# 13

# WIRELESS SENSOR NETWORKS

13.1 Introduction
13.2 Sensor Network Applications
13.2.1 Habitat Monitoring
13.2.2 Structural Health Monitoring
13.2.3 Miscellaneous Applications
13.3 Sensor Network Architecture and Sensor Devices
13.3.1 Sensor Network Architecture
13.3.2 Overview of Sensor Devices
13.3.3 Commercial Sensors
13.3.4 Future Directions
13.4 The Physical Layer in Sensor Networks
13.4.1 Spectrum
13.4.2 Path Loss
13.4.3 Gray Zone
13.4.4 Modulation Schemes
13.5 The MAC Layer in Sensor Networks
13.5.1 Issues in Medium Access for Sensor Networks
13.5.2 IEEE 802.15.4 Medium Access Control
13.5.3 Low-Duty-Cycle Medium Access Controls
13.5.4 Low-Latency Medium Access Controls
13.6 Higher Layer Issues in Sensor Networks
13.6.1 Establishing the Sensor Network
13.6.2 Routing
13.6.3 Coverage, Connectivity, and Topology Control
13.6.4 Synchronization
13.6.5 Security

## 13.1 INTRODUCTION

Over the last decade, sensor networks and applications relying on sensor networks have become prevalent, as also has research in academia and a burgeoning industry in sensor networking devices. Sensor devices have evolved over the last decade to support various

*Networking Fundamentals: Wide, Local and Personal Area Communications*, by Kaveh Pahlavan and Prashant Krishnamurthy

applications, such as asset monitoring, surveillance, structural health monitoring, habitat monitoring, and even underwater sensing. In this chapter, we provide an overview of wireless sensor networks.

We start in Section 13.2 by describing several applications that have considered implementations of sensor networking, such as habitat monitoring and structural health monitoring. In Section 13.3 we provide an overview of sensor network architectures. We also discuss wireless sensor devices, the actual hardware that makes sensor networks work, and some of the platforms that sensor devices employ. Sections 13.4 and 13.5 discuss the physical and MAC layers of sensor networks and Section 13.6 provides a discussion of higher layer issues for sensor networks. In these sections, some of the discussion is devoted to IEEE 802.15.4 low-rate wireless personal area networking devices and the standard. This has some commonality with the treatment of Zigbee and 802.15.4 in Chapter 10. We include this redundancy in treatment for completeness and to have a self-contained chapter on sensor networks.

## 13.2 SENSOR NETWORK APPLICATIONS

Sensor network applications are diverse, ranging from habitat monitoring (e.g. studying the climate on redwood trees and their impact on the ecosystems therein) to surveillance and physical intrusion detection (e.g. detecting breaking into museums). Applications can be of academic or scientific interest, or commercial where the sensor network can have significant impacts; for example, in improving crop yields. There still appears to be a significant disconnect between the actual deployment of sensor networks for real applications and academic research in sensor networking as discussed by Raman and Chebrolu [Ram08]. We ignore this issue in this chapter. In the following subsections, we look at examples where sensor networks have been or are being actively deployed. Where possible, we link the applications to the networking issues in subsequent sections.

### 13.2.1 Habitat Monitoring

Habitat monitoring is a scientific application that has benefited significantly from the deployment of sensor networks. In habitat monitoring applications, it is necessary to monitor a variety of environmental characteristics, such as temperature, humidity, barometric pressure, and other physical parameters, over significantly long periods of time and/or significantly large geographical areas. When such characteristics are monitored underneath the soil surface, inside a lake, on the surface of a tree, etc., the corresponding conditions are referred to as "microclimates."

An example of habitat monitoring using sensor networks, widely cited in the literature [e.g. Sze02a, Sze02b, Hu07], is that of understanding the impact of microclimates on habitat selection by sea birds. In particular, the objective of this project was to improve the understanding of sensor networks by using them to monitor the occupancy of nesting burrows on Great Duck Island in Maine. The hope here was that passive IR sensors used to detect heat from birds that were nesting and measures of temperature and humidity to indicate extended inhabitation of burrows could eventually replace laborious manual

sampling and direct inspection of the island. A two-tier sensor network (see Example 13.3) was deployed by researchers over a 4-month period with 150 sensor devices. The researchers evaluated the performance of the sensor network, its lifetime, how nodes failed, and how nodes could be recovered after the completion of the project (about 50% of the devices were recovered).

