number of emerging topics pertaining to a data-centric view of wireless sensor networks. These topics include data-driven routing, tracking mobile objects, constructing and maintaining reporting trees, dynamic evolution of a monitoring region for moving targets (mobicast), disseminating monitoring tasks, data gathering, receiving reports from a particular area of interest, and sending information and task assignment from a sink to all the sensors inside a geographic region (geocasting). The chapter also discusses various other issues, including sensor training options, data aggregation, data storage, as well as design guidelines for data aggregation and clustering, and rate-based data propagation in wireless sensor networks.

13.1 INTRODUCTION

13.1.1 Sensors and Sensor Networks

Recent technological advances have enabled the development of low-cost, low-power, and multifunctional sensor devices. These nodes are autonomous devices with integrated sensing, processing, and communication capabilities. A sensor is an electronic device that is capable of detecting environmental conditions such as temperature, sound, or the presence of certain objects. Sensors are generally equipped with data-processing and communication capabilities. The sensing circuitry measures parameters from the environment surrounding the sensor and

transforms them into electric signals. Processing such signals reveals some properties about objects located and/or events happening in the vicinity of the sensor. The sensor sends such sensed data, usually via radio transmitter, to a command center either directly or through a data collection station (a base station or sink). To conserve the power, reports to the sink are normally sent via other sensors in a multihop fashion. Retransmitting sensors and the base station can perform fusion of the sensed data in order to filter out erroneous data and anomalies and to draw conclusions from the reported data over a period of time. For example, in a reconnaissance-oriented network, sensor data indicates detection of a target, while fusion of multiple sensor reports can be used for tracking and identifying the detected target.

The block diagram of a typical sensor is depicted in Figure 13.1. The functionality of the sensing circuitry depends on the sensor capabilities. In general, the sensing circuitry generates analog signals whose properties reflect the surrounding environments. These signals are sampled using the analog/digital (A/D) converter and stored in the on-board memory as a sequence of digital values. The sensed data can be further processed using a data processor (microprocessor or digital signal processor (DSP)) prior to sending them over to the base station using the radio transceiver. The capabilities of the data processor are subject to a trade-off. A powerful DSP can be advantageous, since it will allow the sensor to transmit only important findings rather than excessive raw readings. Reducing the sensor's traffic generation rate can save the energy consumed by the radio transmitter and can decrease radio signal interference and collisions among the deployed sensors. On the other hand, sophisticated data processing can consume significant energy and can be a cost and a design burden by increasing the complexity of the sensor design. In all cases, the sensor has to include some control logic to coordinate the interactions among the different functional blocks. Such a control function also can be performed by the data processor, if included. Individual sensors have severely limited bandwidth and battery power. State-of-the-art sensors use one-to-all communication provided by omnidirectional antennas and communication on a single common

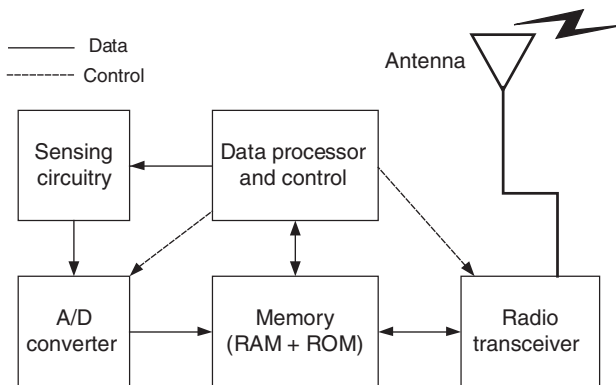


Figure 13.1 The block diagram design of a typical sensor.

channel (sensors using several frequencies, frequency hopping, or several transceivers and receivers are also being considered). Variants of IEEE 802.11 (designed to operate efficiently at low power consumption) are candidate medium-access control protocols for sensor networks, while Bluetooth appears to be an energy-expensive solution (Chapter 8 in this book is devoted to medium access). Sensor memory and processing capabilities are limited. Routing tables, if used at all, must be small. Data-compression and error-control schemes for sensor networks must be carefully selected. Secure operation is difficult to provide. There exists a great risk when using sensors. Sensor nodes can be defective, lost, damaged, compromised, or expired. Sensors in the active state spend considerably more energy than sensors in the sleep state, as discussed in several chapters in this book.

13.1.2 Applications and Physical Properties of Wireless Sensor Networks

Once deployed, the sensors are expected to self-configure into a *wireless network*. Sensor networks consist of a large number of sensor nodes that collaborate together using wireless communication and asymmetric many-to-one data. Indeed, sensor nodes usually send their data to a specific node called the *sink node* or *monitoring station*, which collects the requested information. The limited energy budget at the individual sensor level implies that in order to ensure longevity, the transmission range of individual sensors is restricted, perhaps of the order of a few meters. In turn, this implies that wireless sensor networks should be multihop. An important difference between wireless sensor networks and conventional networks is that sensor nodes do not need node addresses (e.g., medium-access control (MAC) address and Internet protocol (IP) address). In conventional networks (e.g., Internet), the node address is used to identify every single node in the network. Various communication protocols and algorithms are based on this low-level naming. However, wireless sensor networks are information-retrieval networks, not point-to-point communication networks. That is, wireless sensor network applications focus on collecting data, rather than on providing communication services between network nodes. Node address is not essential for sensor network applications.

Wireless sensor networks are a special case of ad hoc networks. However, there are several major differences between wireless sensor networks and ad hoc networks. To begin, the nodes of a wireless sensor network are generally densely deployed (e.g., hundreds or thousands of sensors may be placed, mostly at random, either very close or inside the phenomenon to be studied). Also, the number of nodes is typically not the same: while there are hundreds or thousands of sensors, the number of nodes (laptops, personal digital assistants (PDAs), palmtops, etc.) in an ad hoc network normally ranges from tens to hundreds. The sensors have a larger failure rate and feature lower data reliability, and are subject to stringent limitations in the energy budget, computing capacity, and memory. The nodes of an ad hoc network are normally distinguished by their IP addresses or other identifiers, while sensors are usually *anonymous*, lacking fabrication-time identifiers. Consequently, they are being addressed and named using various strategies that either

endow sensors with temporary IDs or else rely on data or position-driven naming. While ad hoc networks normally rely on topological information in their operation (e.g., knowledge of one-hop and often times 2-hop neighbors), such information may not be available in wireless sensor networks simply because of the lack of IDs at the individual sensor level. In some cases, however, the sensors benefit from a sense of relative geographic position with respect to the monitored environment and/or with respect to a sink. Thus, positional information (covered in Chapter 9 in this book) may be essential in some applications of sensor networks, although it may not be essential for ad hoc networks.

Depending on the application, different architectures and design goals/constraints have been considered for wireless sensor networks. We attempt to capture architectural design issues and highlight their implications on the network infrastructure and operation models proposed in the literature. We use the routing protocol as a vehicle for discussion in order to highlight how the infrastructure has been set to fit the network operational model and to deal with the specific architectural issue.

There are three main components in a sensor network. These are the sensor nodes, the sink, and the monitored events. Aside from the few architectures that utilize mobile sensors, most of the network architectures assume that sensor nodes are stationary. On the other hand, supporting the mobility of sinks, clusterheads (CHs), or gateways is sometimes deemed necessary. Routing messages from or to moving nodes is more challenging, since route stability becomes an important optimization factor, in addition to energy, bandwidth, and the like. The sensed event can be either dynamic or static depending on the application. For instance, in a target detection/tracking application, the event (phenomenon) is dynamic, whereas forest monitoring for early fire prevention is an example of static events. Monitoring static events allows the network to work in a reactive mode, simply generating traffic when reporting. Dynamic events in most applications require periodic reporting, and consequently generate significant traffic to be routed to the sink.

An important design consideration is the topological deployment of nodes. This is usually application-dependent and affects the performance of the communication protocol. The deployment is either deterministic or self-organizing. In deterministic situations, the sensors are manually placed and data are routed through predetermined paths. In addition, collision among the transmissions of the different nodes can be minimized through the prescheduling of medium access. However, in self-organizing systems, the sensor nodes are scattered randomly, creating an infrastructure in an ad hoc manner. In that infrastructure, the position of the sink or the CH is also crucial in terms of energy efficiency and performance. When the distribution of nodes is not uniform, optimal clustering becomes a pressing issue to enable energy-efficient network operation. During the creation of an infrastructure, the process of setting up the network topology is greatly influenced by energy considerations. Since the transmission power of a wireless radio is proportional to distance squared or even higher order in the presence of obstacles, multihop routing will consume less energy than direct communication. However, multihop routing introduces significant overhead for topology management and MAC. Direct routing would perform

well enough if all the nodes were very close to the sink. Most of the time sensors are scattered randomly over an area of interest, and multihop routing becomes unavoidable. Arbitrating medium access in this case becomes cumbersome.

Depending on the application of the wireless sensor network, the data-delivery model to the sink can be continuous, event-driven, query-driven, and hybrid. In the continuous-delivery model, each sensor sends data periodically. In event-driven and query-driven models, the transmission of data is triggered when an event occurs or when a query is generated by the sink. Some networks apply a hybrid model using a combination of continuous, event-driven, and query-driven data delivery. The routing and MAC protocols are highly influenced by the data-delivery model, especially with regard to the minimization of energy consumption and route stability. For instance, it has been concluded in that for a habitat monitoring application where data are continuously transmitted to the sink, a hierarchical routing protocol is the most efficient alternative. This is due to the fact that such an application generates significant redundant data that can be aggregated en route to the sink, thus reducing traffic and saving energy. In addition, in the continuous data-delivery model time-based medium access can achieve significant energy saving, since it will enable turning off sensors' radio receivers. Carrier sense multiple access (CSMA) medium-access arbitration is a good fit for event-based data-delivery models, since the data are generated sporadically.

In a wireless sensor network, different functionalities can be associated with the sensor nodes. In the early work on sensor networks, all sensor nodes are assumed to be homogenous, having equal capacity in terms of computation, communication, and power. However, depending on the application a node can be dedicated to a particular special function, such as relaying, sensing, and aggregation, since engaging the three functionalities at the same time on a node might quickly drain the energy of that node. Some of the hierarchical infrastructures proposed in the literature designate a CH different from the normal sensors. While some networks have selected CHs from the deployed sensors in other applications a CH is more powerful than the sensor nodes in terms of energy, bandwidth, and memory. In such cases, the burden of transmission to the sink and aggregation is handled by the CH.

13.1.3 Transport Layer Issues in Wireless Sensor Networks

The transport layer in wireless sensor networks is different from its counterpart in ad hoc and other types of wireless networks. There are several reasons for this. First, the reduced amount of traffic in sensor networks implies fewer congestion problems. Second, traditional end-to-end reliability does not usually apply in wireless sensor networks. Additionally, acknowledgments consume significant amounts of energy and are consequently avoided; similarly, the small on-board memory makes data-significant buffering at the individual sensor nodes level infeasible. The reliability of *individual* sensor measurement is low, and the goal is to provide good reliability of global sensor *network* measurement. Finally, quality of service (QoS) issues are of a different type in sensor networks: here it is more important to provide the reliability of a small amount of information rather than providing

bandwidth or delay guarantees. Therefore, the transport control protocols designed for wired networks or for other kinds of wireless networks cannot be used for wireless sensor networks.

When an event occurs, there is usually a multiple correlated data flow from the event to sink. A spatial correlation exists among the data reported. Several reports may arrive at the sink, or several reports can be combined at intermediate nodes to reduce communication (data fusion). The sink makes a decision on the event based on these reports, which has a certain degree of *collective reliability*. The transport-layer problem in wireless sensor networks can be defined concisely as follows: to configure the reporting rate to achieve the required event detection reliability at the sink with minimum resource utilization.

13.1.4 Query Processing

In other types of networks, queries are normally *address-centric* in the sense that they are sent to an individual node using, for example, IP-based routing. By contrast, the anonymity of sensors suggests that in wireless sensor networks queries be either *location-centric* or *data-centric*. Queries are addressed to a geographic region rather than to individual sensors. Since, as we discussed, the sensors do not have unique IDs, routes are created based on the nature and value of data collected by sensors. An example of data-driven routing is the response to a query that is asking to report all sensor readings with temperature over 40°C.

