CHAPTER 4

A Virtual Infrastructure for Wireless Sensor Networks

STEPHAN OLARIU and QINGWEN XU
Old Dominion University, Norfolk, Virginia

ASHRAF WADAA
Intel Corporation, Hillsboro, Oregon

IVAN STOJMENOVIĆ
University of Ottawa, Ontario, Canada

Overlaying a virtual infrastructure over a physical network is a time-honored strategy for conquering scale. There are, essentially, two approaches for building such an infrastructure. The first is to design the virtual infrastructure in support of a specific protocol, routing, for example. However, more often than not, the resulting infrastructure is not useful for other purposes. The alternative approach is to design the general-purpose virtual infrastructure with no particular protocol in mind. The challenge, of course, is to design the virtual infrastructure in such a way that it can be leveraged by a multitude *of different protocols.*

The main goal of this chapter is to propose a lightweight and robust virtual infrastructure for a network, consisting of tiny energy-constrained commodity sensors massively deployed in an area of interest. In addition, we present evidence that the proposed virtual infrastructure can be leveraged by a number of protocols ranging from routing to data aggregation.

4.1 INTRODUCTION

Recent advances in nanotechnology have made it possible to develop a large variety of microelectromechanical systems (MEMS), miniaturized low-power devices that integrate sensing, special-purpose computing, and wireless communications

Handbook of Sensor Networks: Algorithms and Architectures, Edited by I. Stojmenović
ISBN 0-471-68472-4

capabilities [1–5]. It is expected that these small devices, referred to as *sensors*, will be mass-produced, making their production cost-negligible. Individual sensors have a small, nonrenewable energy supply and, once deployed, must work unattended. For most applications, we envision a massive deployment of sensors, perhaps in the thousands or even tens of thousands [6–9].

Aggregating sensors into sophisticated computational and communication infrastructures, called *wireless sensor networks*, will have a significant impact on a wide array of applications, ranging from military, to scientific, to industrial, to health care, to domestic, establishing ubiquitous wireless sensor networks that will pervade society, redefining the way in which we live and work [10–13]. The novelty of wireless sensor networks and their tremendous potential for relevance to a multitude of application domains has triggered a flurry of activity in both academia and industry. We refer the reader to refs. [7,14–19] for a summary of recent applications of wireless sensor networks.

The fundamental goal of a sensor network is to produce, over an extended period of time, globally meaningful information from raw local data obtained by individual sensors. Importantly, this goal must be achieved in the context of prolonging as much as possible the useful lifetime of the network and ensuring that the network remains highly available and continues to provide accurate information in the face of security attacks and hardware failure. The sheer number of sensors in a sensor network combined with the unique characteristics of their operating environment (anonymity of individual sensors, limited energy budget, and a possibly hostile environment), pose unique challenges to the designers of protocols. For one thing, the limited energy budget at the individual sensor level mandates the design of ultralightweight data gathering, aggregation, and communication protocols. An important guideline in this direction is to perform as much local data processing at the sensor level as possible, avoiding the transmission of raw data through the sensor network.

Recent advances in hardware technology are making it plain that the biggest challenge facing the wireless sensor network community is the development of ultralightweight communication protocols ranging from training, to self-organization, to network maintenance and governance, to security, to data collection and aggravation, to routing [12,20,21].

4.1.1 The Name of the Game: Conquering Scale

Overlaying a virtual infrastructure over a physical network is a time-honored strategy for conquering scale. There are, essentially, two approaches to this exercise. The first is to design the virtual infrastructure in support of a specific protocol. However, more often than not, the resulting infrastructure is not useful for other purposes. The alternate approach is to design a general-purpose virtual infrastructure with no particular protocol in mind. The challenge, of course, is to design the virtual infrastructure in such a way that it can be leveraged by a *multitude* of different protocols [22].

To the best of our knowledge, research studies addressing wireless sensor networks have thus far taken only the first approach. To wit, in ref. [15] a set of

paths is dynamically established as a result of the controlled diffusion of a query from a source node into the network. Relevant data are routed back to the source node, and possibly aggregated, along these paths. The paths can be viewed as a form of data-dissemination and aggregation infrastructure. However, this infrastructure serves the sole purpose of routing and data aggregation, and it is not clear how it can be leveraged for other purposes. A similar example is offered by ref. [23], where sensors use a discovery procedure to dynamically establish secure communications links to their neighbors; collectively, these links can be viewed as a secure communications infrastructure. As before, it is not clear that the resulting infrastructure can be leveraged for other purposes.

We view the principal contribution of this chapter at the conceptual level. Indeed, we introduce a simple and natural general-purpose virtual infrastructure for wireless sensor networks, consisting of a massive deployment of anonymous sensors. The proposed infrastructure consists of a dynamic coordinate system and a companion clustering scheme. We also show that the task of endowing the wireless sensor network with the virtual infrastructure—a task that we shall refer to as *training*—can be performed by a protocol that is at the same time lightweight and secure. In addition, we show that a number of fundamental tasks, including routing and data aggregation, can be performed efficiently once the virtual infrastructure is in place.

The remainder of this chapter is organized as follows: Section 4.2 discusses the sensor model used throughout the work. Section 4.3 discusses wireless sensor networks, as a conglomerate of individual sensors that have to self-organize and self-govern. In particular, we discuss interfacing wireless sensor networks with the outside world, as well as a brief preview of the training process. Next, Section 4.4 offers a brief overview of location awareness in wireless sensor networks. We also provide a lightweight protocol allowing the sensors to acquire fine-grain location information. Section 4.5 presents an overview of the general-purpose virtual infrastructure for wireless sensor networks. Specifically, Subsection 4.5.1 discusses the details of our dynamic coordinate system, the key component of our general-purpose virtual infrastructure; and Subsection 4.5.2 discusses the clustering scheme induced by the dynamic coordinate system. Section 4.6 is the backbone of the entire chapter, presenting the theoretical underpinnings of the training process. Section 4.8 proposes routing and data-aggregation algorithms in a trained wireless sensor network. Section 4.9 takes a close look at the problem of energy expenditure related to routing data in a wireless sensor network. Finally, Section 4.10 offers concluding remarks and maps out areas for future investigations.

4.2 THE SENSOR MODEL

We assume a sensor to be a device that possesses three basic capabilities: sensory, computation, and wireless communication. The sensory capability is necessary to acquire data from the environment; the computational capability is necessary for aggregating data, processing control information, and managing both sensory and communication activity. Sensor clocks drift at a bounded rate allowing only

short-lived and group-based synchronization, where a group is loosely defined as the collection of sensors that *collaborate* to achieve a given task. The details of a lightweight synchronization protocol for wireless sensor networks will be the subject of another chapter in this book.

We assume that individual sensors operate subject to the following fundamental constraints:

- Sensors are *anonymous*—they do not have fabrication-time identities.
- Sensors are tiny, commodity devices that are mass-produced in an environment where testing is a luxury.
- Each sensor has a nonrenewable energy budget; when the on-board energy supply is exhausted, the sensor becomes nonoperational.
- In order to save energy, each sensor is in *sleep* mode most of the time, waking up at random points in time for short intervals under the control of an internal timer.
- Each sensor has a modest transmission range, perhaps a few meters. This implies that outbound messages sent by a sensor can reach only the sensors in its proximity, typically a small fraction of the sensors deployed.
- Once deployed, the sensors must work *unattended*, it is either infeasible or impractical to devote attention to individual sensors.

At any point in time, a sensor, will be engaged in performing one of a finite set of possible operations, or will be asleep. Example operations are sensing (data acquisition), routing (data communication; sending or receiving), and computing (e.g., data aggregation). We assume each operation performed by a sensor consumes a known fixed amount of energy and that a sleeping sensor performs no operation and consumes essentially no energy.

It is worth mentioning that while the energy budget can supply short-term applications, sensors dedicated to work over years may need to scavenge energy from the ambient environment. Indeed, it was shown recently that energy scavenging from vibration, kinetics, magnetic fields, seismic tremors, pressure, and so on, will become reality in the near future [24,25].

4.2.1 Genetic Material

We assume that just prior to deployment (perhaps onboard the aircraft that drops them in the terrain) the sensors are injected with the following *genetic material*:

- A standard public-domain pseudorandom number generator
- A set of *secret* seeds to be used as parameters for the random number generator
- A perfect hash function ϕ
- An initial time, at which point all the clocks are synchronous; later, synchronization is lost due to clock drift

The way in which this genetic material is used by individual sensors will be discussed in detail later in the chapter. For a more detailed discussion and applications to securing sensor networks we refer the interested reader to refs. [26] and [27].

4.3 STRUCTURE AND ORGANIZATION OF A WIRELESS SENSOR NETWORK

We envision a massive deployment of sensors, perhaps in the thousands or even tens of thousands. The sensors are aggregated into sophisticated computational and communication infrastructures, called wireless sensor networks, whose goal is to produce globally meaningful information from data collected by individual sensors. However, the massive deployment of sensors, combined with anonymity of individual sensors, limited energy budget and, in many applications, a hostile environment, pose daunting challenges to the design of protocols for wireless sensor networks. For one thing, the limited energy budget at the individual sensor level mandates the design of ultralightweight communication protocols. Likewise, issues concerning how the data collected by individual sensors could be queried and accessed, and how concurrent sensing tasks could be executed internally, are of particular significance. An important guideline in this direction is to perform as much local data processing as possible at the sensor level, avoiding the transmission of raw data through the network. Indeed, it is known that it costs 3 J of energy to transmit 1 kb of data a distance of 100 m. Using the same amount of energy, a general-purpose processor with the modest specification of 100 million instructions/watt executes 300 million instructions [20,21].