Sensor networks are extremely useful for monitoring physical phenomena over large geographical areas requiring dense deployment of sensors. One such example, of what has been called a "macroscope" because of its similarity to a microscope in terms of its ability to reveal complex details, is the monitoring of air temperature, relative humidity, and amount of active solar radiation along the length of a 70 foot ($\sim$21.3 m) coastal redwood tree [Tol05]. The microclimate along a redwood tree can have significant differences, both spatially and temporally. The treetop gets sunlight, while the bottom is typically moist and cool because of the shade from leaves. The tree also moves large volumes of water from the soil upwards and eventually into the air. A wireless sensor network that collected data over 44 days, every 5 min, with nodes placed every 2 m along the redwood tree is described by Tolle *et al.* [Tol05]. The monitoring and subsequent analysis of data did reveal dynamic gradients of phenomena, as expected by the biologists.

An example of a mobile sensor network for habitat monitoring is the ZebraNet project [Ju02], where a sensor network was deployed to study the behavior of zebras in the Mpala Research Centre in Kenya. The idea in this project was that sensors equipped with GPS and peer-to-peer ad hoc networking can yield more information about animal behavior than simple radio collars that are sampled during the day by researchers driving around the natural area. No fixed infrastructure was available, which makes this network different from the previous examples.

### 13.2.2 Structural Health Monitoring

Structural health monitoring refers to the continual or periodic monitoring of the *health* of large structures such as bridges, buildings, or ships. The vibration data from bridges can be used to detect the health of bridges (e.g. whether it is ambient vibrations or some other serious condition). Examples of projects that have looked at monitoring bridges include those of Xu *et al.* [Xu04] and Kim *et al.* [Kim06]. Kim *et al.* [Kim06] monitored a 260 foot ($\sim$80 m) long suspension footbridge using 13 sensors that measured the vibration of the bridge using accelerometers and proposed extending the work to the Golden Gate Bridge in San Francisco.

### 13.2.3 Miscellaneous Applications

Other applications of sensor networks include asset monitoring, surveillance, etc. As described in Section 13.3, many vendors now carry sensor devices and the corresponding networking infrastructure to perform asset monitoring.

## 13.3 SENSOR NETWORK ARCHITECTURE AND SENSOR DEVICES

We start with a brief overview of the typical architecture of a sensor network to provide a view of how sensor devices fit into the network. Our objective in this section is to discuss the

capabilities of sensor devices and the different commercial product platforms that are available on the market. We do not explore the details of the integrated circuit chips, radios, or other components on the sensor network platforms in great detail, except as necessary to illustrate their capabilities. Section 13.3.2 includes a brief history of sensor devices and their development. Section 13.3.3 describes a variety of commercially available sensor devices. Section 13.3.4 considers the evolution of sensor devices in the future and emerging issues.

### 13.3.1 Sensor Network Architecture

A typical sensor network architecture consists of a *sensor field*, which is the physical environment where the sensor nodes or devices are deployed (see Fig. 13.1). Sensor nodes can possibly be deployed in extremely large numbers, on the order of thousands of sensor nodes in the field. Consequently, the cost of these nodes should be low. A low-cost device can thus be expected to have fairly limited computational and communication capabilities, considering the fact that sensing capabilities are also to be included in the device. Moreover, in many applications, sensor nodes are deployed in hostile areas or physically inaccessible regions where it is not easy to have human intervention to maintain sensor nodes. Such sensor nodes have to operate on limited battery power and the batteries cannot be replaced easily. In such cases, sensor nodes have to be designed so that power-consuming operations such as the central processing unit or the radio used for communications are shut down when they are not being used. Of course, for specific applications (e.g. physical intrusion detection using cameras), sensor nodes may have more advanced capabilities. Thus, sensor devices may range from millimetre-sized devices fabricated on custom silicon to more general purpose cell-phone-sized devices with advanced capabilities.

Figure 13.1 shows a schematic of a simple sensor network architecture. Sensor nodes with limited capabilities deployed in the sensor field communicate to a powerful BS that links them to the Internet and a central manager for processing the sensed data. Communications to the BS have to go through several sensor nodes first, because all sensor nodes will not be typically able to communicate directly with the BS. This may be due to limited communication range, distance from the BS, intermittent sensor activity, and so on.