Queries can be distinguished along several orthogonal axes. Spatially, queries may be *global* and be sent to the *entire* deployment area, or *area-specific*, in which case they are addressed to a *geocasting* region (where only sensors inside a geographic region are asked to report), or to *multigeocasting* regions (where all sensors located inside several geographic regions are asked to report). In terms of the reporting mechanism there are several possible types of queries. We only mention the following three: event-driven, on-demand, and persistent. In an *event-driven query*, the sensor itself decides when it has something to report (for instance, when it measures high temperature, which may indicate incipient fire). In an *on-demand query*, the request comes from the end user via the sink. In a *persistent query*, the end user expresses a long-term interest in an event or a disjunction of events. The various sensors tasked with answering the persistent query report whenever a trigger event occurs during the lifetime of the interest.

13.1.5 Data Aggregation in Wireless Sensor Networks

When data are measured or arrive from a neighbor, the sensor needs to decide whether or not they are important enough to forward them. The coding techniques used need to minimize the number of forwarded bits. The new data may also be combined with other received data, in order to minimize the number of bits to forward. Such *data aggregation* (also referred to as *fusion*) from multiple sensors is important, because of severe energy and bandwidth limitations as well as for numerous other reasons, including reliability. The reliability of individual measurements

depends on the sensing distance and other factors. For instance, some sensors may be malfunctioning (there are also some security issues, see Chapter 7 in this book). The process of data aggregation from multiple sensors is also referred to as *collaborative signal processing*. Some sensors may aggregate data by doing some computation, such as computing the average of received values, computing the sum total of received values, and computing the largest/smallest of the received values. In order to maximize efficiency, wireless sensor networks may espouse division of work and functional specialization of sensors. For example, based on their relative position and remaining energy level, some sensors may forgo sensing, limiting their activities to data aggregation and data forwarding, while some other sensors may engage in a larger spectrum of activities or even in all the activities for which they qualify. An interesting aspect of the division of work is that it is done dynamically, balancing the load of the various sensors in order to extend as much as possible the useful life of the network.

13.1.6 Deployment Strategies, Time Synchronization and Position Awareness

There are several strategies for deploying wireless sensor networks. The sensors can be *embedded* in the ambient environment, be embedded in the asphalt covering streets and highways, in the walls of building, in trees, and so on. They can be placed *deterministically* by humans or robots, or incorporated in the paint coating walls, or deployed in a purely random fashion. Most research is devoted to *random placement*, where the sensors are dispersed randomly by plane, artillery, humans, or robots. Further, the initial deployment may be followed by later redeployment, as necessary.

Wireless sensor network self-organization includes a time component. One aspect of the problem is the time at which each sensor starts to operate. In many protocols, there exists an implicit assumption that all sensors start to operate at the same time, which could be preprogrammed, or may be externally decided and communicated. The later option is avoided because sensors need to be in the idle state to receive any instruction, which is much more energy-consuming compared to the sleep state (when receivers are turned off). Sensor network operation may require *time synchronization* (covered in Chapter 7 in this book), whether or not all sensors follow the same time or at least have synchronized time slots. Time synchronization can be provided by a global positioning system (GPS), by collaborative efforts, or can be achieved by some other means.

Some applications benefit or even require that the sensory data collected by sensors be supplemented with location information, which encourages the development of communication protocols that are location aware and perhaps location dependent. The practical deployment of many sensor networks will result in sensors initially being *unaware* of their location: they must acquire this information postdeployment. In fact, in most of the existing literature, the sensors are assumed to have learned their geographic position. The *location-awareness* problem is for individual sensors to acquire location information either in absolute form (e.g., geographic coordinates) or relative to a reference point. The *localization* problem is for individual sensors to

determine, as precisely as possible, their geographic coordinates in the area of deployment. One simple solution to the localization problem is to use a GPS, where sensors receive signals from several satellites and decide their position directly. However, for tiny sensors such direct position learning may not be possible or may not be sufficiently accurate enough (if a GPS signal is not provided with sufficient accuracy) for the assigned task. However, due to limitations in form factor, cost per unit, and energy budget, individual sensors are not expected to be GPS-enabled. Moreover, in many occluded environments, including those inside buildings, hangars, or warehouses, satellite access is drastically limited.

Since direct reliance of GPS is specifically proscribed, in order to obtain location awareness individual sensors exchange messages to *collaboratively* determine their own geographic position (absolute or relative) in the network. The vast majority of collaborative solutions to the localization problem are based on multilateration or multiangulation. These solutions assume the existence of several *anchors* that are aware of their geographic position (e.g., sinks or specialized sensors that can engage in satellite communication). By exchanging messages with their neighbors, individual sensors can conceivably measure signal strengths and/or time delays in communication. Some approaches are based on hop-count distances to reference points. Sensors receiving location messages from at least three sources can approximate their own locations. For a good survey on localization protocols for wireless sensor networks, we refer the interested reader to the relevant Chapter 9 in this book.

In some other applications, exact geographic location is not necessary: all that individual sensors need is *coarse-grain* location awareness. There is an obvious trade-off; coarse-grain location awareness is lightweight, but the resulting accuracy is only a rough approximation of the exact geographic coordinates. One can obtain this coarse-grain location awareness by a *training* protocol that imposes a coordinate system onto the sensor network. Olariu et al. [1] have shown that an interesting by-product of such a training protocol is that it provides partitioning into clusters and a structured topology with natural communication paths. The resulting topology will make it simple to avoid collisions between transmissions of nodes in different clusters, between different paths, and also between nodes on the same path. This is in contrast with the majority of papers that assume routing along spanning trees with frequent collisions. In the training protocol of Olariu et al. [1] the deployment area is endowed with a virtual infrastructure (for details see a dedicated Chapter 4 in this book). To make the presentation self-contained, however, we now outline the idea. Referring to Figure 13.2, the coordinate system divides the sensor network area into equiangular wedges. In turn, these wedges are divided into sectors by means of concentric circles or coronas centered at the sink. The task of training the wireless sensor network involves establishing:

- *Coronas*: The deployment area is covered by coronas determined by concentric circles centered at the sink
- *Wedges*: The deployment area is ruled into a number of angular wedges centered at the sink.

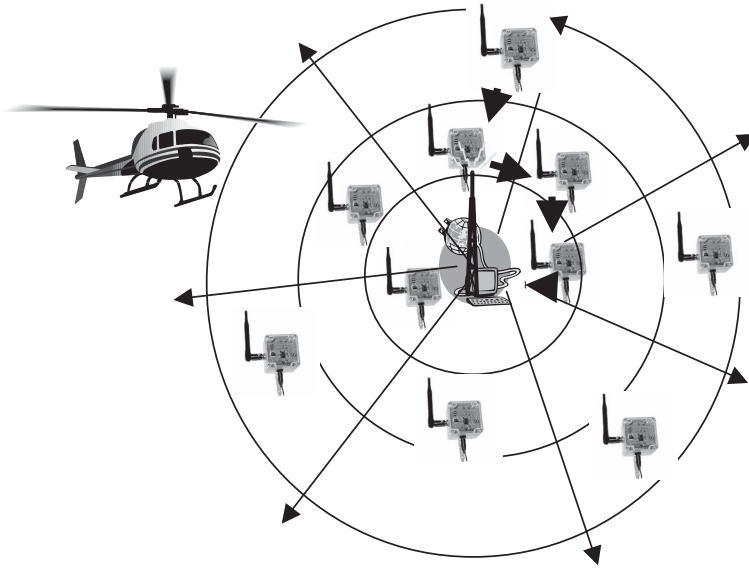


Figure 13.2 Training a wireless sensor network.

Individual sensors can acquire the desired coarse-grain location awareness by learning the identity of the corona and the wedge to which they belong. As it turns out, the training protocol is lightweight and does not require sensors to have IDs; moreover, sensors are not aware of their neighbors within the same sector. It is worth noting that location awareness is modulo the sector to which the sensor belongs. Since accurate position information is unreliable because of shadowing, scattering, multipaths, and time synchronization problems, training provides a viable alternative.

13.1.6 Topology Control and Area Coverage

In addition to gaining a sense of their location, sensors also need to gain some sense of their neighborhood. This can be achieved with various degrees of self-organization. For example, if sensors have IDs, they can discover neighbors by exchanging “hello” messages, and decide which neighbors and links are needed for their best operation, or what transmission range to select, to provide a certain density for reliable reporting and route construction. The communication may be critical for sparse networks, while for dense networks collisions, congestions and excessive energy expenditures may occur.

Since sensor batteries cannot be recharged under present-day technology, energy consumption is considered the most important parameter contributing to the longevity of the network. The best energy-conservation method is to have as many sensors as possible in *sleep* mode, where energy consumption is minimal. The network must be connected to remain functional, so that the monitoring station can receive the message sent by any of the active sensors. An intelligent strategy for selecting

and updating a set of active sensors that are connected is needed in order to extend the network lifetime. After learning about neighboring sensors, sensors decide whether to remain active or to go to sleep if their sensing areas are already covered. This problem is known as the connected *area coverage* problem, which aims to dynamically activate and deactivate sensors while maintaining the full coverage of the monitoring area. Efficient solutions to the connected area coverage problem are discussed in Chapter 11 in this book. When this coverage step is performed first, the large sensor network becomes reasonably sparse, but remains connected.

In the case of training [1], the optimal solution might be to keep a few active sensors in each sector, which can be decided by a simple leader election process. For example, each sensor may choose a time-out based on its remaining energy, and send a packet containing its sector information and remaining lifetime, so that other sensors in the same sector can hear that message, cancel their own transmission, and decide how long they could sleep.

Topology in sensor networks may change more frequently, because of failures, changes in sleep/active periods, and perhaps mobility. Designing efficient protocols for many operations requires a backbone, which is a subset of sensors, so that each sensor is either in a backbone or near it. Backbone examples include clustering and connected dominating sets. Active sensors can organize themselves into *clusters*. In a clustering process, some sensors may be selected as clusterheads (CHs), and every other sensor is assigned to one of the clusters. The alternative organization is to create backbones via *connected dominating sets* (each node is in such a set or is a neighbor of a node from the set). Backbone creation and sensor area coverage (which decides activity schedules) are covered in Chapter 11 in this book.

13.1.7 Localized versus Centralized Protocols

Estrin, Govindan, Heidemann, and Kumar [2] promoted the design of *localized* rather than *centralized* protocols in wireless sensor networks. Due to a number of factors, the topology of wireless sensor networks changes frequently and self-organization must be adaptive to local changes. *Centralized* protocols require global network information at each sensor (sink only, respectively, with sink making decisions) for making sensor decisions. This includes the use of topological structures, such as minimal spanning tree (MST), whose local links cannot be locally determined. There are a number of combinatorial optimization formulations of sensor network design problems with linear programming solutions. These protocols can perform well only when sensor networks are small. We do not discuss centralized approaches further, since we believe in and assume large-scale wireless sensor networks where centralized protocols do not work well.

Localized protocols only require local knowledge for making decisions, and a limited (usually constant) amount of additional information (e.g., the position of the sink). Some localized protocols may require preprocessing, such as constructing a suitable topology for further operation. One typical example is setting up a cluster structure. In addition to localized protocol operation, it is also important to consider

the maintenance cost of such topology. For instance, if the cluster structure is adopted, what happens when CHs move or fail? Does the update procedure remain local, and, if so, what is the quality of the maintained structure over time? Some maintenance procedures may not remain local. This happens when local change triggers message propagation throughout the network. Of course, *localized maintenance* is preferred, meaning that local topology changes should be performed by a procedure that always remains local, involving only the neighborhood of the affected sensors.

A number of protocols in the literature are localized, but use an excessive number of messages between neighboring sensors. For instance, some topology control and position determination protocols require over a dozen (sometimes even thousands of) messages to be exchanged between neighbors. Because of the severely limited bandwidth and energy budget and medium-access problems caused by excessive messaging, messages between neighbors to construct/maintain topology, determine position, or perform any other operation should be minimized, possibly avoided entirely (e.g., some backbone construction methods do not require any message after hello messages to learn that neighbors have been exchanged).