As a consequence, the wireless sensor network must be multihop, and only a limited number of the sensors count the sink among their one-hop neighbors. For reasons of scalability, it is assumed that no sensor knows the topology of the network.

4.3.1 Interfacing Wireless Sensor Networks

We assume that the wireless sensor network is connected to the outside world (e.g., point of command and control, the Internet, etc.) through a *sink*. The sink may or may not be collocated with the sensors in the deployment area. In case of a noncollocated sink, the interface with the outside world may be achieved by a vehicle driving by the area of deployment, or a helicopter, aircraft, or low earth orbit (LEO) satellite overflying the sensor network, and collecting information from a select group of reporting nodes. In such scenarios communication between individual sensors is by radio, while the reporting nodes are communicating with the noncollocated sink by radio, infrared, or laser [8,9]. One can easily contemplate a collection of mobile sinks for fault tolerance.

When the sink is *collocated* with the wireless sensor network, it can also be in charge of performing any necessary training and maintenance operations. Throughout this chapter we shall assume that the sink is collocated with the sensors, and we shall refer to it occasionally as *training agent* (TA, for short), especially in contexts

where the sink engages in training operations. Moreover, we shall assume that the sink is centrally placed in the deployment area. This is for convenience only; it will be clear that the virtual infrastructure induced by the sink is topologically invariant to translating the sink out of its central position. A corollary of this is that our approach works equally well with eccentric sinks as well as with moving ones. We shall not elaborate this point further in this chapter.

4.3.2 Synchronization

The problem of synchronizing sensors has deep implications on the types of applications for which wireless sensor networks are a suitable platform. Not surprisingly, the synchronization problem has received a good deal of well-deserved attention in the recent literature [28,29]. To the best of our knowledge, all the synchronization strategies used are *active* in the sense that time awareness is propagated from sensor to sensor in the network. Our strategy is *passive* in the sense that the sensors synchronize to a master clock running at the sink. In addition to being simpler, our method promises to be far more accurate as we avoid the snowballing effect of errors inherent to active propagation.

Using the genetic material, each sensor can generate (pointers into) three sequences of random numbers as follows:

1. A sequence $t_1, t_2, \ldots, t_i, \ldots$ of time-epoch lengths
2. A sequence $n_1, n_2, \ldots, n_i, \ldots$ of frequency sets drawn from a huge universe, for example, the industrial, scientific, medical (ISM) band
3. For every i ($i \geq 1$), a permutation $f_1^i, f_2^i, \ldots$ of frequencies from n_i

The interpretation of these sequences is: time is ruled into epochs: during the ith time epoch, of length t_i, frequency set n_i is used, subject to the hopping sequence $f_1^i, f_2^i, \ldots$. Thus, as long as a sensor is synchronous to the TA, it knows the current time epoch, the offset into the epoch, the frequencies, and the hopping pattern for that epoch.

Suppose that the TA dwells τ microseconds on each frequency in the hopping sequence. For every i ($i \geq 1$), we let l_i stand for t_i/τ (assumed to be an integer); thus, epoch t_i involves a hopping sequence of length l_i. Think of epoch t_i as being partitioned into l_i slot, each slot using its own frequency selected by the hopping pattern from the set n_i. We refer the reader to Figure 4.1 where some of these ideas are illustrated. For example, time epoch t_{i-1} uses a set of frequencies $n_{i-1} = \{1, 3, 4, 5, 12, 13, 14, 15, 16\}$. Similarly, t_i uses the set of frequencies $n_i = \{2, 3, 6, 7, 10, 11, 12, 14\}$, while epoch t_{i+1} uses $n_{i+1} = \{4, 5, 8, 9, 13, 16\}$. The figure also illustrates the specific frequencies used in each slot.

It is clear that determining the epoch and the offset of the TA in the epoch is sufficient for synchronization. Our synchronization protocol is predicated on the assumption that sensor clock drift is bounded. Specifically, assume that whenever a sensor wakes up and its *local* clock shows epoch t_i, the master clock at the TA

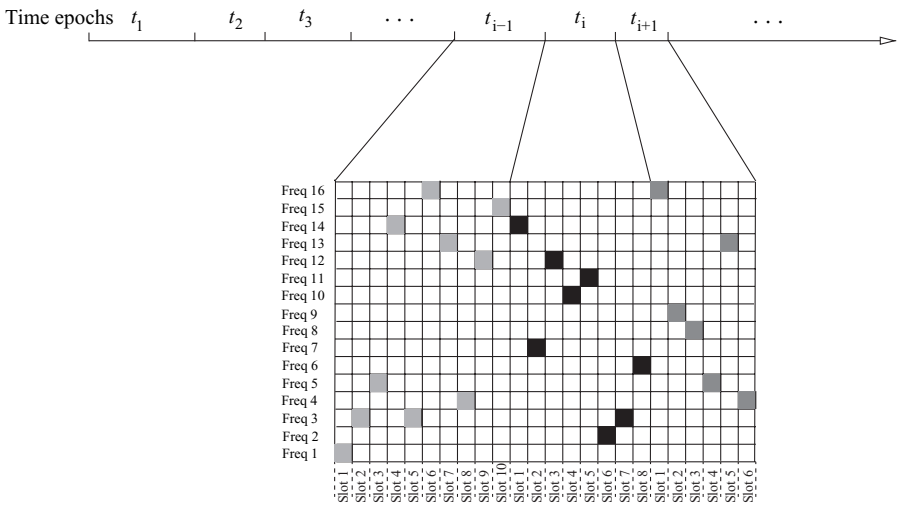

Figure 4.1 Sensor synchronization.

is in one of the time epochs t_{i-1}, t_i, or t_{i+1}. Using its genetic information, the sensor knows the last frequencies λ_{i-1}, λ_i, and λ_{i+1} on which the TA will dwell in the time epochs t_{i-1}, t_i, and t_{i+1}, respectively. Its strategy, therefore, is to tune in, cyclically, to these frequencies, spending $\tau/3$ time units on each of them. It is clear that eventually the sensor meets the TA on one of these frequencies. Assume, without loss of generality, that the sensor meets the TA on frequency λ in some (unknown) slot s of one of the epochs t_{i-1}, t_i, or t_{i+1}. To verify the synchronization, the sensor will attempt to meet the TA in slots $s+1$, $s+2$, and $s+3$ at the start of the next epoch. If a match is found, the sensor declares itself synchronized. Otherwise, the sensor will repeat the process just delineated.

It is important to understand that the synchronization protocol outlined is probabilistic: even if a sensor declares itself synchronized, there is a slight chance that it is not. However, a missynchronization will be discovered quickly and the sensor will reattempt to synchronize.

4.4 LOCATION AWARENESS IN WIRELESS SENSOR NETWORKS

Consider a circular deployment area along with a centrally placed TA equipped with a long-range radio and a steady energy supply, that can communicate with the sensors in the deployment area. Recall that, as noted before, the role of the TA is played by the collocated sink.

It was recognized that some applications require that the collected sensory data be supplemented with location information, encouraging the development of

communication protocols that are location-aware and perhaps location-dependent [7,30–33]. The practical deployment of many wireless sensor networks results in sensors initially *unaware* of their location: they must acquire this information post-deployment. Further, due to limitations in form factor, cost per unit and energy budget, individual sensors are not expected to be global positioning system (GPS)-enabled. Moreover, in many probable application environments, including those inside buildings, hangars, or warehouses, satellite access is drastically limited.

The *location awareness* problem, then, 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 closely as possible, their geographic coordinates in the area of deployment. Prominent solutions to the localization problem are based on multilateration or multiangulation [30–36]. Most of these solutions assume the existence of several *anchors* that are aware of their location (perhaps by endowing them with a GPS-like device). Sensors receiving location messages from at least three sources can approximate their own locations. For a good survey of localization protocols for wireless sensor networks, we refer the reader to ref. [37].

For the sake of completeness, we now outline a very simple localization protocol for wireless sensor networks that does not rely on multiple anchors.

4.4.1 A Simple Localization Protocol for Wireless Sensor Networks

The task of *localization* is performed immediately after deployment. If the sensors are considered stationary, localization is a one-time operation.[1] Unlike the vast majority of existing protocols that rely heavily on multilateration or multiangulation and on the existence of a minimum of three anchors with known geographic position, our protocol only requires one anchor—the TA—whose role can be played by a collocated sink. The key idea of our protocol is to allow each sensor to determine its position in a polar coordinate system centered at the TA. In particular, each sensor determines its *polar angle* with respect to a standard polar axis as well as a *polar distance* to the TA.

Referring to Figure 4.2, assume without loss of generality that the TA is centrally located.[2] The TA knows its own geographic coordinates, is not energy constrained and it has (highly) directional transmission capabilities.

For some predetermined time, the TA transmits a rotating beacon, as illustrated in Figure 4.2. The rotation is uniform with a period of T time units, known to all the sensors in the deployment area. Every time the beacon coincides with the polar axis the TA transmits a synchronization signal on a channel λ, known to the sensors.

In outline, the protocol is as follows. A generic sensor a wakes up according to its internal clock. It listens to channel λ for T time units. Let t_0 be the moment at which

[1] In fact, even if the sensors are stationary, they may move from their original deployment position due to such factors as wind, rain, and small ground tremors.

[2] The reader should have no difficulty confirming that this is assumed for convenience and the eccentric TA case is perfectly similar.