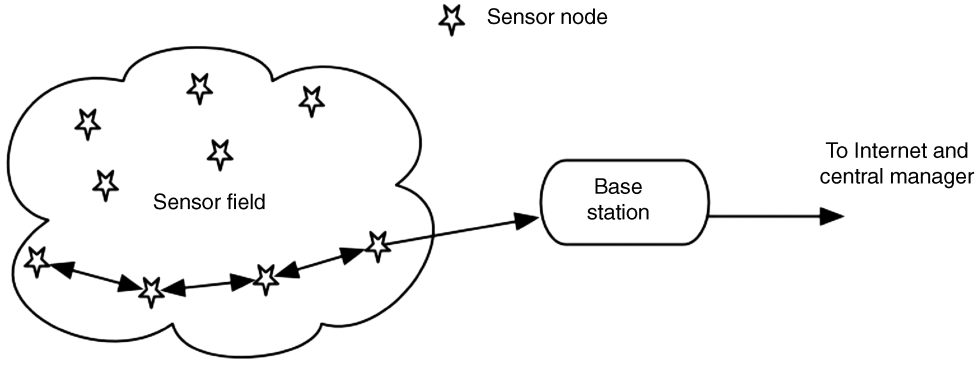


**FIGURE 13.1** Typical sensor network architecture.

The simple architecture shown in Fig. 13.1 is expanded upon by Hill *et al.* [Hil04], where four classes of sensor networking devices are described. At the lowest level of the hierarchy, the actual sensing device could be very specialized, with a tiny form factor and very limited capabilities. Such devices may not even be capable of receiving information and may simply transmit information when they sense an event or perform other application-related activity. We will call these devices *submotes* in this chapter for ease of reference.

It is the responsibility of the second tier in the hierarchy (called the *mote* class) to receive this information and convey it using multiple hops towards the gateway or BS that forms the highest level in the hierarchy. The word "mote" itself means a tiny piece of substance, but the mote-class sensor device is larger in form factor and capability than the very basic sensing device. Figure 13.1 explicitly shows only the mote-class device and the BS.

The third device described by Hill *et al.* [Hil04] is a sensor device that is more complex than a mote in terms of its ability to communicate using higher bandwidths (e.g. using Bluetooth radios) and may have more random access memory (RAM) on its chip, and a processing unit that is more powerful than the one on the mote-class device. We will refer to this as a *supermote* in this chapter for easy reference.

The gateway or BS is the most powerful sensor device in the network. It typically has two interfaces: one that can connect to a mote-class device or a supermote device and the other that can connect to a larger network (cell-phone network, WLAN, or a wired LAN). The gateway may have other processing and storage capabilities that will be useful for the sensor application.

***Example 13.1: Redwood Microclimate Monitoring*** In the Redwood "macroscope," mote-class sensor devices were deployed along the length of a redwood tree. The mote-class devices were equipped with sensors that measured total solar radiation by inspecting the spectrum and barometric pressure, but pressure measurements were not used. The solar radiation impacts photosynthesis and, thus, was a quantity of interest. Motes were deployed from about 15 m above the ground to 70 m above the ground on the west side of the tree very close to the tree trunk. The sensors delivered data to a BS-class device that connected to the Internet using a GPRS cellular network (see Chapter 7 for a discussion of data services over cellular networks).

***Example 13.2: Hybrid Sensor Networks for Cane Toad Monitoring*** Monitoring of cane toad populations in Australia using sensor networks is reported by Hu *et al.* [Hu07]. This application is resource intensive, in that several FFTs have to be computed by the sensor devices to identify vocal characteristics of toads against background and other noise. One approach suggested by Hu *et al.* [Hu07] is to employ supermote- or BS-class devices that have more computational and communication resources. This approach is quite expensive due to the cost of these devices. Instead, a hybrid network of mote-class devices and BS-class devices can make the system more cost effective. In the work described by Hu *et al.* [Hu07], mote-class devices deployed on a larger scale take acoustic samples, perform some preliminary processing (compression) to reduce communication costs (see Section 13.6 on data aggregation and in-network processing), and send them to BS-class devices. The BS-class sensor devices use the received information from mote-class devices to determine the existence of cane toads.

***Example 13.3: Network Architecture in Great Duck Island*** A tiered architecture was implemented on Great Duck Island [Sze02a] to monitor seabird occupancy. One network used single-hop transmissions from mote-class devices to a BS-class device. A second network used a multi-hop topology where mote-class sensors would route information to a BS-class device. Two types of mote were deployed: burrow motes that had IR and temperature/humidity sensors for use inside burrows and weather motes that monitored temperature, humidity, and barometric pressure on the surface. Node lifetimes were longer in the single-hop network, with most weather motes in the single-hop network being functional even after 4 months. Burrow motes performed additional sensing and had shorter lifetimes.