13.1.8 Roadmap of the Chapter

This chapter concentrates on localized protocols, featuring localized maintenance, and a limited number of messages between neighboring sensors. We begin by discussing *data gathering*—the most fundamental problems in wireless sensor networks. Data gathering has an implicit routing component, with or without involving data aggregation. Protocols for reporting an event (upon detecting it) by a single sensor are described in Sections 13.2 and 13.4. These protocols can be considered as responses to an event-driven query. The event may be detected by a group of sensors, but a single sensor reports it after data are aggregated first. Section 13.2 is devoted to protocols where a report is sent to the sink based on its position or merely distance to it (the later suffices in the case of direct transmission with omnidirectional antennas), without using any local information inferred by a dissemination originating from the sink. Section 13.3 discusses various ways for disseminating monitoring tasks from sink to sensor nodes. This is mainly done by applying broadcasting and geocasting protocols. Section 13.4 is devoted to data-gathering methods that are based on broadcast trees, which are constructed during the task-dissemination process from the sink, with sensors memorizing certain information that is later used for reporting. Section 13.5 and 13.6 discuss the case when all sensors in an area are requested to report as a reply to an on-demand query. Section 13.5 focuses on the case where data aggregation is not applied, whereas Section 13.6 looks at data aggregation as well. Data aggregation can be applied to all active sensors, or only to the active sensors within a region or a cluster. Section 13.7 discusses the case of mobile sinks or sensors. In Section 13.8, we discuss the problem of sending enough reports about an object to the sink so that the sink can accurately determine the position of the object. It also discusses tracking mobile objects using tree reconfiguration and mobicast protocols. Section 13.9 discusses

the problem of rate-based data propagation in sensor networks. Section 13.10 discusses an important corollary of the data-centric view of wireless sensor networks, namely, anonymity. Conclusions, exercises, and references complete this chapter.

13.2 DATA GATHERING WITHOUT MEMORIZING LINKS TOWARD THE SINK

13.2.1 Direct Reporting by Individual Sensors

The simplest way of reporting an event is to simply send a packet with sufficient power to reach the sink. If communication is omnidirectional, the exact position of the sink is not needed, since the approximate signal strength needed to reach the sink will suffice. Since sinks can always operate with more power than sensors, they can send a packet to all sensors announcing its presence, or assigning a task, or perhaps informing about its location. Sensors can also apply the power-increasing method to reach the sink, for example, to double the power applied for transmission until the sink acknowledges the receipt of the report. Since such direct communication may be over a long distance, it will drain the power quickly, and will drain it from all sensors. Therefore this method is presented here only for completeness, and has not been seriously considered as a viable option except for some small-size networks such as the home environment.

13.2.2 Direct Reporting by Cluster Heads

Heinzelman, Chandrakasan, and Balakrishnan [3] described the low-energy adaptive clustering hierarchy (LEACH) protocol for reporting data to the sink. Each node randomly decides whether or not to become a CH. The parameter used in decision making is the percentage of desired CHs. Sensors that decide to become CHs send a packet with their decision. Each node reports to the CH with the highest signal strength, and therefore clusters correspond to Voronoi diagrams of CHs. The CHs assign to each sensor from their cluster a time slot for reporting, aggregate data received from individual sensors, and send aggregated data directly to the sink. The selection of CHs is repeated periodically, to balance energy consumption. The optimal number of clusters is not investigated. LEACH is illustrated in Figure 13.3. The major problem with LEACH is that the sink may be very far from many CHs, therefore direct reporting may be extremely energy-consuming or even impossible. This basic method has variants that depend on how clusters are created. In some scenarios (e.g., military applications, with sensors attached to soldiers), there may exist natural cluster organization, especially if different types of sensors are being used. Different methods for forming reporting clusters are investigated in ref. [4]. Each sensor chooses a time-out interval. If no message is heard during that interval, the sensor decides to form a cluster and to report; otherwise, it becomes the follower of the sensor that sent the message.

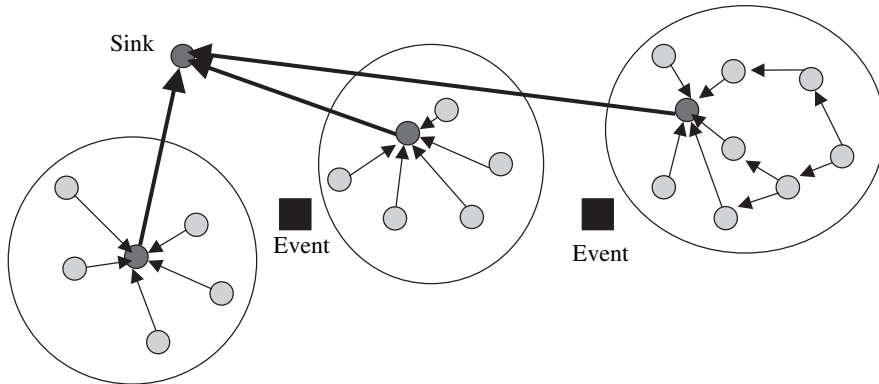


Figure 13.3 Data aggregation at CHs and direct reporting to sink.

13.2.3 Design Guidelines for Clustering and Aggregation in Sensor Networks

Mhatre and Rosenberg [5] considered the organization of sensors into clusters. Sensors could use either a single- or multihop mode of communication to send their data to their respective CHs. The CHs send their data directly to base stations. The energy needed for communication between two nodes at distance r is proportional to $r^\alpha + c$, where α is a power attenuation factor (between 2 and 6), and c is a constant that accounts for minimum reception energy and energy to run circuitry. The goal is to minimize and balance energy consumption. The authors analyze two modes of communicating between sensors and base stations, and derive conditions under which single-hop transmission by all nodes is best. One of the conclusions made is that, for $\alpha = 2$, there is no benefit from multihop communication. When multihop communication is better, each CH is assumed to be at the center of a circle divided into rings of equal width (equal to the used transmission radius). Therefore, they assume that each multiple hop is of approximately equal length and they find the optimal forwarding distance for each hop. The authors [5] do not prove that it is indeed optimal to use each hop of equal length (i.e., that the rings indeed all need to have equal width for optimality). Their result is based on minimizing the energy in a ring that is found to be critical. However, other rings may not be critical at that time. Our analysis [6], presented below, shows that, in fact, the rings are not of the same size for the optimal case (i.e., the sensor uses different transmission radii for maximizing network lifetime). We also note that the communication from CHs to base stations can also be multihop, via other CHs or even other sensors, instead of being single-hop. Finally, overall analysis is based on each sensor having an equal amount of data to report, which may not hold in a real application. Mhatre and Rosenberg [5] also studied the problem of determining and selecting the optimal number of CHs and required battery energy. Their derivations, however, are very complex.

Suppose that N sensors are randomly placed in a circle with fixed radius P . The task is to subdivide the circle into n rings, and determine their widths R_1, R_2, \dots, R_n , so that the network lifetime is optimized, where $R_1 + R_2 + \dots + R_n = P$ (that is, the sum is fixed). Optimization variables are therefore n, R_1, R_2, \dots, R_n . The sink is in the center of the first circle, of radius R_1 . It is assumed that the energy required for communication between two sensors (or sensor and sink) is proportional to $d^\alpha + c$, for some constants α ($2 \leq \alpha \leq 6$) and c . For instance, two particular models considered in ref. [7] are $\alpha = 2, c = 10^4$, and $\alpha = 4, c = 10^8$. For simplicity, the energy is charged fully to the transmitting node. Suppose that the sensor distribution is uniform, and therefore the number of sensors N_i in the i th ring is proportional to the areas of the rings. That is, $N_i/N_j = R_i^2/R_j^2$, and the sum $N_1 + N_2 + \dots + N_n = N$ is fixed, equal to the total number of sensors. It is assumed that each sensor helps a proportional number of sensors from the rings farther from the sink in retransmitting. To send a message from sensor in the i th ring to a sensor in the $(i - 1)$ th ring, we assume that the energy needed is proportional to $R_i^\alpha + c$, as an average amount with respect to ring size, or worst-case amount for the first ring (transmitting to sink, which is the zeroth ring). We also assume that the maximum transmission radius is limited, equal to T .

The sensors in the first ring spend (in the worst case) energy proportional to $R_1^\alpha + c$ to send their own message directly to the sink, and to retransmit each message. In addition, each of them retransmits a proportional number of messages from sensors in all other rings. The number of sensors it helps is therefore $(N_2 + \dots + N_n)/N_1 = (N - N_1)/N_1$, and the total number of messages it sends is $1 + (N - N_1)/N_1 = N/N_1 = P^2/R_1^2$. Thus the energy needed for these transmissions is $(R_1^\alpha + c)P^2/R_1^2$. This is a function of one variable, which has the minimum at $r_1 = (2c/(\alpha - 2))^{1/\alpha}$. For $\alpha = 2$, the energy is minimized for the maximal transmission range. The (unrealistic) case $c = 0$ is easy to discuss. We therefore continue the discussion only for the case $c > 0$ and $\alpha > 2$. Interestingly, the minimal energy for the first ring is obtained for the target radius that does not depend on P, N , and even n ! The target radius for the first ring is adopted if $r_1 < T$, otherwise, it must be changed to $r_1 = T$. If $P \leq r_1$, then the optimal number of rings is $n = 1$. Moreover, in this case $r_1 = P$. We assume that $P > r_1$ in the remaining analysis.

Now consider sensors in the last ring. They send only one message to sensors in the previous ring, which requires energy proportional to $R_n^\alpha + c$. The sensor network will maximize its lifetime when all sensors die at approximately the same time. Otherwise, the sensor network will not be able to either monitor or report the event. Therefore the optimal value of R_n is obtained when energies in the first and last rings are equal, that is, when $R_n^\alpha + c = (r_1^\alpha + c)P^2/r_1^2$ (here the optimal value for the first ring is already assumed). This equation has a straightforward solution for r_n (see ref. [6] for the formula) as the optimal ring size. Interestingly, the solution again does not depend on n and N , but it does depend on P , the overall circle size. If this optimal ring size is $>T$, it should be reduced to T . Note that $r_n \neq r_1$ (unless they are both “trimmed” to the same value T), which can be easily verified from the equation.

If $r_1 + r_n \geq P$, then the optimal value for n is $n = 2$. In the case of strict inequality, further energy savings cannot be achieved, because the limit on the first ring width does not depend on P . If $r_1 + r_n < P$, then more rings are needed, and the process can continue iteratively, from the last rings toward the first ring. Sensors in ring R_i will forward messages from a proportional number of sensors from rings R_j for $j > i$. The number of such sensors is $(N_n + \dots + N_{i+1})/N_i = (R_n^2 + \dots + R_{i+1}^2)/R_i^2$. Therefore, in the worst case, the sensors in the i th ring are expected to spend energy $(1 + (R_n^2 + \dots + R_{i+1}^2)/R_i^2)(R_i^k + c)$. Assume that the optimal values for rings $n, \dots, i + 1$ are already determined. The equation to be solved is then $(1 + (r_n^2 + \dots + r_{i+1}^2)/R_i^2)(R_i^k + c) = r_n^\alpha + c$. This is the equation of one variable, and the function has its minimum, obtained by standard calculus methods (finding a derivative) [6]. Let the optimal solution be $R_i = r_i$. If this solution is larger than T , then it should be changed to T . If $r_1 + r_n + r_{n-1} + \dots + r_i \leq P$, then $i = 2$, which determines the final value of n . Otherwise, it continues with the next value of i , effectively increasing n by 1. Note that, in the analysis presented, all sensors are assumed to be active.

Assume now that the transmission radii of all sensors are the same and fixed to T . Energy consumption can then be balanced by applying nonuniform sensor distribution. This problem was studied in ref. [8], with solution techniques involving sleep periods and energy consumption for routing tasks. We will extend the preceding solution to the case of nonuniform densities, following ref. [6], keeping all sensors active. Suppose that N sensors are randomly placed in a circle with fixed radius P . The task is to subdivide the circle into n rings of the same fixed widths $T = R_1 = R_2 = \dots = R_n$, and corresponding sensor densities $\rho_1, \rho_2, \dots, \rho_n$ in these rings so that the network lifetime is optimized. The number of rings n is therefore $n = P/T$, since $R_1 + R_2 + \dots + R_n = P$. Optimization variables are therefore $\rho_1, \rho_2, \dots, \rho_n$. The sink is in the center of the first circle, of radius T and density ρ_1 . Since ring areas are the same, the number of sensors N_i in the i th ring is proportional to their densities. That is, $N_i/N_j = \rho_i/\rho_j$, and the sum $N_1 + N_2 + \dots + N_n = N$ is fixed, equal to the total number of sensors. If all densities were the same, then balancing energy consumptions would not be possible, because sensors in rings closer to the sink are getting an increasing number of forwarding tasks, and the transmission energy per task is fixed. Therefore, for balanced energy consumption we have $\rho_1 \geq \rho_2 \geq \dots \geq \rho_n$. Suppose that $\rho_n = 1$, since other values would simply result in multiplying other densities by the same factor. The energy consumption is proportional to the numbers of messages sent. Sensors in the last ring send one message per considered time unit (which depends on reporting the rate), as a result of their monitoring. Sensors in the first ring, being more densely spread, require a lower reporting rate for their own monitoring. They therefore send $1/\rho_1$ reports in the same time frame. Similarly, sensors in the i th ring generate $1/\rho_i$ reports. Sensors in the first ring retransmit a proportional number of messages from sensors in all other rings. The number of sensors each of them helps is therefore $(N_2/\rho_2 + \dots + N_n/\rho_n)/N_1 = (n - 1)/\rho_1$, and the total number of messages it sends is $1/\rho_1 + (n - 1)/\rho_1 = n/\rho_1$. Thus, $n/\rho_1 = 1$ or $\rho_1 = n$. Continuing this discussion, we conclude that $\rho_i = n + 1 - i$. With this solution, each sensor sends on average one message per unit time, independently on the ring it is contained in.