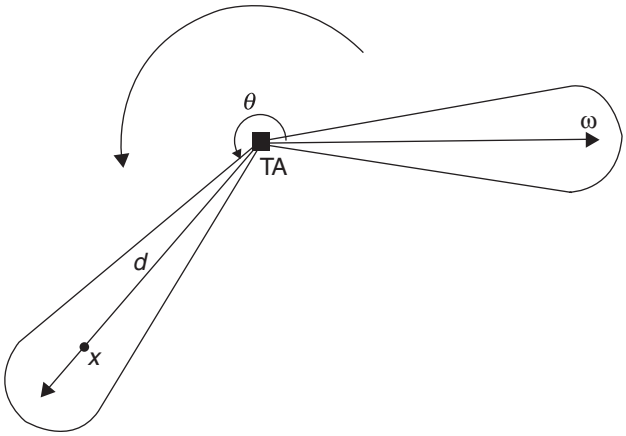

Figure 4.2 The localization protocol.

it hears the synchronization beacon. At that point it switches to channel μ, on which the rotating beacon is transmitted. Assume that the rotating beacon is received by sensor a at time t_1. The polar angle θ corresponding to a is

$$\theta = \frac{2\pi(t_1 - t_0)}{T} \tag{4.1}$$

Similarly, the polar distance ρ can be determined by using the well-known formula

$$\rho = \left(\frac{P_T}{cP_R}\right)^{1/\alpha} \tag{4.2}$$

where

P_T and P_R represent, respectively, the transmitted and received energy levels

c and α are constants that depend on the atmospheric conditions at the moment when the localization takes place. These values may be passed on by the TA, along with P_T.

It is worth noting that a sensor may perform several determinations of θ and ρ and use averages to improve the accuracy of the localization. Indeed, once t_1 is known, the sensor can go to sleep until time $t_1 + T$, at which it knows that it needs to wake up to receive the beacon again.

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. In this chapter

we show that sensors acquire coarse-grain location awareness by the *training* protocol that imposes a coordinate system onto the network. An interesting by-product of our training protocol is that it provides a 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.

4.5 THE VIRTUAL INFRASTRUCTURE

The main goal of this section is to present a broad overview of the main components of the proposed general-purpose virtual infrastructure for wireless sensor networks.

4.5.1 A Dynamic Coordinate System

To help with organizing the virtual infrastructure we assume a centrally placed TA, equipped with a long-range radio and a steady energy supply, that can communicate with both the sink and the sensors in the deployment area.

Referring to Figure 4.3(a) the coordinate system divides the wireless sensor network area into equiangular wedges. In turn, these wedges are divided into *sectors* by means of concentric circles or coronas centered at the TA (sink). As will be discussed in Subsection 4.5.2, the sensors in a given sector map to a cluster, the

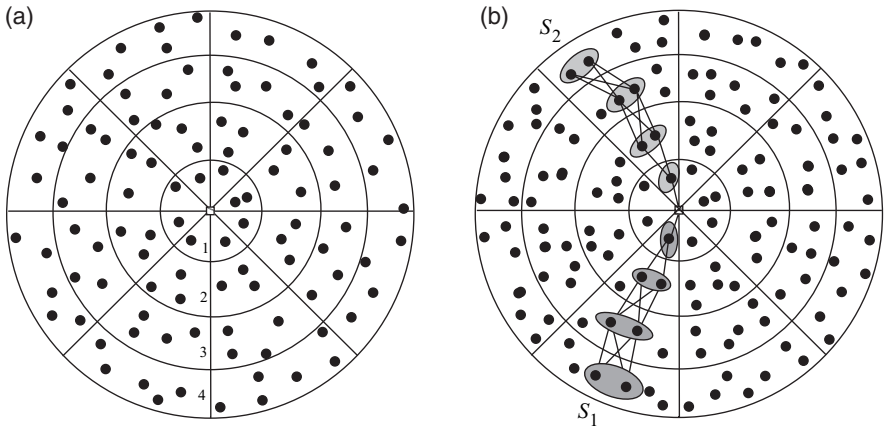

Figure 4.3 Different perspectives of the dynamic coordinate system: (a) the dynamic system, and (b) routing in a wireless sensor network.

mapping between clusters and sectors being one-to-one. The task of training a wireless sensor network involves establishing:

Coronas. The deployment area is covered by k coronas determined by k concentric circles of radii $0 < r_1 < r_2 < \cdots < r_k \leq t_x$ centered at the sink.

Wedges. The deployment area is ruled into a number of angular wedges centered at the sink.

As illustrated in Figure 4.3(a), at the end of the training period each sensor has acquired two coordinates: the identity of the corona in which it lies, as well as the identity of the wedge to which it belongs. It is important to note that the locus of all the sensors that have the same coordinates determines a cluster.

4.5.2 The Cluster Structure

Clustering was proposed in large-scale networks as a means of achieving scalability through a hierarchical approach. For example, at the medium access layer, clustering helps increase system capacity by promoting the spatial reuse of the wireless channel; at the network layer, clustering helps reducing the size of routing tables and striking a balance between reactive and proactive routing. It is intuitively clear that wireless sensor networks benefit a great deal from clustering; indeed, separating concerns about intercluster management and the intracluster management can substantially decrease and load balance the management overhead. Given the importance of clustering, a number a clustering protocols for wireless sensor networks have been proposed in the recent literature [38–40]. However, virtually all clustering protocols for wireless sensor networks assume tacitly or explicitly that individual sensors have identities.

The dynamic coordinate system suggests a simple and robust *clustering scheme*: a cluster is the locus of all sensors having the same coordinates. It is important to note that clustering is obtained for free once the coordinate system is established. Also, our clustering scheme does not assume synchronization and accommodates sensor anonymity: sensors need not know the identity of the other sensors in their cluster. For an illustration, refer again to Figure 4.3(a). Each sector in the dynamic coordinate system represents a cluster; indeed, as is easily visible, the sensors in a sector share the same coordinates: the same corona number and the same wedge number.

Recently Olariu et al. [27] showed that one can augment the virtual infrastructure with a task-based management system where clusters are tasks with sensing, routing, or collective data storage.

4.6 THE LIGHTWEIGHT TRAINING PROTOCOL

The model for a wireless sensor network that we adopt assumes that after deployment the sensors must be trained before they can be operational. Recall that sensors

do not have identities and are initially unaware of their location. It follows that untrained nodes are not addressable and cannot be targeted to do work in the network. The main goal of this section is to present, in full detail, our lightweight, highly scalable training protocol for wireless sensor networks. The key advantage of this protocol is that each sensor participating in the training incurs an energy cost that is logarithmic in the number of clusters and wedges defined by the protocol. Being energy-efficient, this training can be repeated on a scheduled or ad hoc basis, providing robustness and dynamic reorganization.

After deployment the individual sensors sleep until wakened by their individual timers. Thus, each sensor sleeps for a random period of time, wakes up briefly, and if it hears no messages of interest, selects a random number x and returns to sleep x time units. Clocks are not synchronized, but over any time interval $[t, t + \Delta t]$ a percentage directly proportional to Δt of the nodes are expected to wake up briefly. During this time interval the sink continuously repeats a call to training, specifying the current time and a rendezvous time. Thus, in a probabilistic sense a certain percentage of the sensor population will be selected for training. The time interval Δt can be adjusted to control the percentage of sensors that is selected. Using the synchronization protocol described in Subsection 4.3.2 the selected sensors reset their clocks and set their timer appropriately before returning to sleep.

4.6.1 The Corona Training Protocol

The main goal of this subsection is to present the details of the corona training protocol. The wedge training protocol being quite straightforward will not be discussed further in this chapter.

Let k be an integer[3] known to the sensors and let the k coronas be determined by concentric circles of radii $0 < r_1 < r_2 < \cdots < r_k \leq t_x$ centered at the sink.

The idea of the corona training protocol is for each individual sensor to learn the identity of the corona to which it belongs. For this purpose, each sensor learns a string of $\log k$ bits, from which the corona number can be determined easily. To see how this is done, it is useful to assume time ruled into slots $s_1, s_2, \ldots, s_{k-1}$ and that the sensors synchronize to the master clock running at the sink, as discussed in Subsection 4.3.2.

In time slot s_1 all the sensors are awake and the sink uses a transmission range of $r_{k/2}$. As a net effect, in the first slot the sensors in the first $k/2$ coronas will receive the message above a certain threshold, while the others will not. Accordingly, the sensors that receive the signal set $b_1 = 0$, the others set $b_1 = 1$.

Consider a k-leaf binary tree T and refer to Figure 4.4. In the figure the leaves are represented by *boxes* numbered left to right from 1 to k. It is very important to note that the intention here is for the k boxes to represent, in left-to-right order, the k coronas. The training protocols is for individual sensors to determine the "box" (i.e., the corona) to which they belong.

[3]For simplicity, we shall assume that k is a power of 2.

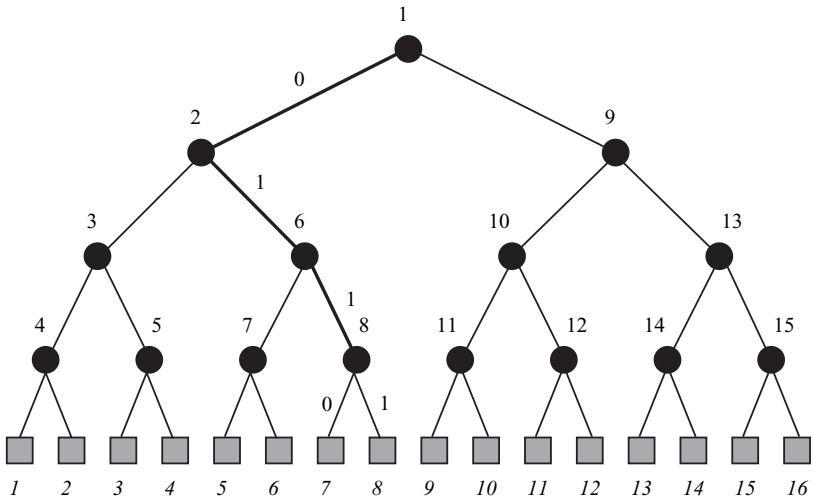

Figure 4.4 Corona training.