### 13.3.2 Overview of Sensor Devices

In this section we present a very brief history of sensor devices and also present a generic architecture of sensor devices. The actual architecture of a sensor device will be different, but the functional components can be expected to be similar.

***Brief History.*** The concept of a network of extremely small devices that can sense physical phenomena (light, temperature, vibrations, motion, etc.) or even induce some activity (actuator) is believed to have its origins in the early 1990s with military applications based on microelectromechanical systems being the primary driver for this concept. The idea of *smartdust* that can be scattered in an environment, elements of which would self-configure themselves into a network, originated in the mid 1990s through several workshops organized by DARPA in the USA. While dust-sized sensor devices are not yet available (see discussion in Section 13.3.4), research in the area of very small devices that could sense phenomena and network with one another has been going on at full steam for more than a decade now.

Much of the initial effort was dedicated to developing sensor devices using off-the-shelf components, thereby reducing the cost of development. The development of WLANs during the 1990s was also useful in terms of enhancing radio communications in unlicensed bands that could also be exploited by sensor devices. The earliest mote-class devices were built using printed circuit boards and were an inch or two in size. The *Rene* mote was developed in 1999 [Hil04]. It had 512 bytes of RAM, 8 KB of flash memory, several expansion connectors, the ability to communicate at a data rate of 10 kb/s and consumed 60 mW of power when active. It was around 1999 that Bluetooth was also reaching stability as a cable replacement technology. In parallel, work on developing an operating system and programming tools for sensor devices was ongoing. The TinyOS operating system [Tiny] and the nesC programming language [Gay03] are examples of this development. Recently, Sun Microsystems has advocated the use of Java 2 micro edition as the software platform for sensor nodes. The early commercial off-the-shelf sensor devices used their own nonstandard radios. The IEEE 802.15.4 working group started standardization of the 802.15.4 physical and MAC layer standards for low-rate WPANs in the early 2000s and was completed in 2003. Most sensor nodes of today use the 802.15.4 standard for medium access and radio communications at the physical layer. The Zigbee standard [Whe07] (ratified in 2004) considers standardization of higher layer issues (such as routing, addressing, and application messaging) for embedded sensors, and many commercial vendors have now adopted the standard.

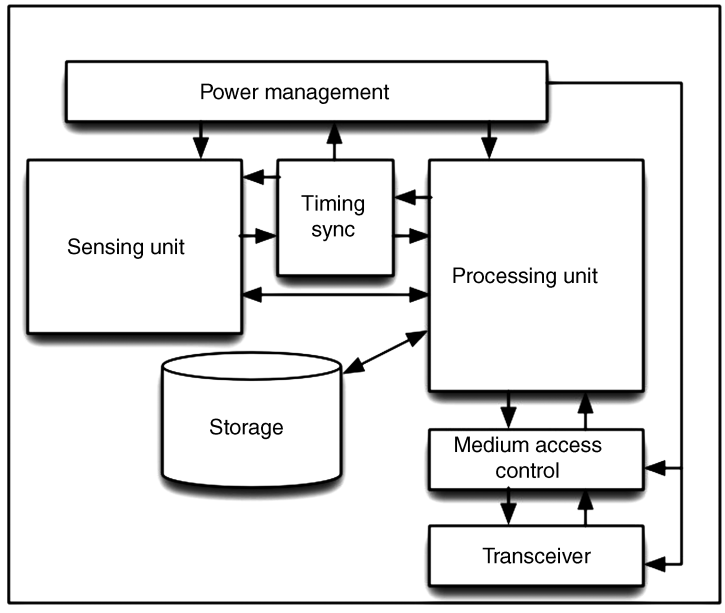


**FIGURE 13.2** Simple schematic of a sensor device.

***Architecture of a Sensor Device.*** Figure 13.2 shows a simple schematic of the internal architecture of a sensor device. The interested reader is referred to [Coo06, Whe07, Aky02] for other examples of architectures. In the simple schematic shown here, a sensor device primarily has a sensing unit, a processing unit, and a power management unit. The sensing unit performs the actual sensing tasks (e.g. detecting changes in temperature). The processing unit (with some internal RAM) is responsible for performing computations on the sensed data in conjunction with a storage unit. It is also responsible for handling communications (by running the operating system code) and working collaboratively with other sensor nodes towards accomplishing the application objectives. The power management unit is important because of the necessity to reduce the power consumption in the sensor node to the maximum extent possible. Also shown in Fig. 13.2 are the storage and time synchronization components. Usually, the storage on sensor devices is comprised of flash memory. A MAC unit works with the transceiver to access the shared air interface. Not shown in the figure are potential external sensing units, the battery that actually powers the sensor device, the antenna, or other components, like those used for localizing a sensor (determining its location).