13.2.4 Data Aggregation with Consensus

A data-aggregation and -consensus algorithm for object location and tracking by a sensor network is described by Kumar, Schwiebert, and Brockmeyer [9]. The first node that detects an event will first generate consensus by obtaining a quorum from nodes having similar interests and area of coverage. If more than half of the sensors close to the event confirm the same observation by acknowledging to the initiating node, the node will report the event.

13.2.5 Multihop Reporting among Nodes or Clusters

The direct communication from CHs to sinks may be impossible because of distance or can be extremely energy-consuming. Further, even communication from any sensor to its own CH can have such problems. If multihop reporting is applied, packets can be forwarded among CHs only until they reach the sink (if transmission power is adjusted to reach a neighboring CH), or the route could include bridge (or gateway) sensors between adjacent clusters. Note that this multihop forwarding using other CHs can be applied with or without further data aggregation, beyond the initial one within each cluster.

Alternatively, cluster organization may not be necessary. Sensors may react to an event by first finding consensus among other nearby sensors that detect the same event (e.g., the consensus method [9]), and then the lead sensor applies multihop reporting, that is, routing via other active sensors in the network toward the sink.

13.2.6 Reporting with Energy-Efficient Routing

Multihop reporting can be performed with a routing algorithm that aims at minimizing hop count. Alternatively, the algorithm may attempt to minimize the energy expenditure needed for a given routing task, or to maximize the network lifetime by considering the remaining energy when selecting forwarding neighbors. In Chapter 12 in this book, routing protocols with mentioned optimization criteria are surveyed. For the sake of completeness, we summarize here a few relevant protocols.

Schurgers and Srivastava [10] propose that nodes collect several packets intended for the same neighbor into a single packet. They claim that compression can be achieved in this way, leading to more energy efficiency. They also propose stochastic schemes where the best neighbor is chosen at random, an energy-based scheme where the best neighbor is selected based on its energy, and a stream-based scheme where busy nodes inform their neighbor by asking them to select other forwarding nodes instead.

Chatzigiannakis and Nikolettseas [11] describe a *routing protocol for sensors* that have the sense of direction, but do not know their coordinates. The monitoring center is a wall known to sensors, and wider than the width of the sensor network. The task of reporting from a sensor to the wall proceeds by a greedy algorithm, which follows the direction orthogonal to the wall. At each step, the node currently holding the message broadcasts a search message looking for another sensor within an angular

range with respect to the wall direction and at a certain minimum distance (and maximum distance, which is the transmission radius). Thus, each sensor has the ability to estimate the distance to neighboring sensors. Each awake sensor located in the desired cyclic sector will report back to current node *A*, but only the first such node *B* will receive the full message from the current node. Current node *A* will wait to hear the forwarded message from *B* to one of its neighbors. If successful, *A* will go to sleep. If there is no closer node to the wall from *B*, a failure message is generated and the message is backtracked to *A*. Note that no two consecutive backtrack steps are possible, so this simple greedy routing may fail. Note also that the *greedy-face-greedy* (*GFG*) algorithm [12] can be used to guarantee delivery.

13.2.7 Sector Routing

In the case of *sector training* [1], messages are not directed toward any particular sensor. Instead, they are directed toward a sector. All active sensors inside the sector receive the message. One of them decides to retransmit, and others in the same sector (if more than one in a given sector is active) can overhear this transmission, which prevents them from their own retransmissions. This assumes that inside a sector the sensors are within the communication range of each other. This may or may not be true in general. In any case, forwarding toward the sink then follows a sector-routing principle: a route is created from sector to sector, with an arbitrary sensor from each sector participating (see Fig. 13.2, showing a route from the top sector toward the sink). In case of empty sectors, a variant of routing with guaranteed delivery [12] can be applied, since sectors are creating a planar graph. In recovery mode, face routing can be employed using direction orthogonal to the wall (that is, with the destination being imagined at infinity).

13.2.8 Data-Centric Storage

In some particular scenarios, wireless sensor networks can operate, at least temporarily, without a sink. In this case, reports by sensors need to be stored in the sensor network itself. Ratnasamy, Estrin, Govindan, Karp, Shenker, Yin, and Yu [13] described a data-centric storage system for application in wireless sensor networks. Sensors have a tiny memory, and therefore limited storage capacity. Therefore, they need to distribute storage among themselves. The algorithm in ref. [13] is to apply a hash function to a keyword assigned to a file, datum, information, or an object, which will map it to a point with geographic coordinates. The hash function needs to be carefully selected so that the obtained point is inside the geographic region containing the sensors. A planar graph over sensor network can be obtained by applying a Gabriel graph (*GG*) structure (described in Chapter 10). The information is stored in all sensors on the face containing the mapped point. In order to retrieve the information, *GFG* routing that guarantees delivery [12] can be applied. Since the mapped node is generally not in the sensor network, routing will create a loop along the face containing it. Sensors on that face have already stored the information and can provide it to the requester.

13.3 DATA DISSEMINATION FROM THE SINK

13.3.1 Broadcasting Short Packets from the Sink

This subsection discusses various ways the sink assigns monitoring tasks. The tasks assigned to sensors need to be propagated to all active sensors in the network (broadcasting), or to all the active sensors located inside a region of interest (geocasting), consisting of currently active sensors (e.g., sensors selected for area coverage). One simple solution, if it is assumed to have sufficient transmission power to cover the entire deployment area, is that the sink sends one message that reaches all the sensor nodes in the network. If the sink does not have sufficient power, the sensors themselves need to retransmit such messages. If the packet containing the task and the location of the sink is relatively short then the data dissemination can be fulfilled by any of the broadcasting protocols, covered in Chapter 11. Most of these protocols assume that the sensors know the position of their local neighbors. Otherwise, *blind flooding* can be applied, meaning that each sensor receiving the packet for the first time will rebroadcast it. This method is the most popular in the existing literature (for instance, it was applied in protocols given in refs. [14] and [15]). Intelligent flooding (broadcasting) schemes are surveyed in Chapter 11. Some sensors do not need to retransmit the task packet, and the task can still be distributed to all the sensors (assuming an ideal medium-access protocol). One such method is beaconless area-based broadcasting [16], where a sensor whose communication area is completely covered by transmissions from other sensors does not need to retransmit. Note that this method does not require prior knowledge of neighbors. When sensors need to report using a broadcast tree, they can link themselves to one of the nodes from which the packet was received.

Acknowledgments for the receipt of monitoring tasks may or may not be sent. If requested, it can be provided, for instance, as follows. Lipman, Boustead, and Chicharo [17] proposed to send acknowledgments only to neighbors along *local minimum spanning tree (LMST)* edges. Each sensor then on average sends only at most two acknowledgments, because of the sparse LMST structure. To construct a LMST [18], each node first constructs a MST of its local neighbors (knowing their geographic positions), and keeps edges that are included in such local MSTs by both end points (see more details in Chapter 10). On average, each node will send one acknowledgment only, since the average degree (average number of neighbors) of a LMST is only slightly larger than two.

13.3.2 Broadcasting Long Packets from Sink

If the message containing a detailed assignment, or other type of message that needs to be disseminated, is relatively long, then an alternative is to send two types of messages instead: a short message is sent first that offers a long message to sensors, followed by a long message sent only to those sensors that require it. In the *sensor protocol for information via negotiation (SPIN)* [19], each node that receives the datum (full message) that is being broadcast will forward the corresponding

metadatum (short message) that has a considerably shorter bit length (e.g., 16 bytes instead of 500) to all its neighbors. The metadatum is thus flooded. Neighboring nodes that did not yet receive the full message will reply to the short message with a request to receive it. The sensor will then respond by sending the full message to all nodes that requested it. If an omnidirectional antenna is used, sensors may retransmit the full message upon receipt of the first request for it. Note, however, that, short request messages may be sent back to the transmitting node only if it is a neighbor in a selected sparse, connected structure, as observed in ref. [20], greatly reducing the amount of short messages needed. For example, if a LMST [18] is used as the sparse structure, the reduction is about $2/d$ times, where d is the average number of neighbors in the network. This reduction is possible if nodes have 2-hop topological or one-hop positional information about their neighbors.

13.3.3 Geocasting in Wireless Sensor Networks

Data dissemination, or task allocation, from the sink does not need to be propagated to all active sensors. If only sensors that are close to a monitoring event (e.g., a factory that pollutes the environment) need to be alerted, then only sensors located inside a geographic region need to receive the task. This problem is known as *geocasting*. A survey of existing geocasting schemes is given in ref. [21]. It was shown that most existing geocasting schemes do not guarantee delivery to all nodes inside a region, the main reasons being either the partitioning of the network inside the region, or applying greedy routing instead of one that guarantees delivery.

Yu, Govindan, and Estrin [22] considered a geocasting variant of the data-gathering problem. They describe the *geographic and energy-aware routing (GEAR)* algorithm, which uses energy-aware neighbor selection to route the packet toward the target region, and recursive geographic forwarding, or restricted blind flooding algorithm, to disseminate the packet inside the destination region. Recursive forwarding applies GEAR to send messages to four subregions in the geocast region, which repeats until the region has a single node inside it. Blind flooding does not guarantee delivery to all sensors inside the region, because of possible partition inside the region (but connectivity outside it), and can be replaced by a more intelligent scheme (see Chapter 11 in this book). The GEAR algorithm selects a forwarding neighbor (among those that are closer to the destination), which minimizes a linear combination of their distance to the destination and the energy they already spent. This is almost equivalent to the cost-aware localized scheme by Stojmenović and Lin [7], originally proposed in 1998 (described in detail in Chapter 12). Yu, Govindan, and Estrin [22] also claim that GEAR can avoid holes by applying a learning A^* algorithm-based approach, without presenting details. To avoid holes, one can use, for example, the depth-first search (DFS) approach [23]. This approach requires memorizing past traffic at nodes. Unfortunately, it does not guarantee delivery to all sensors in the geocasting region, because of possible partitioning inside the geocasting region.

We observe that, to guarantee delivery to all sensors in a geocasting region, and also to avoid memorization, GFG [12] can be applied first, while some optimizations

(described in ref. [22]) can follow later on recursively. Note also that ref. [24] further elaborated on the use of GEAR for various forms of data dissemination, without giving its description.

Three existing geocasting algorithms that guarantee delivery to all nodes inside the geocasting region (subject to the ideal medium-access layer and connectivity of these nodes to the source) are described in refs. [21] and [25]. One algorithm [21,25] is based on multicasting to entrance zones, and flooding from entrance zones to nodes inside the geocasting region. Bose, Morin, Stojmenović, and Urrutia [12] observed that a geocasting algorithm will guarantee delivery if all faces of a planar graph that are inside or intersect the geocasting region are traversed. The geocasting algorithm [12] is based on a DFS of the face tree, constructed from a node inside the geocasting region.

Seada and Helmy [26] observed that it is sufficient to traverse only faces that intersect the boundary of a given geocasting region, and proposed the following algorithm. Source node first uses *GFG* algorithm [12] to forward the packet toward the region. Each node that is inside region will retransmit the packet when receiving it for the first time (“regional flooding”). If the node also has neighbors outside geocast region, it will instruct them to perform face traversals using “right-hand” rule (see chapter on routing in this book for details). The first node inside the region to receive the face traversal packet floods it inside the region or ignores it if that packet was already received and flooded before [26]. Figure 8 in ref. [26] shows that face traversal was not assumed in cases when an outer node brings the packet inside the region (the receiving node then only floods the region, but does not instruct the sender node outside the region to then also perform face traversal). Therefore, as elaborated in ref. [21], the algorithm [26] does not guarantee delivery, despite the claim. A protocol that does guarantee delivery (with proof of it) was described in refs. [21] and [26].