The edges of T are labeled by 0s and 1s in such a way that an edge leading to a left subtree is labeled by a 0 and an edge leading to a right subtree is labeled by a 1. Let l $(1 \leq l \leq k)$ be an arbitrary leaf, and let $b_1, b_2, \ldots, b_{\log k}$ be the edge labels of the unique path leading from the root to l. It is both well known and easy to prove by a standard inductive argument that

$$l = 1 + \sum_{j=1}^{\log k} b_j \frac{k}{2^j} \tag{4.3}$$

As an illustration, applying equation (4.3) to leaf 7, we have $7 = 1 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$.

Referring again to Figure 4.4, let the *interior* nodes of T be numbered in *preorder* from 1 to $k - 1$, and let T' be the tree consisting of the interior nodes only.[4] Let u be an arbitrary node in T', and let $b_1, b_2, \ldots, b_{i-1}$ be the edge labels on the unique path from the root to u. We take note of the following technical result.

Lemma 4.1: Let $p(u)$ be the preorder number of u in T'. Then, we have

$$p(u) = 1 + \sum_{j=1}^{i-1} c_j$$

[4]In other words, T' is the tree obtained from T by ignoring the last level (i.e., the "boxes").

where

$$c_j = \begin{cases} 1 & \text{if } b_j = 0 \\ \dfrac{k}{2^j} & \text{if } b_j = 1 \end{cases}$$

Proof: The proof is by induction on the depth i of node u in T'. To settle the basis, note that for $i = 1$, u must be the root and $p(u) = 1$, as expected.

For the inductive step, assume the statement true for all nodes in T' of depth less that u. Indeed, let v be the parent of u and consider the unique path of length $i - 1$ joining the root to u. Clearly, nodes u and v share $b_1, b_2, \ldots, b_{i-2}$ and, thus, $c_1, c_2, \ldots, c_{i-2}$. By the inductive hypothesis,

$$p(v) = 1 + \sum_{j=1}^{i-2} c_j \tag{4.4}$$

On the other hand, since v is the parent of u, we can write

$$p(u) = p(v) + \begin{cases} 1 & \text{if } u \text{ is the left child of } v \\ \dfrac{k}{2^{i-1}} & \text{otherwise} \end{cases} \tag{4.5}$$

Notice that if u is the left child of v we have $b_{i-1} = 0$ and $c_{i-1} = 1$; otherwise, $b_{i-1} = 1$ and $c_{i-1} = k/2^{i-1}$. This observation, along with equations (4.4) and (4.5) combined, allows us to write

$$p(u) = 1 + \sum_{j=1}^{i-2} c_j + c_{i-1} = 1 + \sum_{j=1}^{i-1} c_j$$

completing the proof of the lemma. ■

Let u be an arbitrary node of T' and let $n(u)$ denote its inorder number in T'. Let m be the left-to-right rank among the leaves of T of the rightmost leaf of the left subtree of T rooted at u.

Lemma 4.2: $n(u) = m$.

Proof: We proceed by induction on the inorder number of a node in T'. Indeed, if $n(u) = 1$, then u must be the leftmost leaf in T' and, thus, its left subtree in T consists of the leftmost leaf of T', settling the base case.

Assume that the statement is true for all nodes of T' with inorder number smaller than that of u. we shall distinguish between the following two cases:

Case 1: v is an ancestor of u in T'. Let $T'(v)$ be the subtree of T' rooted at v. In this case, u must be the leftmost leaf in the right subtree of $T'(v)$. Let q be the left-to-right

rank among the leaves of T of the rightmost leaf of the left subtree of $T'(v)$. By the inductive hypothesis, $n(v) = q$. Since u is a leaf in T', it has exactly two children in T, namely, the leaves of ranks $q+1$ and $q+2$. Thus, in this case, $n(u) = n(v) + 1 = q + 1$, as claimed.

Case 2: u is an ancestor of v in T'. Let $T'(u)$ be the subtree of T' rooted at u. In this case, v must be the rightmost leaf in the left subtree of $T'(u)$. Assume that $n(v) = r$. Observe that v has exactly two leaf children T. By the induction hypothesis, these children have ranks r and $r+1$. Thus, in this case, $n(u) = n(v) + 1 = r + 1$, as claimed.

This completes the proof of the lemma. ■

To illustrate Lemma 4.2, refer again to Figure 4.4 and let u be the internal node labeled "6." Recall that the tree T' consists of the tree T with the level removed. It is easy to verify that "6" is, in fact, the inorder number of u in T'. By Lemma 4.2 this coincides with the label of the box that is the leftmost leaf in the right subtree of $T'(v)$ rooted at u.

With these technicalities out of the way, we now return to the corona training protocol. In our setting, the preorder and inorder numbers of internal nodes in T correspond, respectively, to time slots in the training protocol and to the transmission ranges used by the sink. More precisely, consider an arbitrary integer i, $(2 \leq i \leq \log k - 1)$, and assume that at the end of time slot s a sensor has learned the leftmost $i-1$ bits $b_1, b_2, \ldots, b_{i-1}$. The following important result is implied by Lemma 4.1 and Lemma 4.2.

Theorem 4.1: Having learned bits $b_1, b_2, \ldots, b_{i-1}$, a sensor must wake up in time slot $z = 1 + \sum_{j=1}^{i-1} c_j$ to learn bit b_i. Moreover in time slot z the sink uses a transmission range of $r_{inorder(z)}$.

To illustrate Theorem 4.1, refer again to Figure 4.4 where the internal nodes are labeled by their preorder numbers. Consider the node labeled 2. It is easy to verify that its inorder number is 4. Thus, all the nodes in the subtree rooted at 2 will be awake in slot 2 and the sink will transmit with a transmission range of r_4. Consequently, the sensors at a distance from the sink not exceeding r_4 will receive the signal, while the others will not.

It is also worth noting that only the sensors that need to be awake in a given time slot will stay awake; the others will sleep, minimizing the energy expenditure. Yet another interesting feature of the training protocol we just described is that individual sensors sleep for as many contiguous slots as possible before waking up, thus avoiding repeated wake–sleep transitions that are likely to waste energy.

At the same time, in case the corona training process has to be aborted before it is complete, Theorem 4.1 guarantees that if the training process restarts at some later point, every sensor knows the exact time slots when it has to wake up in order to learn its missing bits.

Making the training protocol secure is especially important, since training is a prerequisite for subsequent network operations. Recently, Jones et al. [26] and Wadaa et al. [41,42] have shown that the training protocol described earlier can be made secure.

4.7 TASK-BASED DATA PROCESSING AND COMMUNICATION

The goal of this section is to describe a task-based data-processing and communication system for wireless sensor networks that exploits the virtual infrastructure introduced in this chapter. For this purpose, we shall adopt the view that the wireless sensor network performs tasks mandated by a remote end user. The end user issues queries expressed in terms of high-level abstractions, to be answered by the network. The middleware, running at the sink, provides the interface between the application layer (where the end user resides) and the wireless sensor network. Specifically, the sink parses the queries from the application layer, considers the current capabilities of the network including the remaining energy budget and negotiates a contract with the application layer before committing the network [42]. After a contract has been agreed upon, the middleware translates the corresponding query into low-level tasks, assigned to individual clusters. The clusters must then perform these tasks and send the aggregated data back to the sink for consolidation. The consolidated information is then passed on to the application layer.

4.7.1 Associating Sensors with Tasks

For our purposes a task is a tuple $T(A, S, E)$, where

- A describes the action to be performed (i.e., detecting physical intrusion into the deployment area).
- S specifies the identity of the cluster tasked with data collection (sensing).
- E specifies the minimum energy level required of sensors participating in the task.

The suitably aggregated data collected by the sensors is to be routed to the sink before being uploaded to the end user. In addition to the sensors in cluster S, a number of sensors are selected to act as *routers*, relaying the data collected to the sink. Collectively, these sensors are the *workforce* $W(T)$ associated with T.

The process by which $W(T)$ is selected follows. During a time interval of length Δ the sink issues a *call for work* containing the parameters of T. The sensors in the same wedge as S and with corona numbers smaller than that of S that happen to be awake during the interval Δ and that satisfy the conditions specified (membership in S and energy level) stay awake and constitute $W(T)$. It is intuitively clear that by knowing the number of sensors, the density of deployment and the expected value of sleep periods, one can fine-tune Δ in such a way that $W(T)$ is commensurate with the

desired grade of service. It is extremely important to note that, as discussed in Subsection 4.3.2, a by-product of the call for work is that all the sensors in $W(T)$ are synchronized for the duration of the task.

For an illustration of the concepts discussed in this subsection, we refer to Figure 4.3(b). In the figure two tasks are in progress. One of these tasks has mandated sensors in cluster S_1 to collect data in support of a query. The sensors associated with this task as routers are those in the outlined sets in the same wedge as S_1. Since the width of each corona does not exceed the maximum transmission range t_x, communication between sensors in adjacent coronas is assumed. Also note that the sensors that constitute the workforce of this transaction are synchronized. As for the transmission of data, all the sensors in the same sector transmit at the same time. As will be discussed in detail in Subsection 4.8.2, one of the benefits of our scheme is that data aggregation can be accomplished in a straightforward manner.

The figure features a second task that involves data collection in a cluster S_2 along with its workforce. As will be discussed in the next subsection, there is no collision between the two tasks, as they use a different set of frequencies.