***Components and Operation.*** In the same spirit of general overview, we do not discuss the details of the hardware used to make up sensor devices, except at a high level. We also avoid providing an exhaustive list of chips used in sensors. Some details are available in Section 13.3.4.

The processing units are typically simple. Some sensor devices use custom chips (e.g. the submote-class *spec* sensor [Hil04]), while others employ low-power off-the-shelf microcontrollers from ATMEL [Atmel], Jennic [Jennic], Texas Instruments, or Intel. A common standard 802.15.4-compliant radio from Chipcon (now part of Texas Instruments) is used

for the transceiver MAC. Electrically erasable programmable read-only memory or flash memory storage is commonly employed in most sensor devices.

TinyOS is the most common operating system that is run on sensor devices. However, other operating systems are also used in some sensors, such as J2ME, Contiki [Cont], and MantisOS [Mant]. TinyOS was developed at the University of California at Berkeley and has been explicitly designed to support concurrency operations on embedded devices and it is open source. TinyOS has been used in the sensors deployed in several projects, including the monitoring of seabird occupancy on Great Duck Island described in Section 13.2. TinyOS 2.0 is a clean-slate design that foregoes backwards compatibility with the earlier versions to overcome their limitations. MoteWorks is a commercial sensor network software development environment from Crossbow (see Section 13.3.3) that is built using TinyOS 1.1 as the basis. Contiki is an open source operating system with small memory requirements, a microIP stack for TCP/IP communications, and a lightweight communication stack for low-power radios that supports multi-hop communications. MantisOS (also open source) was developed at the University of Colorado at Boulder and is a multithreaded operating system for sensors written in C. Many mote-class sensor devices can run any of the three operating systems.

### 13.3.3 Commercial Sensors

In this section we consider some example commercial sensor devices, their capabilities, and application environments (recommended by the vendors). We look at a couple of popular vendors and their products and cite appropriate references to the devices. This is not intended to be an exhaustive list, but more of a representative list to illustrate the current state of the art. As we discuss below, most sensor devices available on the market use the IEEE 802.15.4 standard for the lowest two layers of communications (the MAC and physical layers). Heterogeneity exists at higher layers, with different vendors selecting different options, such as Zigbee, WirelessHART, and 6LoWPAN. Some vendor products have the ability to organize themselves into a mesh and perform multi-hop routing to a manager or BS. Finally, there are several customized sensor nodes that have been implemented [e.g. Leo03, Sam06, Ayl07] in academic projects. An exhaustive discussion of such sensor devices is beyond the scope of this chapter.

***Crossbow.*** One of the earliest vendors of mote-class sensor devices, Crossbow [Xbow] now makes a variety of sensor devices and supports applications in industrial automation, environmental or climate monitoring, and asset tracking. Crossbow's catalog consists of evaluation and development kits, end-user mote systems, and gateway or BS-capable sensor nodes. There are numerous types of sensor devices, evaluation kits, boards, and processor/radio units in Crossbow's catalog that are suitable for specific applications or for research and development. Different units have different capabilities (e.g. combinations of ability to sense quantities such as light, pressure, or temperature, GPS, accelerometers, external analog inputs, etc.). Some of the units come with software that enables them to create a self-healing mesh network (the software is called Xmesh by Crossbow). They have support for the commercial software development tool (MoteWorks). Most of the units are IEEE 802.15.4 compliant and operate in the 2.4 GHz ISM unlicensed bands (see Chapter 3 for more details). However, some units also operate in the

888 or 916 MHz unlicensed bands. We briefly consider some products from Crossbow in more detail.

The TelosB mote from Crossbow is especially suitable for research and development with support for TinyOS. It has a Texas Instruments microcontroller with 10 KB of RAM, additional flash memory, two AA batteries, and the ability to collect data using a USB port. This port can also be used for programming the sensor. TelosB operates in the 2.4 GHz bands.

The Cricket Mote includes both an RF transceiver and an ultrasound transceiver. It is an example of a sensor that has two different transceivers. The latter can be used for obtaining ranging information using the time of flight of the ultrasound signal. Ranging information can be used to localize sensor nodes after deployment. It is an enhancement of Crossbow's low-power Mica2 mote, which operates in the 868/916 MHz bands.