Algorithm Geocast_traversal_intersecting_faces

- The source node S sends the message toward the geocasting region, using the *GFG* algorithm [12];
- Each node inside the region retransmits the message when receiving it for the first time, and ignores it when receiving it again;
- Each internal border node (node inside a region having neighbor(s) on planar graph outside the region) will instruct (together with retransmission) all its perimeter neighbors outside the region to perform right-hand-based face traversals;
- Each external border node (node outside the region having neighbor(s) on the planar graph inside the region) will initiate right-hand-based face traversal(s) with respect to all edges leading to internal-perimeter neighbors, after receiving the first copy of the message, and will ignore further received copies unless a packet is received from an external neighbor following a different “external” face (in which case it forwards it along that face, as requested). Each traversal is performed until another node that is inside the region is found.

13.3.4 Multicasting in Wireless Sensor Networks

A monitoring task can also be disseminated to all the sensors located in several geocasting regions. Assuming that these regions are relatively small, a position-based multicasting protocol [27] can be applied. Mauve, Fusler, Widmer, and Lang [27] proposed two multicasting schemes, with some optimizations. In the *optimal-paths* method, each node receiving a multicasting message for a group of nodes will forward it to each neighbor that is closest to one of the group members. More precisely, each group member is assigned to the neighbor that is closest to it (provided that neighbor is closer to it than the current node). In the *aggregate-paths* method, for each neighbor A , the number of destinations for which A is the closest node is determined. Then a covering algorithm is applied. Basically, a neighbor is chosen that covers the maximum number of destinations, these destinations (and other nodes for which a selected node makes some progress) are eliminated from the list, then another neighbor is chosen that covers the maximal number of remaining destinations, and so on. The forwarding list of multicast group is similarly changed as in the previous algorithm [27]. In both schemes, if no neighbor is closer to one or more destinations, then the recovery mode in the GFG algorithm [12] is applied. The virtual destination used for the recovery mode is calculated as the position representing the average of the positions of the affected destination nodes. When a node receives a multicast packet in recovery mode, it checks for each destination, if it is closer to that destination than the node where the packet entered recovery mode. For all destinations where this is the case, greedy multicast forwarding can be resumed, as described in the corresponding scheme. For all other destinations, recovery mode is continued, with an updated average of positions of affected nodes (those not recovered yet). Both optimal- and aggregate-path methods can be modified by considering metrics different from hop count, such as power, cost, or delay. Greedy routing can be replaced by power and/or cost-aware routing (see Chapter 12), and forwarding neighbors will be judged based on the metric in question, combined with their coverage ability, for their selection.

13.4 DATA GATHERING BASED ON MEMORIZED BROADCASTING TREES

Sensors (or their CHs, when they are clustered), may report back to the sink using a tree structure that is constructed together with the task allocation from the sink. This tree is referred to here as the *broadcasting tree*, since it is usually set during the *broadcasting* operation. It sends task allocation and other (e.g., sink position) information from the sink to all the sensors in the network. The use of broadcasting tree also implies the need to memorize some information made available during broadcasting process, which is then used in the reporting phase. The broadcasting tree consists of links along which the sensors learned about the position of the sink(s). Therefore, the sink monitoring implicitly informs the sensors that link to be used for replying. Sensors then create links for reporting along the *reverse broadcast tree*. This term is used, since reporting is normally applied in the direction that is

opposite to the direction of request propagation from the sink. Most of the literature considers this type of sensor training for reporting. Note that the broadcasting operation is applied when all active sensors are alerted to report possible events. The broadcasting tree can also be set during a *geocasting* operation, where the monitoring station requests reports only from sensors located inside a geographic region. This can be further generalized to *multigeocasting* operations, which disseminates a request to all sensors located inside several geocasting regions. Because of the volatility of individual sensors (failures or changes between sleep and active periods), the use of broadcasting trees for reporting has certain risks, since a particular node or link may not be available although demanded by the memorized response path.

Carle and Simplot-Ryl [28] proposed the following framework for wireless sensor network operation. Sensor-area monitoring consists of three phases or subproblems. The first one is to select the sensors that are needed for *connected-area coverage*, placing other sensors in sleep mode. The second phase is to construct a *broadcasting tree* from the sink to all *active* sensors. They consider two types of trees, minimum energy broadcasting or dominating set based. The last phase is to *report* events using the reverse broadcast tree.

13.4.1 Directed Diffusion

Directed diffusion [14] is a often cited scheme for data gathering by using a data-centric routing scheme. The data sink identifies a set of attributes and propagates an *interest* message throughout the network. The interest is flooded throughout the network (apparently blind flooding was used). Each receiving node records the interests and establishes the so called *gradient*, the state indicating the next hop direction for other nodes to report data of interest. When an interest arrives at a data producer, data are being forwarded to the sink along established gradients. Note that the algorithm is similar to the well-known ad hoc on-demand distance vector (AODV) routing scheme [29], considered as a possible routing standard. Flooding the interest with attribute-based addressing corresponds to the route discovery with IP or ID addressing. Instead of comparing their *address* with the destination address as in AODV, sensors in directed diffusion compare the *interest* from the packet with the data they measure and their location if the interest is location specific. Therefore various AODV optimizations that exist in the literature are applicable in the context of directed diffusion. Although it is an on-demand localized scheme that does not require prior “hello” messages, the scalability is questionable. If the interest is location specific, then obviously it is much more efficient to route the request (using, e.g., a protocol that guarantees delivery [12]) toward the location of interest instead of flooding it to the whole network. The protocol described in ref. [14] uses path memorization for reporting the sensor measurements back toward sink.

13.4.2 Reporting via Neighbor with Smaller Hop Count

Ding, Sivalingam, Kashyapa, and Chuan [15] considered the problem of finding a route from a sensor to the single sink in a wireless sensor network. Following a

reactive route discovery strategy, the sink floods the network and sets the routes. The difference is that each sensor does not memorize the whole route, or a single pointer to the previous sensor on the route, but instead memorizes its hop-count distance to sink. When a packet is sent toward the sink, any neighbor at one less hop distance can forward it, instead of reporting back to the first node that sent the task assignment packet to it. For instance, a report can be sent to the neighbor with the highest energy and smaller hop count, or any neighbor that sent the packet with a smaller hop count from the sink [15]. The node can memorize few such alternatives during the setup phase and try them one by one. Alternatively, a neighbor at one less hop distance can simply retransmit, and the node can block further retransmissions by a separate blocking packet.

Fujiwara, Iida, and Watanabe [30] proposed a mechanism that allows nodes to maintain their routes to the base station via multihopping, if needed. If a direct link between any node and its base station is broken, the node starts monitoring communications in its neighborhood to find a node that is still connected to the base station, either directly or by multihopping. When the node finds a connected neighbor, which should be one hop nearer, it marks it as its router and sends to it the packets that must be sent to the access point. This allows nodes to always be able to connect to their base station. The authors consider only the case of a single access point.

Zou, Nikolaidis, and Harms [31] described several localized schemes for constructing reporting trees for sensors to the sink. The tree construction starts at a sink node, which floods a message in the network. Upon receipt of several copies of the message, a given node may decide which of the nodes that sent the message is best to use for reporting data back to the sink. Authors described several possible localized criteria for selecting the best neighbor: minimum distance to the next hop, maximum distance to the next hop, random next hop, maximum degree of the next hop neighbor, and the maximum size of the 2-hop neighbor set of the next hop neighbors.

Wireless sensor networks with multiple sinks are special cases of hybrid wireless networks considered in ref. [32]. The hop distances to the closest sink, and therefore the routes, can be similarly determined as in the case of the single sink [15,30], as described in ref. [32]. Each access point sends messages toward sensors to establish reporting links. Each sensor may receive such messages from multiple sinks, but will forward them only from the closest sink. This will reduce the amount of traffic, without affecting the choice of the closest sink. If all sinks start the process synchronously, at the same time, then only one message is forwarded by each sensor. Otherwise, sensors will forward a new message only if it comes from a sink that is closer than the previously closest sink from which such a message was already received. This algorithm constructs the reporting trees from each sink to all sensors for which that sink is the closest one.

13.4.3 Reporting via Alternate Paths in a Broadcast Tree

Most of the current methods for reporting sensor data first construct a broadcast tree from the source, then use this tree for reporting in reverse order. Nodes in the tree

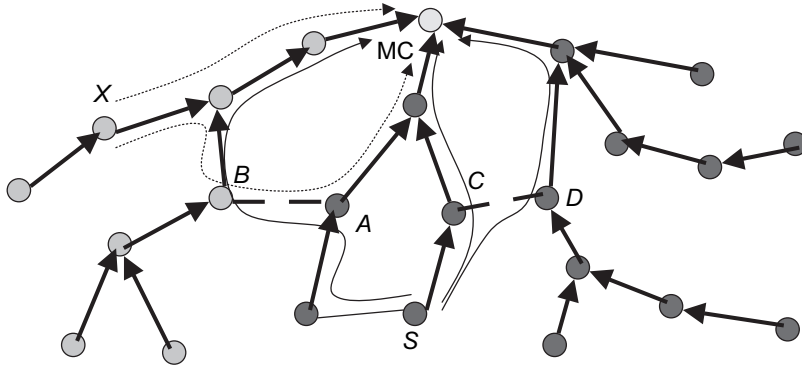


Figure 13.4 Reporting from source S to MC via three alternate paths.

may first be selected so that they make minimal connected-area coverage. The problem is that if a node in the tree fails or is malfunctioning, then the reports can be lost or compromised. To enhance the security of reporting, secondary and ternary paths for reporting are proposed in refs. [33] and [34]. A primary broadcast tree is constructed from the monitoring center (MC). All nodes are labeled based on the first hop out of the MC. Thus the network is effectively partitioned based on which neighbor of the MC delivers the report to it. Now consider an edge AB in the network, where A and B have two different labels. Such edges can serve as bridges for the second copy of messages. Suppose that node S wants to send a report to the MC (see Fig. 13.4). Node S can send one copy of the report directly to the MC using the reverse broadcast tree. The second copy can be sent instead to a node A with the same label as S , which has a neighbor B with a different label. Then the report can be sent from S to A , and from A to B , and from B using the reverse broadcast tree to the MC. The two paths should be disjoint, if possible. Similarly, S may find even a ternary path through another bridge, on the other side of the tree, for even higher security, or to provide the majority consensus to the MC, which can choose two out of three reports. The construction of the secondary and ternary paths may proceed as follows. Both end points A and B of each bridge edge AB can initiate the construction of their private trees within the neighborhood with the same label, in the same way the MC constructs its own broadcast tree. These trees are labeled with the label of the other end point. That is, the private tree of A within the neighborhood of A with the same primary label as A is labeled with the label of B , and vice versa. Each node Q receiving such a message for the first time will join the corresponding broadcast tree, and forward the message so that the tree is enlarged. If the message is already received from another bridge node, with the same “bridge” label, Q ignores it to avoid too much traffic and constructing additional trees that may not significantly enhance security. If a message is received with a different secondary label, Q accepts the participation in yet another tree and enlarges it by rebroadcasting the message. Routing the report then follows the

constructed trees. The source node S sends the primary report on the primary tree, and uses the secondary and perhaps the ternary tree to which it belongs to send additional copies of the same report. Security can be jeopardized with the new method if several nodes close to the sink, one or two hops away, located on several branches of the tree, are all compromised. To prevent that, the sink must be particularly responsible for the authentication of all nodes that are one hop and perhaps two or three hops away from it in the broadcast tree.

13.5 PERIODIC REPORTS BY ALL SENSORS

13.5.1 PEGASIS: Chain of Reporting Nodes

Lindsey, Raghavendra, and Sivalingam [35] proposed a framework for energy-efficient data gathering algorithms in wireless sensor networks. Their power-efficient gathering in sensor information system (PEGASIS) protocol [35] first organizes sensors into a chain, by a centralized algorithm (e.g., the sink can decide about ordering of reports). Thus, sensors are initialized as c_0, c_1, \dots, c_{n-1} . Data gathering is performed in rounds. In round k , first find $i = k \bmod n$. Each round consists of n iterations. In each iteration, only one sensor is sending a message, containing data gathered by that sensor. Iterations are performed as follows: c_0 sends to c_1 , c_1 to c_2, \dots, c_{i-1} to c_i . Then c_{n-1} sends to c_{n-2} , c_{n-2} to c_{n-3}, \dots, c_{i+1} to c_i . Finally, c_i sends the gathered data to the MC. The distance to the MC is assumed to be larger than the distances between the sensors. Chains can be difficult to construct in multi-hop sensor networks. For single-hop networks, initialization algorithm needs to run, or the MC needs to assign reporting indices to individual sensors. Once constructed, when sensors change activity status (from active to passive) or stop functioning, the order scheme needs to run again, or a maintenance procedure is needed. The scheme is also not sensitive to the energy levels of the sensors, as different sensors consume different amount of energy, depending mainly on their distances to the MC.