4.7.2 Task-Based Synchronization

The generic synchronization protocol discussed earlier in this chapter can be used as a building block for a more sophisticated task-based synchronization protocol. The motivation is to support multitasking. Indeed, it is often desirable for the sensors in a cluster to perform several tasks in parallel.[5] However, any attempt at synchronization using the generic synchronization protocol will result in all the concurrent tasks using *exactly* the same frequency set and the same hopping sequence, creating frequent collisions and the need for subsequent retransmission.

Suppose that we wish to synchronize the workforce $W(T)$ of a task T that uses some color class c and that the generic synchronization protocol would show that the actual time epoch is t_i. The idea is to use the perfect hash function ϕ to compute a *virtual* time epoch t_j with $j = \phi(i, k(c), T)$ to be used by $W(T)$. Therefore, the sensors in $W(T)$ will act as if the real time were t_j, using the frequency set n_j and the frequency hopping sequence $f_1^j, f_2^j, \ldots$. Thus, different concurrent tasks will employ different frequency sets and hopping sequences minimizing the occurrence of collisions.

4.8 ROUTING AND DATA AGGREGATION

The main goal of this section is to show that once a wireless sensor network has been trained, both routing and data aggregation become easy and straightforward.

[5]However, the sets of sensors allocated to these tasks *must* be disjoint.

4.8.1 Routing

The routing problem in sensor networks differs rather substantially from routing in other types of wireless networks. For one thing, individual sensors are anonymous, lacking identities; thus, standard addressing methods do not work directly. For another reason, the stringent energy limitations present in the sensor network render the vast majority of conventional routing protocols impractical.

Given the importance of routing, it is not surprising to see that a number of routing protocols specifically designed for wireless sensor networks were proposed in the literature [15,43–46]. For example, in ref. [15] Intanagonwiwat et al. describe *directed diffusion* and a companion routing protocol based on interest tables at the expense of maintaining a cache of information indexed by interest area at each node. Shah and Rabaey [46] responds to client requests by selecting paths that maximize the longevity of the network rather than minimize total energy consumed by a path with path options established by local flooding. Other routing protocols include rumor routing [43], and multipath routing [44], among others. As we are about to demonstrate, our training protocol provides a novel solution to the routing problem by yielding energy-efficient paths-based routing.

Recall that sensor networks are multihop. Thus, in order for the sensing information to be conveyed to the sink, routing is necessary. Our cluster structure allows a very simple routing process, as described in the following paragraphs. The idea is that the information is routed within its own wedge along a virtual path joining the outermost sector to the sink, as illustrated in Figure 4.3(b). The collection of all the virtual paths (one per wedge) defines a tree. In this tree, each internal node, except for the root, has exactly one child, eliminating medium access control (MAC)–level contention in sending sensor information to the sink.

Recently, a number of MAC-layer protocols for wireless sensor networks have been proposed in the literature [47–49]. In fact, in our routing scheme by appropriately staggering transmissions in neighboring wedges, collision and, therefore, the need for retransmissions is completely eliminated. Thus, our training protocol implies an efficient MAC protocol as well.

4.8.2 Data Aggregation

Once sensory data is collected by a multitude of sensors, the next important task is to consolidate the data in order to minimize the amount of traffic to the sink. We place the presentation in the context of our work model. To be more specific, we assume that the cluster identified by (i, j)—that is, the set of sensors located in sector $A_{i,j}$—are tasked to perform a certain task $\mathcal{T}$. A number of sensors in sectors $A_{i,1}, A_{i,2}, \ldots, A_{i-1,j}$ are selected to act as *routers* of the data collected by the sensors in $A_{i,j}$ to the sink. Collectively, these sensors are the *support* sensors of task $\mathcal{T}$.

It is, perhaps, of interest to describe the process by which the sensors associated with $\mathcal{T}$ are selected. To begin, during a time interval of length Δ the sink will issue a *call for work* specifying the identity j of the wedge in which the task is to be performed, as well as the identity i of the corona in which data are to be collected.

The sensors in wedge j that happen to wake up during the interval Δ and that have an appropriate energy level stay awake and will participate in the task either as data collectors or as routers, depending on their respective position within the wedge. It is intuitively clear that by knowing the number of sensors, the density of deployment and the expected value of sleep periods, one can fine-tune Δ in such a way that a suitable number of routers will be awake in wedge j in support of $\mathcal{T}$. Likewise, we can select the set $\mathcal{D}$ of data collecting sensors in $A_{i,j}$. Let $\mathcal{S}$ denote the set of support sensors for $\mathcal{T}$. It is appropriate to recall that a by-product of the call for work is that all the sensors in $\mathcal{S}$ are synchronized. In order to make the task secure the sensors in $\mathcal{S}$ will share a secret key that allows them access to a set of time epochs, a set of frequencies to be used in each time epoch, and a hopping sequence to be used within each epoch. For details, we refer the reader to Section 4.2.

Assume that the results of the data collection specific to task $\mathcal{T}$ can be partitioned into 2^m, ($m \geq 0$), disjoint groups. Thus, each sensor performing data collection will encode its data in a string of m bits.

Since, typically, $\mathcal{D}$ contains a large number of sensors, it is important to *fuse* individual results into a final result that will be sent to the sink. We now outline a possible solution to the data-aggregation problem. Using the algorithms of Nakano and Olariu [50,51] which do not require sensors to have identities, the sensors in $\mathcal{D}$ acquire *temporary* identities ranging from 1 to $|\mathcal{D}|$. Using their newly acquired identities, individual data values are being transmitted to the sensor whose identity is 1, which will perform data aggregation and will send the final result to the sink. The advantage of this data-aggregation scheme is that there is no data loss and all the collected values will be correctly fused. There are, however, many disadvantages. For one thing, the initialization algorithm of [50] requires every sensor in $\mathcal{D}$ to expend an amount of energy proportional with $\log |\mathcal{D}|$. For another, the final result of the data collection is concentrated in a single sensor (i.e., the sensor with temporary identity 1), which is a single point of failure.

We now propose a much simpler data-aggregation scheme that involves some data loss, but that is fault tolerant and does not require the sensors in $\mathcal{D}$ to have unique identities. The idea is that the sensors in $\mathcal{D}$ transmit the data collected bit by bit, starting, say, left to right, as follows: a value of 0 is not transmitted, while a 1 will be transmitted. The sensors in $A_{i-1,j}$ that have been elected as routers in support of task $\mathcal{T}$ pick up the values transmitted. The following disambiguation scheme is used:

- No bit is received—in this case, a 0 is recorded.
- A bit of 1 is received—in this case, a 1 is recorded.
- A collision is recorded—in this case a 1 is recorded.

It is clear that as a result of this disambiguation scheme, every sensor in $A_{i-1,j}$ that is in support of $\mathcal{T}$ stores the logical OR of the values stored by sensors in $\mathcal{D}$. Note also that while there was loss of information in the process of fusing data, no further loss can occur in traversing the path from $A_{i-1,j}$ to the sink: this is because all routers in $A_{i-1,j}$ transmit the same bit string.

4.8.3 An Example

For an example of data aggregation consider a wireless sensor network that is tasked to monitor and report the temperature in cluster $A_{i,j}$. Referring to Table 4.1, for the application at hand temperatures below 111°F are considered to be noncritical, and if such a temperature is reported, no specific action is to be taken. By contrast, temperatures above 111°F are considered to be critical, and they trigger a further monitoring action. The encoding featured in Table 4.1 is specifically designed to reflects the relative importance of various temperature ranges. For example, the temperature ranges in the noncritical zone are twice as large as those in the critical zone. Also, notice that the leftmost bit differentiates critical from noncritical temperatures. Thus, if the sink receives a reported temperature whose leftmost bit is a 1, then further action is initiated; if, on the other hand, the leftmost bit is 0, then no special action is necessary.

Let us see how our data aggregation works in this context. Referring to Figure 4.5, assume that a group of three sensors in $A_{i,j}$ have collected data and are about to transmit them to the sensors in $A_{i-1,j}$. The values collected are encoded, respectively, as 0110, 0101, and 0110. Thus, none of the values indicates a critical situation. After transmission and disambiguation, the sensors in $A_{i-1,j}$ will store 0111, which is the logical OR of the values transmitted. Notice that although the data-aggregation process involves loss of information, we do not loose critical information. This is because the logical OR of noncritical temperatures must remain noncritical. Conversely, if the logical OR indicates a critical temperature, one of the fused temperatures must have been critical, and thus action must be initiated. It is also interest-

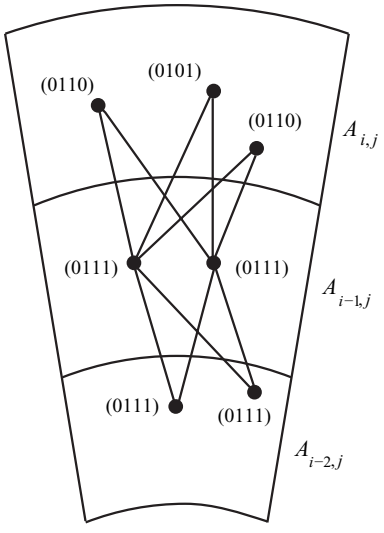

Figure 4.5 Data aggregation.

TABLE 4.1 Temperature Ranges and Their Encoding

Temperature	51–60	61–70	71–80	81–90	91–100	101–110	111–115	116–120	121–125	126–130	131–135	136–140	141–145	146–150
Code	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

ing to note that when the sensors in $A_{i-1,j}$ transmit to those in $A_{i-2,j}$, no further loss of information occurs.

4.8.4 Lossless Aggregation

It is worth noting that there is an interesting interplay between the amount of loss in data aggregation and the amount of energy expended to effect it. As we are about to show, if we are willing to expend slightly more energy, lossless data aggregation can be achieved.