The Stargate BS from Crossbow includes a 400 MHz Intel processor, an embedded Linux platform, the ability to communicate using Ethernet or other kinds of interfaces using PCMCIA connectors, and support for communications with mote-class sensor devices.

***Dust Networks.*** Like Crossbow, Dust Networks is a vendor targeting applications such as process control, asset management, environmental monitoring, and health safety monitoring [Dust]. The SmartMesh products from Dust Networks include those specially designed for industrial automation, low power (using system-on-a-chip), and for harsh wireless environments. Most of the products operate using IEEE 802.15.4 radios in the 2.4 GHz ISM bands, although some of the products operate around 900 MHz. Evaluation kits are also available.

The SmartMesh-IA products from Dust Networks, built for industrial automation, also employ the WirelessHART protocol [HART]. Several million intelligent instruments in the process control industry already employ the HART communication protocol for configuration, status checks, and exchanging information using wired networking. The Wireless HART protocol also allows HART-enabled devices to communicate with sensors. It is reliable and secure and employs a combination of mesh networking for redundancy, channel hopping to avoid interference, time-synchronized messaging, and 128-bit encryption and authentication.

***Sun.*** Sun Microsystems, which has been a major player in the computer industry in developing chips, operating systems, and in enabling services, has invested in the so-called "Small Programmable Object Technology" (SPOT) as part of its research and development efforts [Sunspot]. SunSPOT sensors are not being sold with particular commercial applications in mind, but rather as more powerful wireless platforms running a Java virtual machine that can perform complex tasks for use in exotic applications such as swarm intelligence or rocket launch monitoring. Sun makes its sensor nodes available for education, government, military, and industrial applications, as well as for hobbyists. SunSPOT sensors are IEEE 802.15.4 compliant.

***Others.*** Other vendors of sensor networks include Ember Corporation, Sensinode [Sensinode] in Finland, and Sentilla, which is focused on software solutions for sensors using Java. Companies such as Jennic manufacture integrated solutions for Zigbee where an

**TABLE 13.1 List of Some Vendors of Sensor Devices and Features Of Some Products**

| Company | Types of sensor device | Frequency bands | Standards supported | Applications supported |
|---|---|---|---|---|
| Crossbow | Evaluation kits, BSs, motes, data-acquisition boards | 2.4 GHz, 868 MHz, 916 MHz | 802.15.4, ZigBee | Asset monitoring, climate control, surveillance |
| Dust Networks | Evaluation kits, motes, managers | 2.4 GHz, 902–928 MHz | 802.15.4, WirelessHART | Process control, asset monitoring, health safety monitoring |
| Ember Corporation | ZigBee chips, software, development tools | 2.4 GHz | 802.15.4, ZigBee | Home and building automation, asset management, defense applications |
| Sensinode | Development kits, Nanoseries sensors and routers | 2.4 GHz | 802.15.4, 6LoWPAN | Hospital asset management |

802.15.4 radio is integrated with a microcontroller and software that supports Zigbee. Such products are not marketed as complete sensor networking solutions, but can be used by other companies to develop their own sensor network solutions for specific applications.

Table 13.1 lists some of the vendors, their products, and targeted applications.

### 13.3.4 Future Directions

In this section we briefly discuss some future possibilities for sensor devices that are being explored in the research literature (and in practice). We first consider the reduction in form factor of sensor devices, which can result in efficient use of computing and energy resources. Then we consider mobility and other issues.

***Reducing the Size of Sensor Devices.*** Sensor devices are becoming smaller and yet more capable with time in a manner similar to computers, which started out as being huge devices in the second half of the twentieth century and whose capabilities are now matched by today's laptops and PDAs. One of the technologies driving the miniaturization of sensors is the ability to fabricate an entire "system" on a single silicon chip. Today's sensor devices include several commercial off-the-shelf components that have been "put together" to create the device. While this reduces the cost of manufacturing sensor devices, such devices do not have the extremely small form factors required for some applications and they are quite inefficient in the use of power and other resources. If all the digital (processing and communication), analog (typically sensing), and RF (communication) components can be integrated onto a single chip, then the form factor of sensors can be reduced significantly. For example, some of the SmartMesh products claim to be more efficient than other comparable 802.15.4 devices because of tight integration of components on a chip. The feasibility of a single-chip sensor mote is discussed by Cook

*et al.* [Coo06]. Issues of integrating elements such as the antenna for wireless communications and quartz crystal for timing and synthesis of high-frequency signals are challenges that are difficult to overcome when considering integration of RF components on a chip. The sensing capabilities may also be hard to integrate with the other components on a single chip, as also is the energy source.