13.5.2 LMST- and Geocasting-Based Data Gathering

Several localized solutions are proposed in refs. [33] and [34]. One is a localized algorithm that first constructs LMST (or other sparse structure such as the relative neighborhood graph (*RNG*)). Instead of creating a chain, a token is circulated in the network. The node currently having the token will send it to one of its LMST neighbors. This can be done in different ways. Nodes can forward with equal probability of sending to one of its neighbors (not returning to the neighbor it came from). Since the average degree of LMST is about 2.04 [18], there is on average one such neighbor to forward the token. The forwarding probability may depend on node densities. Neighbors with more LMST neighbors should have a smaller probability of getting the token (since they may get tokens from more neighbors in the process). Next, neighbors with more energy left may have a higher probability of getting the token. Finally, in the case of monitoring an event that can be

geographically located, sensors nearby need to preserve more energy, and thus they may decide to postpone reporting to the MC. Thus, instead of reporting every n th time to the MC, the frequency may also be decided probabilistically, depending on the energy level of the node. This scheme does not offer an immediate alternative for the data aggregation. LMST may be converted to MST [36], or a spanning tree may be constructed by broadcasting a message from the node holding the token. The constructed tree may be used for data gathering from other sensors, before the node holding the token sends the report to the MC. MST can also be used for data aggregation, since reports can be sent toward the node holding the token, and aggregated on the way.

Another solution proposed in refs. [33] and [34] is to apply the geocasting algorithm [12,37], which follows a single path from the source while visiting all nodes in the region. The algorithm guarantees to see all nodes, and on average it does so twice during a single geocasting process, which can be repeated periodically. If the sink is fixed, preprocessing can be done to decide the entry edges and reduce communication time, as described in refs. [21] and [25]. The advantage over the solution just described is to guarantee the participation of each node on a fairly regular basis.

13.6 DATA GATHERING WITH DATA AGGREGATION

Data collection, known as data gathering or data dissemination, can be considered as a reverse multicasting task, with all nodes from the multicasting group reporting their data to the MC. There are several cases that need to be distinguished in these tasks. The data collected may or may not be aggregated at intermediate sensor nodes. Data aggregation is applied when sensor measurements are correlated, which is reasonable to assume when they measure the same event in nearby positions. In this case, it is not necessary that each individual report (which may not be sufficiently reliable) reaches the monitoring center. Intermediate nodes may combine (aggregate) data received from several neighbors, and possibly one measured by itself, into a single report, and forward it toward the MC. In the case of data aggregation, a distinction can be based on whether or not forwarding sensors have their own data measured. Obviously, all sensors that want to report data need to be included in the reporting tree. If data aggregation is not applied, then clearly each reporting sensor may apply one of the routing algorithms with guaranteed delivery (e.g., ref. [12]) for sending its report to the MC. If data aggregation is applied, a distinction can be made between protocols applied within the geocasting region, and outside of it. Outside the geocasting region, not all sensors need to participate in reporting. The problem appears, then, to be the inverse of the multicasting problem, that is, the multicast tree that is set while sending the request to the sensors can be used to report data back from the sensors (assuming an on-demand query was issued to all sensors).

13.6.1 Power-Efficient Data-Gathering and Aggregation Protocol

Tan and Korpeoglu [38] proposed a power-efficient data-gathering and aggregation protocol (PEDAP) that assumes that locations of all nodes are known by the sink a priori. The sink constructs a MST which is then used for data gathering and aggregation. In the power-aware version of the same protocol, the MST is constructed with weights on each edge calculated as the product of power consumption on the edge and the reluctance of a neighbor to receive the packet (reluctance is the inverse of the remaining energy at the node).

13.6.2 LMST-Based Data-Aggregation

Inside the geocast region, two cases for data aggregation may occur: all nodes in the region sense the event, or some nodes are there only to forward traffic. If all sensors within the region are reporting, the optimal tree to use is apparently the MST (as observed in ref. [39]). The existing distributed algorithm for constructing a MST require $O(n \log n)$ messages, with a high constant involved. The algorithm presented in ref. [36] is based on breaking all cycles created in the LMST. Each of the LMST's links is broken by identifying the longest edge in it and removing it [36]. The removal of one such edge may lead to a longer cycle, which is broken in the next iteration. The LMST can also be broken differently, in the considered context. The new solution [33,34] is to also start with the *LMST* structure. The (MC) will create a tree out of the LMST by forwarding its request, within the geocasting region, only along the LMST's edges. When a node receives a message from the MC, it will forward it on only on its remaining LMST edges, if any. However, if a neighbour already received a message from the MC, that link is not used; an LMST cycle is broken that way. The obtained tree is not necessarily MST, but its approximation, which is expected to be very close to it in performance. Figure 13.5 illustrates the construction of a LMST-Based data aggregation tree. The edges of the LMST are drawn with thick edges, and all but one (drawn with a dashed line) is included in the broadcasting tree. Edge *WF* is the only one that is in the LMST, but not in the MST.

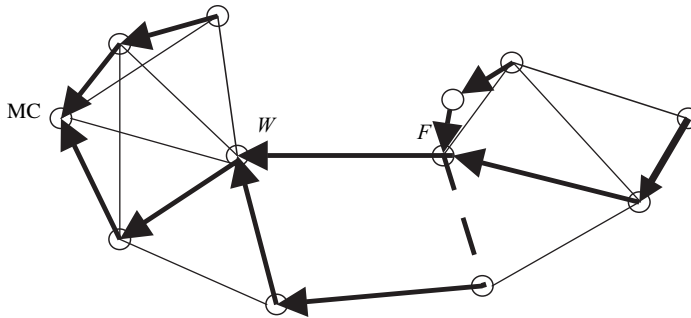


Figure 13.5 LMST-based energy-efficient data-aggregation tree.

13.7 MOBILE SINKS OR SENSORS

13.7.1 Two-Tier Data Dissemination

Data dissemination for large-scale wireless sensor networks was considered in ref. [40], for the case when multiple mobile sources send information constantly to multiple mobile destinations. The authors [40] proposed a two-tier data-dissemination approach, source to destination and destination to source, with grid subdivision of the area and greedy forwarding. The initial routes are set by flooding (if the destination locations are not known) or greedy forwarding (otherwise). Sources create grids that contain sensors that are closest to the grid intersections. These sensors act as a backbone for routing. When sources or sinks move to a new cell, they flood new cells to find a new backbone sensor, so that the existing paths can be extended. This article [40] does not give any concrete scheme for path optimization to avoid using long paths.

The grid backbone assumes large sensor density, with all sensors being active. However, if a sensor-area coverage scheme is applied to select active sensors, the grid backbone can be replaced by a more efficient and more natural backbone of covering sensors (discussed in a separate chapter in this book). Grid division is unnecessary overhead, and greedy forwarding may fail. GFG [12] can replace greedy forwarding. If either sources or sinks are fixed, then mobile components will initiate route maintenance toward stationary components. Let S be a fixed source, and D be the initial position of a destination. Path extension from the new destination position D' toward the old position D can be applied up to a certain traveled distance, then D' can initiate routing toward the source until it reaches it. In fact, the maintenance can stop when a node that already knows the path to S is found. Alternatively, the new path search may terminate after reaching a node A' that is a neighbor of a node A on the original path SAD . Instead of $SADD'$ the new path, is then $SAA'D'$ [33,34].

Further optimizations can be achieved by merging some reporting streams toward same sink or the same source. Suppose that messages from sensors A and B , sent toward the same sink or source, are heard by sensor C , their common neighbor. Sensor C can then offer to merge these streams, reporting its position. Nodes A and B evaluate the gain obtained by each of such candidates C , in the case of several such offers, and select the best one [33,34]. Note that A and B may or may not be neighbors themselves, which results in two different protocols. The problem exists when both sources and sinks can be mobile, since then the updates do not have precise destinations. The procedure then is an alternate the application of the described procedure from both ends until the packets meet somewhere in the network.

13.7.2 Mobile Collectors

Tirta, Li, Lu, and Bagchi [41] proposed using a mobile collector, such as an airplane or a vehicle, to collect sensor data from remote fields. The sensor network is clustered and only CHs report data. They present three different schedules for the collector.

In the *round-robin* scheme, each CH is visited in a predefined order, regularly for same amount of time. In the *rate-based* scheme, the frequency of visits depends on the amount of data reported. In the *min-movement* scheme, CHs are visited in specific order, but the time spent with each of them depends on the amount of data to report (more precisely, the collector stays with each CH until all data are collected).

13.7.3 Mobile Sensors

Taherian and O’Keefe [42] proposed an energy-aware event-dissemination protocol for mobile sensor networks. In this protocol, each sink proactively constructs a redundant tree in the network. This redundant tree is combined with probabilistic forwarding. The main idea is to limit the number of parent and sibling nodes as the redundant tree grows. The proposed redundant tree is not guaranteed to be connected, that is, be useful for event dissemination. It is also not guaranteed to provide full coverage of the sensor network area. Routing follows tree pointers if they exist; otherwise, it applies a probabilistic forwarding scheme (which is similar to existing beaconless routing schemes reviewed in Chapter 12).

13.8 TRACKING OBJECTS IN SENSOR NETWORKS

13.8.1 Tracking Objects Without Data Aggregation

We now discuss problems associated with tracking an object, possibly moving. This section considers problem aspects when data aggregation is not involved. Each report therefore needs to be sent directly to the base station. The problem is to send a sufficient number of reports so that the position of the object can be reliably determined by the base station, while minimizing the number of sensors that send the report. We assume that each sensor knows its own location, and location of its neighboring sensors. Although in reality the reliability of sensor observation depends on the distance to the object, we consider a simplified model, assuming that all sensors located within the sensing radius from the object can reliably detect it. After detecting the object, sensors can measure either the distance to it, or the direction (angle) toward it. The case of distance is similar to the position determination problem, discussed in Chapter 9. We therefore now study the case of measuring an angle toward an object, without knowing the distance to it. All reports will be assumed accurate, although in reality some reports may be false, and security and report reliability issues need to be studied as well. Estrin, Govindan, Heidemann, and Kumar [2] were the first to investigate this problem. In their solution, the sensor network is clustered first. CHs for which all neighboring CHs lie on the same side of a line drawn toward the object elect themselves to participate in object location. The goal is to elect sensors that form the longest baseline for triangulation. There are several problems with this approach. In general, there are two such nodes, which are tangent nodes from the object to the convex hull of CHs. If one of them for any reason does not see the object (because of obstacles),

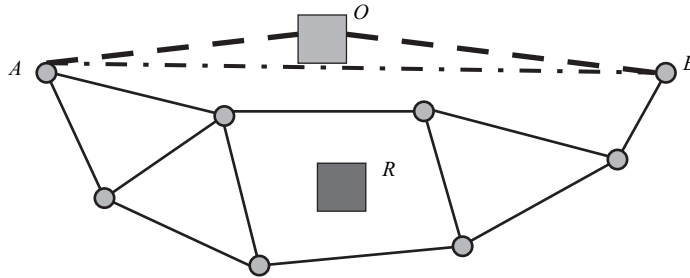


Figure 13.6 Longest baseline may not accurately determine the location of the object O ; the object R is inside the convex hull of the observing sensors.

the object cannot be accurately determined. Next, the longest baseline is not always the best choice for accuracy (see Fig. 13.6 where the object R is drawn as a square). Small or large angles in the triangle containing the baseline may cause large errors in object positioning. Short and long baselines, with respect to other network measurements, also cause large errors. What is a better criterion? It appears that it is better to maximize the minimal angle in the triangle used to decide object location [43].

The next problem occurs when the object is inside the convex hull of participating CHs, as object R in Figure 13.6. In this case, no sensor reports the direction of the object in ref. [2]. Therefore, there is a need for designing new protocols. In fact, the problem appears to be quite challenging. Some proposals were given in ref. [43] as follows. Each node can decide whether or not it is locally a northernmost (N), easternmost (E), westernmost (W), or southernmost (S) node, by verifying whether or not any sensor exists on the corresponding horizontal or vertical line passing through the sensor. Similarly, NE, NW, SE and SW sensors can be found, by considering directions $\pi/4$ and $3\pi/4$. Every sensor that can see the object could report it to its locally northernmost node, by sending/forwarding a message to its northernmost neighbor, using the greedy routing or routing scheme [12]. The exact protocol details and analysis are still under investigation [43]. The best approach appears to be to use the multilateration technique, which is applied, for instance, in ref. [4] for position determination based on distances. It is also interesting to consider the different ways of selecting which sensors will participate. In addition, clustering organization could be replaced by another backbone, for example, connected dominating sets. Any type of backbone could be applied on an area covering the set of sensors.