The corresponding trade-off is interesting in its own right, being characteristic of choices that present themselves in the design of protocols for wireless sensor networks. For illustration purposes, assume that it is necessary to determine the maximum of the bit codes stored by the sensors in $A_{i,j}$ and refer to Figure 4.6.

To solve this problem, all the sensors in $A_{i,j}$ that have collected relevant information engage in the following protocol, which is guaranteed to aggregate the values into the maximum. Assume that each sensor stores a d-bit code for the range.

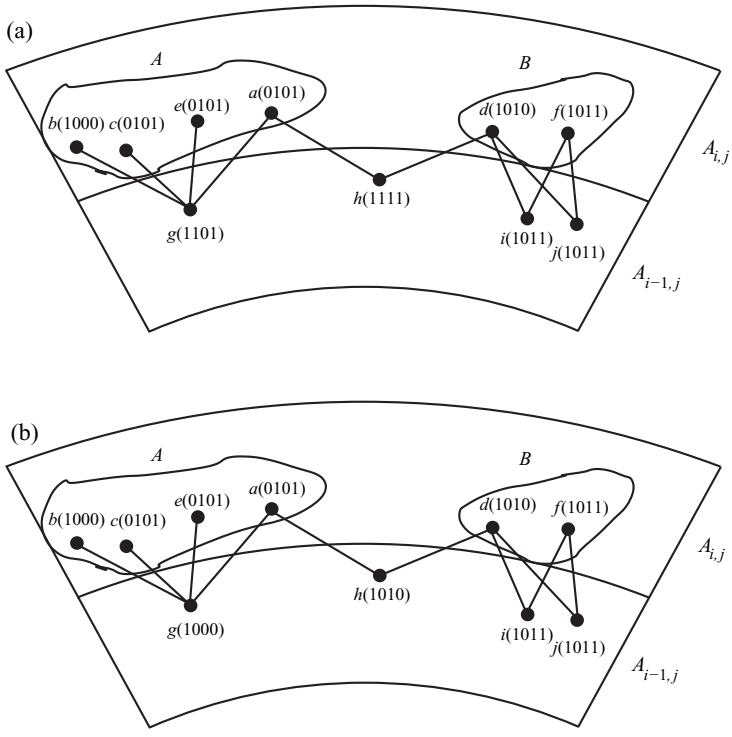

Figure 4.6 Lossless data aggregation.

Protocol (Correct_Maximum): For every position p starting with the most significant bit to the least:

1. Sensors in $A_{i,j}$ that have a 0 in position p listen for two time slots; if in any of these slots a 1 or a collision message is received, they terminate their participation in the protocol.
2. Sensors that have a 1 in position p transmit in the first time slot and sleep in the second.
3. Sensors in $A_{i-1,j}$ do the following:
   3.1. Any sensor that has received a 1 or a collision in the first time slot, echoes a 1 in the second.
   3.2. Any sensor that has not received a transmission in the first slot sleeps in the second slot.

To see why the two time slots for transmitting a single bit are necessary consider the situation depicted in Figure 4.6(a) and the following simple "algorithm":

Protocol (Incorrect_Maximum): For every position p starting with the most significant bit to the least:

1. Sensors in $A_{i,j}$ that have a 0 in position p listen; if a 1 or a collision message is received, they terminate their participation in the protocol.
2. Sensors that have a 1 in position p transmit.

Figure 4.6(a) depicts the case where, due to energy depletion the sensors that participate in the protocol are sparsely deployed. Implicit in the protocol Incorrect_Maximum is that every sensor can hear the transmission of every other sensor. In particular, notice that in group A sensor a does not hear the transmission of sensor b and continues transmitting even though it should not. Indeed, for this reason, the value received by sensor g in $A_{i-1,j}$ is not the correct maximum of values stored by the sensors in group A. A similar situation occurs when sensor h in $A_{i-1,j}$ heard the transmission of sensors a in group A and d in group B. Clearly h stores a value that corresponds to no maximum.

Notice how protocol Correct_Maximum is sidestepping this difficulty. The transmission of a single bit is separated into two time slots: first, all the sensors in $A_{i,j}$ transmit their corresponding bit. In the second slot, the sensors in $A_{i-1,j}$ echo back the values received. Since the sensor in $A_{i,j}$ that store a 0 listen for two time slots, they will realize that some sensor in $A_{i,j}$ has a 1 in that bit position and, consequently, they should drop out. The result is illustrated in Figure 4.6(b).

4.9 EVALUATING ROUTING-RELATED ENERGY EXPENDITURE

The main goal of this section is to explore the problem of energy expenditure related to routing data in a wireless sensor network. Indeed, we adopt a task-based model

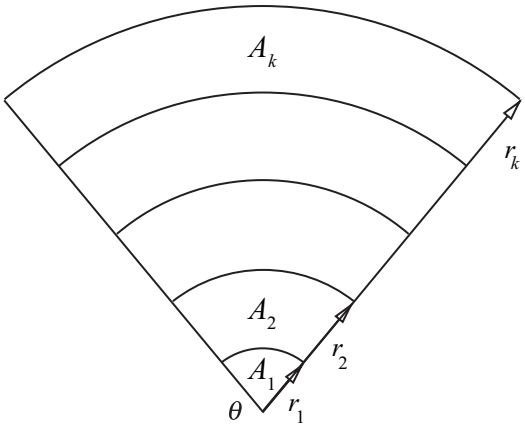

Figure 4.7 A wedge W and the associated sectors.

[27,41,42] whereby the sensor network is subjected to a set T of tasks. Each task involves the nodes in a sector (i.e., a cluster) and involves performing local sensing by the sensors, data aggregation, and sending the resulting information to the sink. Recall that, as discussed in Section 4.8, one of the key benefits of our training is that transmitting the result of the task from a sector to the sink amounts to routing the information along a path lying within the same wedge (see also Fig. 4.3(b)). Thus, we associate each task with such a path. We will now analyze the energy expended by sensors to fulfill their path-related duties.

Throughout the remainder of this chapter we assume a sensor network deployed in a circular area and a collocated sink placed at its center. Consider a wedge W subtended by an angle of θ and refer to Figure 4.7. The wedge W is partitioned into k sectors $A_1, A_2, \ldots, A_k$ by its intersection with k concentric circles, centered at the sink, and of monotonically *increasing* radii $r_1 < r_2 < \cdots < r_k$. It is important to note that r_k, the deployment radius, is a system parameter, and thus a constant for a particular sensor network.

For convenience of notation we write $r_0 = 0$ and interpret A_0 as the sink itself. Let t_x denote the *maximum* transmission range of a sensor.[6]

Let n denote the total number of sensors deployed in wedge W. We assume a uniform deployment with density ρ. In particular, with A standing for the area of wedge W, we can write

$$n = \rho A = \frac{\rho\theta}{2} r_k^2 \tag{4.6}$$

Let $n_1, n_2, n_3, \ldots, n_k$ stand for the number of nodes deployed in the sectors $A_1, A_2, A_3, \ldots, A_k$, respectively. Since the deployment is uniform, it is easy to

[6]Of course, t_x is a system parameter that depends on the particular type of sensors deployed.

confirm that for every i ($1 \leq i \leq k$),

$$n_i = \rho A_1 = \frac{\rho\theta}{2}(r_i^2 - r_{i-1}^2). \tag{4.7}$$

Let N denote the number of sector-to-sink paths (henceforth, simply denoted by *paths*) that the wedge W sees during the lifetime of the sensor network. By our previous discussion there is a one-to-one map between paths and tasks. Thus, N equals the total number T of tasks that the wedge can handle during the lifetime of the network.

We make the following assumptions motivated by the uniformity of the deployment:

- Each sensor in W is equally likely to be the source of a path to the sink
- For $2 \leq i \leq k$, each sensor in sector A_{i-1} is equally likely to serve as the next hop for a path that involves a node in A_i.

By virtue of the first assumption, the expected number of paths originating at a node in W is

$$\frac{N}{n} \tag{4.8}$$

Consider sector A_1. Since the N paths have the sink as their destination, the nodes in sector A_1 must collectively participate in all the N paths. Since A_1 contains n_1 nodes, the expected number of transmissions per node is N/n_1. Assuming a power-degradation factor of α, $2 \leq \alpha \leq 6$, the energy expended by a node in A_1 per path served is $r_1^\alpha + c$ for some *nonnegative* constant c. Thus, the total energy E_1 consumed by a node in A_1 to fulfill its routing duties is

$$E_1 = \frac{N}{n_1}\left[r_1^\alpha + c\right]$$

which, by equation (4.7), can be written as

$$E_1 = \frac{N}{n_1}\left[r_1^\alpha + c\right] = \frac{2N}{\rho\theta r_1^2}\left[r_1^\alpha + c\right] = \frac{2N}{\rho\theta}\left[r_1^{\alpha-2} + \frac{c}{r_1^2}\right] \tag{4.9}$$

It is very important to note that equation (4.9) allows us to determine the optimal value r_1^{opt} of r_1 that minimizes the value of E_1. For later reference, we note that this value is

$$r_1^{opt} = \begin{cases} t_x & \text{if } \alpha = 2 \\ \min\left\{\left(\dfrac{2c}{\alpha-2}\right)^{1/\alpha},\ t_x\right\} & \text{if } 2 < \alpha \leq 6 \end{cases} \tag{4.10}$$

Let $\overline{T}$ denote the total number of tasks performed by the entire wireless sensor network (not just wedge W) during its lifetime, and let $\overline{N}$ be the corresponding number of node-to-sink paths. Assuming that the $\overline{T}$ tasks are uniformly distributed throughout the sensor network, we can write

$$\frac{\overline{N}}{2\pi} = \frac{N}{\theta} \tag{4.11}$$

By equations (4.9) and (4.11) combined, the total energy needed by a node in A_1 to handle its routing duties is

$$E_1 = \frac{2N}{\rho\theta}\left[r_1^{\alpha-2} + \frac{c}{r_1^2}\right] = \frac{\overline{N}}{\rho\pi}\left[r_1^{\alpha-2} + \frac{c}{r_1^2}\right] \tag{4.12}$$

Let E denote the total energy budget of a sensor. Since the sensors in A_1 must have sufficient energy to handle their routing duties, by using equation (4.12) we can write

$$\frac{\overline{N}}{\rho\pi}\left[r_1^{\alpha-2} + \frac{c}{r_1^2}\right] < E$$

Recalling that in our working model there is a one-to-one correspondence between tasks and sector-to-sink paths, this inequality can be written in its equivalent form

$$\frac{\overline{T}}{\rho\pi}\left[r_1^{\alpha-2} + \frac{c}{r_1^2}\right] < E \tag{4.13}$$

4.9.1 Reasoning About the System Parameters

Inequality equation (4.13) can be interpreted in several ways, each expressing a different view of the limiting factors inherent to the sensors deployed. The goal of this subsection is to look at some of possible interpretations of inequality (4.13).