At the extreme, sensor devices are expected to be of the size of specks of dust. Nanosensors employing nanotechnology are considered by Sailor and Link [Sai05] in their discussion of sensor devices that are really of the size of dust motes. We recall that a sensor has to perform sensing, processing, and communications. Consequently, identifying or addressing a sensor device is important. Sailor and Link [Sai05] discuss the technique of etching spectral "bar codes" (that can be scanned using lasers at distances of up to 20 m) on porous silicon dust for identifying sensors the size of dust. Similarly, sensing with such small form factors is difficult, but it is possible for some applications, such as sensing the presence of biological molecules. Processing and communications are also potentially possible at the nanoscale.

***Mobile Sensors.*** Deploying sensors that can move to better locations for sensing or for delivering information can benefit applications that deploy a large number of sensors. Mobile sensors can improve the application performance as well as improve the efficiency of deployment. Mobility, however, is costly, especially since it is difficult to guide, monitor, and modify the movement of sensor devices. Typically, only large sensors are mobile and employ robots for moving the sensor devices, although efforts have been made to reduce the size of such robots [Ber00, Sib02]. Alternatively, BS-class sensors can be made mobile so that they can move around the deployment area to gather data from mote-class or submote-class sensors. Such sensors have been called "data mules" in the research literature [Sha03].

The real challenges occur when submote-class sensors or sensors that are the size of dust specks need to be mobile. As discussed by Sailor and Link [Sai05], there are great challenges in enabling autonomous mobility of sensors that are the size of specks of dust, even if the mobility is random. It is possible, however, to enable *directed* mobility, where sensors can be made to move in specific directions using the environment in which they are deployed or external mechanisms or forces such as electrical, magnetic, or photonic stimuli. Work in all areas related to making extremely small sensor devices mobile is in the nascent stages. If and when mobile sensors that are of the size of dust specks become reality, one can expect several new applications to emerge.

***Other Issues.*** An important issue with sensor devices owing to their small size, limited resources, and reduced functionality is the ability to maintain and verify consistency of applications and reprogrammability of the applications or retasking the sensors as needed. Such needs may include changing some applications in response to the environment and maintaining scalability as more sensors are deployed. A short survey of software approaches for sensor devices is presented by Hadim and Mohamed [Had06]. Recent work is focusing on improving and enabling such reprogrammability of sensors in an efficient manner. For example, a dynamic operating system SOS is described by Han *et al.* [Han05] that has a common kernel for all sensor nodes. Applications can be dynamically loaded at run time in SOS. This is different, for instance, from TinyOS, which is statically compiled. Consequently, the entire system image must be redistributed if any

changes need to be made in the applications, making it expensive in a large-scale sensor network.

## 13.4 THE PHYSICAL LAYER IN SENSOR NETWORKS

In sensor networks, the PHY layer has to be simple for several reasons. The cost of fabricating a sensor should be low if several hundreds or thousands of devices have to be deployed to satisfy application objectives. At the same time, a sensor has to have a small form factor, implying that the transceiver cannot be made very complex. Chapters 2–4 provide a detailed treatment of radio propagation, modulation, and error control coding. In this section, we discuss aspects of these in relationship to sensor networks.

### 13.4.1 Spectrum

The characteristics of radio propagation depend upon the frequency of operation, as discussed in Chapter 2. At frequencies beyond 500 MHz or so, propagation of radio waves can be approximated as if they are optical rays propagating from the transmitter. While this somewhat simplifies theoretical analyses and simulations, the fact remains that radio propagation is extremely site specific and frequency specific. In general, the higher the frequency is, the larger is the attenuation of the signal strength with distance because the larger is the absorption in the surrounding medium and the larger is the loss in the antennas.