13.8.2 Tree Reconfiguration for Tracking Mobile Objects

Zhang and Cao [44] discussed how to monitor an object by sensors located inside a monitoring region, such as a circle. These sensors are organized into a tree, with one of the sensors serving as the root. The root collects all reports, aggregates them, and routes them to one or more sinks (base stations). In this method, the root keeps

monitoring its distance to the target. When the distance becomes larger than a certain threshold d , it will be replaced by the node that is closest to the center of the current monitoring region.

13.8.3 Mobicast Protocol for Tracking Mobile Objects

One particular application of geocasting is tracking mobile objects. Mobile objects create geocast regions that are time dependent, and data collection is performed by the sensors in the vicinity of a moving object. The sink may collect reports from the sensors in the vicinity of the object, and may send periodic signals to the sensors adjusting the geocasting region, following the trajectory along with the object advances. In mobicast application [45], however, the sensors themselves adjust the geocasting region.

Huang, Lu, and Roman [45,46] proposed a mobicast protocol where the nodes that belong to the forward, time-dependent region, or belong or are about to enter the geocast region, retransmit the message. Their algorithm presented in ref. [45] is an improved version of the one in ref. [46]. In their problem formulation, the MC is not used to decide and inform about the geocast region. Instead, the sensors themselves cooperate, follow the movement of an object, and inform the sensors, which are approached by the object, to start monitoring. This is achieved by considering the planar graph of covering sensors, and forwarding messages to the faces of the planar graph in the direction of object movement, with proper timing corresponding to the arrival time of the object at the considered face. This is illustrated in Figure 13.7. Suppose that the rectangle object, shown with dashed lines, moved from the far left and is continuing toward the far right. All faces that the object intersected are traversed by messages from sensors in these faces. They are marked in Figure 13.7 by clockwise arrows along the face edges (following the

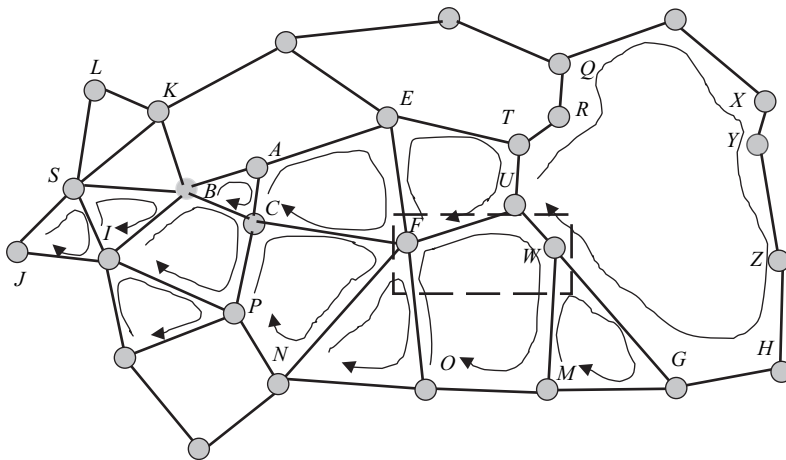


Figure 13.7 Reliable mobicast.

left-hand rule for face traversal). Node F , after receiving the signal from neighbors C , E , or N , starts monitoring. It then estimates when the rectangle will reach two other faces containing F , and will send a signal at the appropriate time, so that nodes in these faces are alerted in time. In turn, node W decides when the rectangle will reach that face, and sends the alert signal at the best possible time, so that all the nodes on the long face are alerted just before the arrival of the rectangle. In that way, alert messages are able to cross over obstacles, or areas without sensors. For instance, node Z will be alerted although there is no direct neighbor of it that monitored the same rectangle before the rectangle reached Z . In the algorithmic details given by the authors [45], each node needs to learn all nodes that are on the same face as the given node, which may not always be available from local knowledge. However, we believe that this may be avoided by slightly changing the description. Proper timing for reaching node Z is also an issue, since the face size may be significantly larger than the average one, and the message may even follow the outer boundary of the network. Face routing may not be necessary for alerting sensors ahead of the arriving object. Greedy routing may be used, or, more precisely, GFG [12]; it is its combination with face routing that guarantees delivery. If the area is convex, completely covered by sensors, and the communication radius is over twice as large as the sensing area, ref. [47] showed that greedy routing guarantees delivery. The problem can even be considered as a variant of geocasting, with sensors themselves issuing geocast messages toward the future location of the object, alerting sensors in a certain area. If several such messages arrive at a given node, it will forward only one of them, to control the overhead.

13.9 RATE-BASED DATA PROPAGATION IN SENSOR NETWORKS

In sensor networks, data sources sample data and propagate them to potential consumers. A consumer may subscribe to a data item at a certain rate, and the subscription rate may vary for different consumers. The problem is to construct a data propagation tree to efficiently disseminate data from a source node at the required rate to each of its consumers.

Singh, Pujar, and Das [48] proposed a breadth-first search-based protocol for a one-to-one network model, assuming that the connections between sources and consumers are with a wired network. We shall consider a sensor network, that is, a wireless network, for connection between sources and consumers. In this scenario, communication is one-to-all. The main difference from Singh et al.'s model [48] is that the rates are marked at each node, instead of at each edge.

The proposed solution [49] is a generalized multicasting position-based protocol (following a solution proposed in ref. [27]), which applies the *multipoint relay (MPR)* strategy (see Chapter 11) in determining forwarding neighbors and their rates.

In a preprocessing step, each consumer reports to the source (assume that there is only one source, for simplicity). Thus the source is informed about customers and their preferred rates. The source then creates a list of customers and their rates and makes forwarding decisions, that is, which neighbors will retransmit and at

what rates. Other nodes, receiving decisions to retransmit, will follow a very similar protocol to determine their forwarding nodes.

Let S be the current node, and let C_1, C_2, \dots, C_n be consumers that S needs to serve. Let R_1, R_2, \dots, R_n be their preferred rates. Let A_1, A_2, \dots, A_m be neighbors of S . For each consumer, only neighbors that are closer to it than S can be considered for serving and covering. Each possible covering of consumer C_i by neighbor A_j ($|A_j C_i| < |S C_i|$) is associated with the cost per progress, defined as $R_i / (|S C_i| - |A_j C_i|)$. More precisely, since A_i , once selected, will cover all consumers it is closer, the cost of selecting A_i is associated with the progress it makes toward all consumers it could serve (it is closer than S to them). To avoid notational difficulties, let B_1, B_2, \dots, B_k be those consumers among C_1, C_2, \dots, C_n that are considered for serving by A_i (they are all closer to A_i than to S , but not all such nodes need to be selected). Let P_1, P_2, \dots, P_k be their corresponding preferred rates. Then the cost per progress for selecting A_i is $\max(P_1, P_2, \dots, P_k) / (|S B_1| - |A_i B_1| + |S B_2| - |A_i B_2| + \dots + |S B_k| - |A_i B_k|)$. An alternative measure is to consider each progress individually: $P_1 / (|S B_1| - |A_i B_1|) + P_2 / (|S B_2| - |A_i B_2|) + \dots + P_k / (|S B_k| - |A_i B_k|)$. However, this is not likely to be a better criterion, since one small progress can easily undermine a number of good progresses made. The selection of covering neighbors and their rates then can proceed in the following manner:

- If there is any consumer served by a single neighbor, then that neighbor is selected; moreover, the selected neighbor will also cover other consumers that are closer to it than to S .
- Select one of the remaining consumers with maximal preferred rate, and consider the cost of each neighbor serving it, and the additional benefit such a choice makes overall. Select the node that then minimizes its own $\max(P_1, P_2, \dots, P_k) / (|S B_1| - |A_i B_1| + |S B_2| - |A_i B_2| + \dots + |S B_k| - |A_i B_k|)$.
- Repeat previous step until all consumers are covered.

13.10 ANONYMITY ISSUES IN WIRELESS SENSOR NETWORKS

In many applications, safeguarding *output data assets*, that is, data produced by the wireless sensor network and consumed by the end user (application), against loss or corruption is a major security concern. In these application domains, a wireless sensor network is typically deployed in a hostile target environment for a relatively long period of time. The network self-organizes and works to generate output data that is of import to the application. For example, a wireless sensor network may be deployed across a vast expanse of enemy territory ahead of a planned attack; the network system monitors the environment and produces reconnaissance data that are *absolutely essential* to a mission planning application. Periodically, during the network lifetime, a mobile gateway, mounted on a person, land or airborne vehicle, or a satellite, collects the output data assets from the network system, to maintain an up-to-date state. This means the network system must store the output data assets

from the time it is produced until it is collected. Therefore, securing the output data assets in the network is an important problem in this class of applications.

We view an attack on the output data assets in the sensor network as a type of denial of service attacks. This view is based on the abstraction that output data are stored in a logical *repository*, and that *access* to this output data repository constitutes, in effect, a “service” provided by the network system to the application; corruption or loss of output data denies the application access to that service.

13.10.1 What Is Anonymity?

Anonymity protects the identity of the sender or receiver and guarantees that both parties involved in a communication remain anonymous to each other. Recent years have seen a flurry of activity, and many anonymous communication systems have been developed for the Internet. Most of the work on anonymity is concerned with *sender* anonymity, *receiver* anonymity, and *mutual* anonymity. Quite recently, *traffic* anonymity has also received well-deserved attention in the literature. Recently, the problem of securing ad hoc networks has received a great deal of well-deserved attention. To the best of our knowledge, the anonymity problem has not been adequately addressed in wireless sensor networks [50].

The threat model assumed by Wadaa et al. [50] comes from a data-centric view of wireless sensor networks. The model is predicated on the assumption that *the end-goal of anonymity attacks on the wireless sensor network is to identify and eliminate the minimum number of sensors to inflict maximum loss of data assets*; eliminating a sensor means disabling it so that it is permanently nonoperational. For any operation cycle, if a sink suffers a permanent failure before transferring the contents of its data repository to the gateway, then a portion of the data assets corresponding to the cycle is irrevocably lost. The goal of the adversary is to eliminate all sinks. This can be accomplished in two ways.

13.10.2 The Anonymity Threat Model

13.10.2.1 Brute-Force (Sink Nodes Not Identified) This can take the form of randomly eliminating nodes in the network on the assumption that, statistically, some sinks will be eliminated in the process. Coarse sink granularity and sink redundancy mitigate the risk of loss of data assets as a result of this type of attack. A straightforward special case is the massive elimination of all sensors in the network.

13.10.2.2 Smart (Sink Nodes Identified) The adversary analyzes network traffic to deduce information about topology, traffic flow patterns, and other system attributes. The goal is to discover sink nodes and to eliminate them. In this chapter we assume the adversary engages in smart elimination attacks. The specifics of the architecture and the implementation of the adversary system are assumed to be unknown.

13.10.3 Sender and Path Anonymity

Sender anonymity is most commonly achieved by transmitting a message to its destination through one or more intermediate nodes in order to hide the true identity of the sender. The message is thus effectively *rerouted* along what is called a *rerouting path*. It is important to study rerouting-based anonymous communication systems in terms of their ability to protect sender anonymity. The selection of rerouting paths is critical for this kind of system. Olariu et al. [51] investigated how different path selection strategies affect the ability to protect sender anonymity. For a given anonymous communication system, they measure this ability by determining how much uncertainty this system can provide in order to hide the true identity of a sender. They call this measure the *anonymity degree*. In ref. [51] the authors assume a *passive adversary model*: the adversary can compromise *one or more* nodes in the system. An adversary agent at such a compromised node can gather information about messages that traverse the node. If the compromised node is involved in the message rerouting, it can discover and report the immediate predecessor and successor nodes for each message traversing the compromised node. We assume that the adversary collects all the information from its agents at the compromised nodes and attempts to derive the identity of the sender of a message.

Common sense indicates that the degree of anonymity increases with increasing number of intermediate nodes between the sender and the receiver. Olariu et al. [51] call this number of intermediate nodes the *path length* of the rerouting path. There is a point, however, beyond which increasing the path length actually *decreases* the degree of anonymity. The authors give a quantitative analysis of how path length affects the degree of anonymity. Rerouting schemes give rise either to paths with *fixed length* (where messages are forwarded to the receiver after traversing a fixed number of intermediate nodes) or *variable length* (for example, where every intermediate node randomly decides whether to forward the message to the receiver directly or to another intermediate node). The authors show that variable path-length strategies perform better than fixed path-length strategies in terms of degree of anonymity. However, when the expected path length is sufficiently long, the difference of anonymity degree is relatively small between different variable and fixed path-length strategies. As a result of this study, Olariu et al. [51] argue that several well-known anonymous communication systems are not using the best path selection strategies. They go on to propose an optimal method to select path lengths, by showing that the path selection problem can be cast as an optimization problem, whose solution yields an optimal path-length distribution that maximizes the degree of anonymity.