1. *Network Longevity*: We interpret $\overline{T}$, the number of transactions that the system can sustain during its lifetime as the *network longevity*. Thus, inequality (4.13) allows us to write

$$\overline{T} < \frac{\rho\pi E r_1^2}{r_1^\alpha + c} \tag{4.14}$$

which tells us that the longevity of the system is upper bounded by the ratio (4.14). More specifically, the longevity is directly proportional to the deployment density and to the reciprocal of $r_1^\alpha + c$. Consequently, if we wish to design a wireless sensor network that must sustain a given number $\overline{T}$ of

transactions, we must select the deployment density as well as the radius of the first corona accordingly. We also need to chose sensors packing an amount of energy compatible with ratio (4.14).

2. *Maximum Transmission Range Close to the Sink*: First, assuming a *known* deployment density[7] ρ, inequality (4.13) shows that for a given energy budget E, in order to guarantee a desired network longevity of $\overline{T}$ tasks, the (maximum) transmission radius of sensors deployed in close proximity to the sink must satisfy

$$r_1^{\alpha-2} + \frac{c}{r_1^2} < \frac{\pi\rho E}{\overline{T}} \tag{4.15}$$

with the additional constraint that $r_1 \leq t_x$ where, recall, t_x stands for the *maximum* transmission range of a sensor.

3. *Deployment Density*: Likewise, for a selected radius r_1 ($t_x \geq r_1$), and for a given energy budget E, in order to guarantee a network *longevity* of $\overline{T}$ tasks, the deployment density ρ must satisfy the inequality

$$\rho > \frac{\overline{T}\left[r_1^{\alpha} + c\right]}{E\pi r_1^2} \tag{4.16}$$

This latter inequality can also be used (perhaps in conjunction with (14) to plan future re-deployments as the existing sensors exhaust their energy budget.

4.9.2 Energy Expenditure

In this subsection we turn to the task of evaluating the energy expenditure per node in an arbitrary sector A_i with $i \geq 1$. Since the case $i = 1$ was handled in the previous section, we now assume $i \geq 2$.

Observe that nodes in a generic sector A_i ($2 \leq i \leq k$) are called on to serve two kinds of paths:

1. Paths originating in a sector A_j with $i < j \leq k$
2. Paths originating at a node in A_i

It is easy to confirm that the number of paths involving nodes in A_i includes all paths except those originating in one of the sectors $A_1, A_2, \ldots, A_{i-1}$. Therefore, the total number of paths that the nodes in A_i must handle is

$$N - \frac{N}{n}(n_1 + n_2 + \cdots + n_{i-1})$$

[7]It is important to note that given the deployment area, the density can be engineered beforehand by simply deploying a suitable number of sensors uniformly at random.

By equations (4.6) and (4.7) combined with elementary manipulations, this expression can be written as

$$N\left[1-\frac{\sum_{i=1}^{k}(r_i^2-r_{i-1}^2)}{r_k^2}\right]=N\left[1-\frac{r_{i-1}^2}{r_k^2}\right] \tag{4.17}$$

Recall that sector A_i contains n_i nodes. This implies that each node in A_i must participate in

$$\frac{N}{n_i}\left[1-\frac{r_{i-1}^2}{r_k^2}\right]$$

paths. Using equation (4.7), the number of paths handled by each node in A_i can be written as

$$\frac{2N}{\rho\theta}\left[1-\frac{r_{i-1}^2}{r_k^2}\right]\frac{1}{r_i^2-r_{i-1}^2} \tag{4.18}$$

Observe that the width of sector A_i is $r_i - r_{i-1}$. It follows that the transmission range needed to send information between A_i and A_{i-1} is $r_i - r_{i-1}$. We shall adopt a most general power-degradation model according to which the energy expended by a node in A_i to send information to sensors in A_{i-1} is

$$(r_i-r_{i-1})^{\alpha}+c$$

where c is a nonnegative constant.

Let the total amount of energy expended by a node in A_i be E_i. By equations (4.11) and (4.18), we have

$$E_i=\frac{\overline{N}}{\pi\rho}\left[1-\frac{r_{i-1}^2}{r_k^2}\right]\frac{1}{r_i^2-r_{i-1}^2}[(r_i-r_{i-1})^{\alpha}+c]$$

Simple manipulations show that

$$E_i=\frac{\overline{N}}{\pi\rho}\left[1-\frac{r_{i-1}^2}{r_k^2}\right]\left[\frac{(r_i-r_{i-1})^{\alpha-1}}{r_i+r_{i-1}}+\frac{c}{r_i^2-r_{i-1}^2}\right] \tag{4.19}$$

For later reference we will find it convenient to write

$$E_i=E_i'+E_i''$$

where

$$E_i' = \frac{\overline{N}}{\pi\rho}\left[1 - \frac{r_{i-1}^2}{r_k^2}\right]\frac{(r_i - r_{i-1})^{\alpha-1}}{r_i + r_{i-1}} \tag{4.20}$$

and

$$E_i'' = \frac{\overline{N}}{\pi\rho}\left[1 - \frac{r_{i-1}^2}{r_k^2}\right]\frac{c}{r_i^2 - r_{i-1}^2} \tag{4.21}$$

We also assume that for all $i,\ 1 \le i \le k,$ every sensor in sector A_i should be within transmission range from *some* sensor in sector A_{i-1}. In particular, every sensor in sector A_1 must be within transmission range from the sink.[8]

4.9.3 Optimizing the Size of Coronas

The main goal of this section is to show how to select the radii $r_1, r_2, \ldots, r_k$ in such a way that *total* energy spent per sector-to-sink routing path is minimized. For this purpose, let ε_i denote the total amount of energy expended by the nodes along a generic path transferring data from sector A_i to the sink. Write $r_0 = 0$ and assume that A_0 is the sink node itself; since in transmitting from A_j to A_{j-1} $(2 \le j \le i)$, the amount of energy spent is $(r_j - r_{j-1})^\alpha + c$, it follows that

$$\varepsilon_i = \sum_{j=1}^{i}\left[(r_j - r_{j-1})^\alpha + c\right] \tag{4.22}$$

Recall the Lagrange identity [ref. 52, p. 64]:

$$\sum_{1\le p<q\le i}(a_p b_q - a_q b_p)^2 = \left(\sum_{p=1}^{i} a_p^2\right)\left(\sum_{p=1}^{i} b_p^2\right) - \left(\sum_{p=1}^{i} a_p b_p\right)^2$$

For every j $(1 \le j \le i)$, write $a_j = (r_j - r_{j-1})^{\alpha/2}$ and $b_j = 1$. Noticing that

- $\sum_{p=1}^{i} a_p^2 = \varepsilon_i - ic$
- $\sum_{p=1}^{i} b_p^2 = i$

and substituting in Langrage's identity, we obtain

$$\sum_{1\le p<q\le i}(a_p - a_q)^2 = i(\varepsilon_i - ic) - \left(\sum_{p=1}^{i} a_p\right)^2$$

[8]For convenience of notation we write $r_0 = 0$ and interpret A_0 as the sink itself.

Thus, we can write

$$i(\varepsilon_i - ic) = \sum_{p=1}^{i} (a_p)^2 + \sum_{1 \le p < q \le i} (a_p - a_q)^2 \tag{4.23}$$

Clearly, the left-hand side of equation (4.23) is minimized whenever

$$\sum_{1 \le p < q \le i} (a_p - a_q)^2 = 0$$

which occurs if and only if

$$a_1 = a_2 = a_3 = \cdots = a_i$$

Now, recalling that the optimal value of r_1 from equation (4.10) is

$$r_1^{opt} = \begin{cases} t_x & \text{if } \alpha = 2 \\ \min\left\{ \left(\dfrac{2c}{\alpha - 2} \right)^{1/\alpha}, \; t_x \right\} & \text{if } 2 < \alpha \le 6 \end{cases}$$

We can set for every

$$\begin{aligned} &j (1 \le j \le i), \\ &r_j - r_{j-1} = r_1^{opt} \end{aligned} \tag{4.24}$$

It is easy to see that equation (4.24) implies

$$r_i = i \times r_1^{opt} \tag{4.25}$$

and so, substituting in equation (4.23), we obtain

$$\varepsilon_i = i \times \min\left\{ \frac{c\alpha}{\alpha - 2}, \; t_x^{\alpha} + c \right\}$$

To summarize, we state the following result.

Theorem 4.2 In order to minimize the total amount of energy spent on routing along a path originating at a sensor in corona A_i and ending at the sink, all the coronas must have the same width and the optimal amount of energy is i times the energy needed to send the desired information between adjacent coronas.