13.11 CONCLUSIONS

We considered some relevant aspects of the process of issuing requests and collecting data, with sensor ad hoc networks as the primary application of the presented methods. Protocol efficiency was the primary goal, with efficiency defined by some metrics or design characteristics (such as localized behavior of protocols).

Ad hoc and sensor networks have recently attracted exponentially increasing interest, including the creation of new conferences, new journals, and publication of a number of books. We envision that this trend will continue in the short term, and we envision that data-centric operation problems, discussed in this chapter, will continue to be intensively studied. We hope that the research efforts will lead toward real applications of ad hoc networks, especially sensor networks.

Sensor networks pose a number of research challenges. In addition to the problems discussed in this and other chapters in this book, we mention two more problem areas. One is about the design of sensor network protocols for heterogeneous sensor networks, the other is the investigation of various scenarios and protocols for wireless sensor and actor networks. Actor nodes are active nodes, with higher energy and computation capabilities, that are able to perform some actions and are able to move around.

EXERCISES

- 13.1. Describe a routing algorithm based on sector training [1] that will guarantee delivery in when there are empty sectors.
- 13.2. Derive a formula for the error involved when the position of an object is determined based on the angles measured from three given sensors [43]. Show that the error is minimized when the minimal angle in the triangle created by three measuring sensors is maximized.
- 13.3. Prove that the *geocast_traversal_intersecting_faces* algorithm guarantees delivery to all nodes inside the geocasting region, which is connected to the source [21,25].
- 13.4. Design an algorithm for finding optimal ring sizes (for extending network lifetime) for reporting to a CH with data aggregation for the case of n rings [49].
- 13.5. Design some protocols for moving the sink to a new position near the old position so that the overall energy consumption for reporting from last-hop sensors is reduced. Design another procedure for moving the sink to reduce the number of incoming reports that violate delay constraints [52].
- 13.6. Design an energy-efficient data-aggregation protocol for the following scenario. There are two types of sensors in a geocasting region, plus sensors outside the region. Some sensors inside the geocasting region are sensing and can perform data aggregation, while some other sensors are not sensing, but can only perform data aggregation, if needed. Sensors outside the geocasting region can only perform data aggregation, if needed, or can simply forward the traffic.
- 13.7. In a heterogeneous sensor network, there are two kinds of sensors. Some super-sensors have high-energy resources and can communicate with each other and with the sink with much smaller delays than communication between regular sensors. Suppose that each node knows the distance to and label of the nearest

supersensor, and that this information is communicated to neighboring sensors. Describe a broadcasting protocol in this environment [32].

- 13.8.** Describe a localized protocol for general multigeocasting problems, where a monitoring task is to be disseminated from the sink to all the sensors located inside several geographic regions that are of arbitrary sizes, shapes, and locations, known to the sink.

ACKNOWLEDGMENT

This research is partially funded by NSERC Discovery grant.

REFERENCES

1. S. Olariu, A. Wadaa, L. Wilson, and M. Eltoweissy. Wireless sensor networks: Leveraging the virtual infrastructure. *IEEE Network*, pages 51–56, July/August 2004.
2. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, pages 263–270, Seattle, Washington, August 1999.
3. W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)*, Volume 8, page 8020, Maui, Hawaii, January 2000.
4. R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proceedings of the 2nd International Workshop in Information Processing in Sensor Networks (IPSN '03)*, pages 333–348, Palo Alto, California, April 2003.
5. V. Mhatre and C. Rosenberg. Design guidelines for wireless sensor networks: Communication, clustering and aggregation. *Ad Hoc Networks*, forthcoming.
6. S. Olariu and I. Stojmenovic. Design Guidelines for Clustering and Aggregation in Sensor Networks. In preparation.
7. I. Stojmenović and X. Lin. Power aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, **12**(11):1122–1133, November 2001.
8. J. Lian, K. Naik, and G. B. Agnew. Data capacity improvement of wireless sensor networks using non-uniform sensor distribution. *International Journal of Distributed Sensor Networks*, forthcoming.
9. M. Kumar, L. Schwiebert, and M. Brockmeyer. Efficient data aggregation middleware for wireless sensor networks. Paper presented at the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004), Fort Lauderdale, Florida, October 2004.
10. C. Schurgers and M. Srivastava. Energy efficient routing in wireless sensor networks. In *Proceedings of MILCOM 2001*, pages 357–361, Vienna, Virginia, October 2001.

11. I. Chatzigiannakis and S. Nikolettseas. A sleep-awake protocol for information propagation in smart dust networks. In *Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS 2003)*, page 225, Nice, France, April 2003.
12. P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (D/ALM'99)*, pages 48–55, Seattle, Washington, August 1999. See also in *Wireless Networks*, 7(6):609–616, 2001.
13. S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, S. Shenker, L. Yin, and F. Yu. Data-centric storage in Sensornets with GHT, a geographic hash table. *Mobile Networks and Applications (MONET)*, 8:427–442, August 2003.
14. C. Intanagonwiawat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*, pages 56–67, Boston, Massachusetts, August 2000. See also *IEEE/ACM Transactions on Networking*, XX:xxx–xxx, 2003.
15. J. Ding, K. M. Sivalingam, R. Kashyapa, and L. J. Chuan. A multi-layered architecture and protocols for large-scale wireless sensor networks. In *Proceedings of the IEEE Vehicular Technology Conference (VCT2003)*, Orlando, Florida, October 2003.
16. I. Stojmenovic. Beaconless Area Based Broadcasting. In preparation.
17. J. Lipman, P. Boustead, and J. Chicharo. Reliable minimum spanning tree flooding in ad hoc networks. *IEEE Transactions on Vehicular Technology*, forthcoming.
18. N. Li, J. C. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. In *Proceedings of IEEE INFOCOM*, Volume 3, pages 1702–1712, San Francisco, California, April 2003.
19. W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pages 174–185, Seattle, Washington, August 1999.
20. M. Seddigh, J. Solano Gonzalez, and I. Stojmenovic. RNG and internal node based broadcasting algorithms for wireless one-to-one networks. *Mobile Computing and Communications Review*, 5(2):37–44, 2001.
21. I. Stojmenovic. *Geocasting in Ad Hoc and Sensor Networks*. Technical Report TR-2004-02, Computer Science, SITE, University of Ottawa, March 2004. See also in *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Jie Wu (ed.), CRC Press, forthcoming.
22. Y. Yu, R. Govindan, and D. Estrin. *Geographic and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks*. Technical Report TR-01-0023, Computer Science, University of California, Los Angeles, August 2001.
23. I. Stojmenović, M. Russell, and B. Vukojevic. Depth first search and location based localized routing and QoS routing in wireless networks. *Computers and Informatics*, 21(2):149–165, 2002.
24. J. Heidemann, F. Silva, and D. Estrin. Matching data dissemination algorithms to application requirements. In *Proceedings of the 1st International Conference on Embedded Networked Sensor System (SenSys)*, pages 218–229, Los Angeles, California, November 2003.

25. I. Stojmenovic. Geocasting with guaranteed delivery in sensor networks. *IEEE Wireless Communications Magazine*, December 2004, to appear.
26. K. Saeda and A. Helmy. Efficient geocasting with perfect delivery in wireless networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2004)*, Volume 5, pages 2555–2560, Atlanta, Georgia, March 2004.
27. M. Mauve, H. Fusler, J. Widmer, and T. Lang. *Position-Based Multicast Routing for Mobile Ad Hoc Networks*, Technical Report TR-03-004, Department of Computer Science, University of Mannheim, March 2003. See also Poster: Position-based multicast routing for mobile ad-hoc networks, In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)* (electronic edition), Annapolis, Maryland, June 2003.
28. J. Carle and D. Simplot-Ryl. Energy efficient area monitoring by sensor networks. *Computer (IEEE)*, **37**(2):40–47, February 2004.
29. C. Perkins and E. M. Royer. Ad hoc on-demand distance vector (AODV) routing. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pages 90–100, New Orleans, Louisiana, February 1999.
30. T. Fujiwara, N. Iida, and T. Watanabe. An ad hoc routing protocol in hybrid wireless networks for emergency communications. In *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops—W6: WWAN (ICDCSW '04)*, pages 748–754, Tokyo, Japan, March 2004.
31. S. Zou, I. Nikolaidis, and J. J. Harms. Efficient data collection trees in sensor networks with redundancy removal. In *Proceedings of the 3rd International Conference on AD-HOC Networks and Wireless (ADHOC-NOW 2004)*, pages 252–265, Vancouver, British Columbia, July 2004.
32. F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Routing and broadcasting in hybrid ad hoc and sensor networks. In *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Jie Wu (ed.), CRC Press, forthcoming.
33. I. Stojmenović. *Geocasting, Data Gathering and Activity Scheduling in Ad Hoc and Sensor Networks*. Technical Report TR-2003-05, Computer Science, SITE, University of Ottawa, August 2003.
34. I. Stojmenović. Data Gathering and Activity Scheduling in Ad Hoc and Sensor Networks. Paper presented at the International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks, Chicago, Illinois, June 2004.
35. S. Lindsey, C. Raghavendra, and K. Sivalingam. Data gathering algorithms in sensor networks using energy metrics. *IEEE Transactions on Parallel and Distributed Systems*, **13**(9):924–935, September 2002.
36. F. J. Ovalle-Martinez, I. Stojmenovic, F. Garcia-Nocetti, and J. Solano-Gonzalez. Finding minimum transmission radii and constructing minimal spanning trees in ad hoc and sensor networks. *Journal of Parallel and Distributed Computing*, forthcoming.
37. P. Morin. Online Routing in Geometric Graphs. Ph.D. thesis, School of Computer Science, Carleton University, January 2001.
38. H. O. Tan and I. Korpeoglu. Power efficient data gathering and aggregation in wireless sensor networks. *ACM SIGMOD Record*, **32**(4):66–71, December 2003.
39. M. Khan, G. Pandurangan, and B. Bhargava. *Energy-Efficient Routing Schemes for Sensor Networks*, Technical Report CSD TR 03-013, Purdue University, July 2003.

40. F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the 8th International ACM Conference on Mobile Computing and Networking (MobiCom)*, pages 148–159, Atlanta, Georgia, September 2002.
41. Y. Tirta, Z. Li, Y. H. Lu, and S. Bagchi. Efficient collection of sensor data in remote fields using mobile collectors. In *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN 2004)*, pages 515–520, Chicago, Illinois, October 2004.
42. S. Taherian and D. O’Keefe. Event dissemination in mobile wireless sensor networks. Paper presented at the 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2004), Fort Lauderdale, Florida, October 2004.
43. I. Stojmenovic. Object Location in Sensor Networks. In preparation.
44. W. Zhang and G. Cao. Optimizing tree reconfiguration to track mobile targets in sensor networks. *Mobile Computing and Communications Review*, 7(3):39–40, July 2003.
45. Q. Huang, C. Lu, and G. C. Roman. Reliable mobicast via face-aware routing. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, March 2004.
46. Q. Huang, C. Lu, and G. C. Roman. *Mobicast: Just-in-Time Multicast for Sensor Networks under Spatiotemporal Constraints*, Technical Report TR WUCS-02-42, Washington University, St. Louis, Missouri, December 2002.
47. G. Xing, C. Lu, R. Pless, and Q. Huang. Greedy geographic routing is good enough in sensing covered networks. *IEEE INFOCOM 2004*.
48. G. Singh, S. Pujar, and S. Das. Rate-Based Data Propagation in Sensor Networks. Paper presented at the IEEE Wireless Communications and Networking Conference (WCNC 2004), Atlanta, Georgia, March 2004.
49. I. Stojmenovic. Rate Based Data Propagation in Sensor Networks. In preparation.
50. A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones. On providing anonymity in wireless sensor networks. In *Proceedings of the 10th International Conference on Parallel and Distributed Systems (ICPADS-2004)*, pages 411–418, Newport Beach, California, July 2004.
51. S. Olariu, A. Wadaa, L. Wilson, K. Jones, and M. Eltoweissy. Enforcing Anonymity in Wireless Sensor Networks. In preparation.
52. K. Akkaya, M. Younis and M. Bangad. Sink repositioning for enhanced performance in wireless sensor networks. *Computer Networks*, forthcoming.