4.10 CONCLUDING REMARKS AND DIRECTIONS FOR FURTHER WORK

In this chapter we have proposed a general-purpose virtual infrastructure for a massively deployed collection of anonymous sensors. The key component of the virtual infrastructure is a dynamic coordinate system that suggests a simple and robust clustering scheme. We have also shown that *training* the sensors—the process of learning their coordinates—can be performed by a protocol that is lightweight. Being energy efficient, this training can be repeated on either a scheduled or ad hoc basis to provide robustness and dynamic reorganization.

We also showed that in a trained wireless sensor network the tasks of routing and data aggregation can be performed by very simple and energy-efficient protocols.

It is important to point out that Olariu et al. [27] have shown that the virtual infrastructure can be leveraged by a number of applications, including in-network data storage and security-related problems. This is an extremely important problem, as the information provided by the sensor network may be used for decision making in military or civilian environments where human life is at stake.

The genetic material discussed in Subsection 4.2.1 has many other applications. One of then is *generational learning* discussed in [53,54] in the context of modeling wireless sensor networks, and by Jones et al. [55] in the context of biology-inspired protocols for wireless sensor networks.

REFERENCES

1. C. C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris. WiseNET: An ultralow power wireless sensor network solution. *Computer (IEEE)*, **37**(8):62–69, 2004.
2. See at http://www.darpa.mil/mto/mems/.
3. See at http://www.stanford.edu/class/ee321/ho/MEMS-14-sensors.pdf.
4. See at http://www.xs4all.nl/ganswijk/chipdir/m/sensor.htm.
5. V. V. Zhirnov and D. J. C. Herr. New frontiers: Self-assembly and nano-electronics. *Computer (IEEE)*, **34**(1):34–43, 2001.
6. J. Hill, M. Horton, R. Kling, and L. Krishnamurthy. The platforms enabling wireless sensor networks. *Communications of the ACM*, **47**(6):41–46, 2004.
7. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, **38**(4):393–422, 2002; *IEEE Wireless Communications*, **9**(1):40–48, 2002.
8. J. M. Kahn, R. H. Katz, and K. S. J. Pister. Mobile networking for smart dust. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Computing and Networking (MobiCom'99)*, Seattle, Washington, August 1999.
9. B. Warneke, M. Last, B. Leibowitz, and K. Pister. SmartDust: Communicating with a cubic-millimeter computer. *Computer (IEEE)*, **34**(1):44–51, 2001.
10. D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *Computer (IEEE)*, **37**(8):41–49, 2004.

11. D. Culler and W. Hong. Wireless sensor networks. *Communications of the ACM*, **47**(6): 30–33, 2004.
12. National Research Council. *Embedded, Everywhere: A Research Agenda for Systems of Embedded Computers* Committee on Networked Systems of Embedded Computers, for the Computer Science and Telecommunications Board, Division on Engineering and Physical Sciences, Washington, D.C., 2001.
13. P. Saffo. Sensors, the next wave of innovation. *Communications of the ACM*, **40**(2):93–97, 1997.
14. J. Agre and L. Clare. An integrated architecture for cooperative sensing networks. *IEEE Computer*, **33**(5):106–108, 2000.
15. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, **11**(1): February, 2003.
16. K. Martinez, J. K. Hart, and R. Ong. Sensor network applications. *Computer (IEEE)*, **37**(8):50–56, 2004.
17. C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo. Sensor information networking architecture and applications. *IEEE Personal Communications*, pages 52–59, August 2001.
18. R. Szewczyk, E. Osterweil, J. Polatre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, **47**(6):34–40,
19. S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless micro-sensor network models. *Mobile Computing and Communications Review*, **6**(2):28–36,
20. G. J. Pottie and W. J. Kaiser. Wireless integrated sensor networks. *Communications of the ACM*, **43**(5):51–58, 2000.
21. K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, pages 16–27, October 2000.
22. L. Wang and S. Olariu. Towards a general-purpose virtual infrastructure for mobile aD-HOC networks. In *Ad Hoc and Sensor Networks*, Y. Xiao and Y. Pan (eds.), Nova Science Publishers, January 2005.
23. K. H. Chan, A. Perrig, and D. Song. Random key pre-distribution schemes for sensor networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley, California, May 2003.
24. S. Roundy, P. K. Wright, and J. Rabaey. *Energy Scavenging for Wireless Sensor Networks with Special Focus on Vibrations*. Kluwer Academic Press, 2004.
25. N. S. Shenck and J. A. Paradiso. Energy scavenging with shoe-mounter piezoelectrics. *IEEE Micro*, **21**:30–41, 2001.
26. K. Jones, A. Wadaa, S. Olariu, L. Wilson, and M. Eltoweissy. Towards a new paradigm for securing wireless sensor networks. In *Proceeings of the New Security Paradigms Workshop (NSPW'2003)*, Ascona, Switzerland, August 2003.
27. S. Olariu, A. Wadaa, L. Wilson, and M. Eltoweissy. Wireless sensor networks: Leveraging the virtual infrastructure. *IEEE Network*, **18**(4):51–56, 2004.
28. M. Sichitiu and C. Veerarithiphan. Simple accurate synchronization for wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2003)*, New Orleans, Louisiana, March 2003.
29. F. Sivrukaya and B. Yener. Time synchronization in sensor networks: A survey. *IEEE Network*, **18**(4):45–50, 2004.

30. N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications*, **7**(5):28–34, 2000.
31. N. Bulusu, J. Heidemann, and D. Estrin. Scalable coordination for wireless sensor networks: Self-configuration localization systems. In *Proceedings of the 6th International Symposium on Communication Theory and Applications (ISCTA 2001)*, Ambleside, Lake District, UK, July 2001.
32. S. Capkun, M. Hamdi, and J.-P. Hubeaux. GPS-free positioning in mobile ad-hoc networks. *Cluster Computing*, **5**(2):157–167, 2002.
33. L. Girod, V. Bychkovskiy, J. Elson, and D. Estrin. Locating tiny sensors in time and space: A case study. In *Proceedings of the International Conference on Computer Design (ICCD 2002)*, Freiburg, Germany, September 2002.
34. L. Doherty, H. S. J. Pister, and L. E. Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of IEEE INFOCOM 2001*, **3**:1655–1663, April 2001.
35. D. Niculescu. Positioning in ad hoc sensor networks. *IEEE Network*, **18**(4):24–29, 2004.
36. C. Savarese, J. Rabaey, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proceedings of the USENIX Technical Annual Conference*, pages 317–328, Monterey, California, June 2002.
37. K. Langendoen and N. Reijers. Distributed localization algorithm. In *Embedded Systems Handbook*, R. Zurawski (ed.), CRC Press, forthcoming.
38. S. Bandyopadhyay and E. Coyle. An efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of IEEE INFOCOM 2003—The Conference on Computer Communications*, **22**(1):1713–1723, March 2003.
39. D. Coore, R. Nagpal, and R. Weiss. *Paradigms for Structure in an Amorphous Computer*, MIT Artificial Intelligence laboratory Technical Report AI-1616, October 1997.
40. S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh. Optimal energy-aware clustering in sensor networks. *Sensors*, **2**:258–269, 2002.
41. A. Wadaa, S. Olariu, L. Wilson, K. Jones, and Q. Xu. On training wireless sensor networks. In *Proceedings of the 3rd International Workshop on Wireless, Mobile and Ad Hoc Networks (WMAN'03)*, Nice, France, April 2003.
42. A. Wadaa, S. Olariu, L. Wilson, K. Jones, and M. Eltoweissy. Training a sensor networks. *Mobile Networks and Applications*, February 2005, forthcoming.
43. D. Braginsky and D. Estrin. Rumor Routing Algorithm for Sensor Networks. Paper submitted to the International Conference on Distributed Computing Systems (ICDCS-22), November 2001.
44. D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly resilient, energy-efficient multipath routing in wireless sensor networks. *ACM Mobile Computing and Communications Review*, **5**(4), 2001.
45. J. Kulik, W. Heinzelman, and H. Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*, **8**(3), March 2002.
46. R. C. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2002)*, Orlando, Florida, March 2002.
47. E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. A physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Computing and Networking (MobiCom 2001)*, Rome, Italy, July 2001.

48. A. Woo and D. E. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Computing and Networking (MobiCom 2001)*, Rome, Italy, July 2001.

49. W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Computing and Networking INFOCOM 2002*, New York, June, 2002.

50. K. Nakano and S. Olariu. Randomized initialization protocols for radio networks. In *Handbook of Wireless Networks and Mobile Computing*, Stojmenović (ed.), pages 195–218, John Wiley & Sons, 2002.

51. K. Nakano and S. Olariu. Uniform leader election for radio networks. *IEEE Transactions on Parallel and Distributed Systems*, **13**:516–526, 2002.

52. R. G. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*, Addison-Wesley, 1989.

53. D. Gracanin, M. Eltoweissy, S. Olariu, and A. Wadaa. On Modeling Wireless Sensor Networks. Paper presented at 18th International Parallel and Distributed Processing Symposium (IPDPS '04), Workshop 12: Fourth International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), Santa Fe, NM, April 2004.

54. D. Gracanin, M. Eltoweissy, S. Olariu, and A. Wadaa. Dependability support in wireless sensor networks. In *Dependable Systems*, H. Diab and A. Y. Zomaya (eds.), John Wiley & Sons, 2005.

55. K. Jones, K. N. Lodding, S. Olariu, A. Wadaa, L. Wilson, and M. Eltoweissy. Biomimetic models for wireless sensor networks. In *Handbook of BioInspired Algorithms*, S. Olariu and A. Y. Zomaya (eds.), CRC Press, 2005.