Because sensor networks are expected to be deployed by a variety of organizations for a variety of applications, it is beneficial to employ *unlicensed* frequency spectra for transmissions. The use of such spectrum does not require permission from the FCC, nor widespread testing to ensure that transmissions do not interfere with other licensed applications. Instead, a reasonable spectrum etiquette that allows low-power transmissions and multiple operators to coexist is implemented. The unlicensed spectra of choice for sensor networks are the 868 MHz band in Europe, the 916 MHz band in the USA, and the 2.4 GHz band that is available almost everywhere in the world. The 2.4 GHz band has the widest available bandwidth and is becoming the most popular choice for sensor networks. The only downside to this choice is that IEEE 802.11 WLANs, cordless phones, Bluetooth, and many other wireless devices also operate in these bands, resulting in significant interference in areas of dense deployment. Recently, the IEEE 802.15 standards committee's task group 4c has been considering standards for the 779–787 MHz bands in China and task group 4d is looking at the 950–956 MHz bands in Japan.

The base IEEE 802.15.4 standard that was approved in 2003 and extended in 2006 specifies three channels in the 868 MHz band, 30 channels in the 916 MHz bands and 16 channels in the 2.4 GHz bands (see later discussion for clarification on channels). Originally, the data rates were dependent on the frequency bands of operation. The lower frequencies supported 20 kb/s and 40 kb/s, while the 2.4 GHz bands supported up to 250 kb/s. Recent revisions to the standard enable higher data rates in all bands and an option for 100 kb/s in the 868 MHz band.

### 13.4.2 Path Loss

One of the most important parameters for design and operation of communication networks is the *transmission range* – essentially the distance up to which reliable communications is possible between a transmitter and a receiver. The *path loss* (see Chapter 2) or the reduction in signal strength as a function of distance is an important parameter that determines how far apart two sensor devices can be and still have reliable communications.

As discussed in detail in Chapter 2, in free space, where there are no obstacles, and with ideal isotropic antennas (antennas that transmit with equal power in all directions), it is possible to demonstrate theoretically that the signal strength drops as the square of the distance. In other words, if the transmit power is $P_t$ and the received power is $P_r$, then the relationship between them is $P_r = L_0 P_t / d^2$, where $d$ is the distance between the transmitter and receiver and $L_0$ is a constant that depends upon the frequency of transmission. The path loss in this case is $P_t/P_r = d^2/L_0$. However, in reality, there are obstacles in the environment and, depending on the site, the drop in the signal strength with distance can be very different. A general model for path loss that is widely used in the research literature uses a general exponent $n$ to define the drop in signal strength. That is, the relationship between the transmit power and received power is represented by

$$P_r = \frac{L_0 P_t}{d^n} \tag{13.1}$$

Writing this in decibels, the relationship becomes

$$P_r(\text{dBm}) = P_t(\text{dBm}) + L_0(\text{dB}) - 10n \log_{10}(d) = P_t(\text{dBm}) - L_p(\text{dB}) \tag{13.2}$$

The path loss is given by $L_p = 10n \log_{10}(d) - K\,(\text{dB})$. Note that the factor of 10 appears in Eq. (13.2) because the units are in decibels, not bels. Transmit and received powers are usually written in dBm, which expresses the power in comparison to 1 mW. The value of $n$ has been empirically determined for a variety of types of sites and transmission frequencies in cellular networks (e.g. big cities, rural areas, hilly areas) and WLANs. In sensor networks, the placement of the sensor determines how $n$ may behave. Sensors are typically places close to the ground and sometimes are in very cluttered environments. It is not uncommon to find values of $n$ that may range from 2 to 6 depending on the environment.

***Example 13.4: Transmission Range for Communication between Sensors*** Let the transmit power of a sensor be 1 mW (0 dBm) and the receiver sensitivity be −94 dBm. This implies that the received signal strength or received power must be at least −94 dBm, and in many cases several decibels higher for reliable reception. In the IEEE 802.15.4 standard, the receiver sensitivity is defined as the signal strength that results in a packet error rate that is smaller than 1% for a given size of a PHY layer packet. If the constant $L_0$ (sometimes called the loss at the first meter) is −45 dB at 2.4 GHz, then the distance-dependent loss can at most be 94 − 45 = 49 dB. If the signal strength drops as the third power of distance (i.e. $n = 3$), then $d = 10^{49/30} = 42$ m. In other words, the transmission range is

42 m. This compares with an indoor range of 30 m specified for a comparable vendor product. Other experiments have shown that, in outdoor areas with little clutter, the range can be as large as 75 m.

The range could be reduced due to a variety of factors. Experiments have shown that humidity, obstacles in the sensor field, the environment (e.g. wooded areas, bamboo plantations, sandy beaches can have different values of $n$), and in some cases the directionality of sensors (because their antennas as not really omnidirectional) have an impact on the transmission range. One experiment showed that the transmission range of Mica2 motes dropped from 55 m to 10 m in the presence of fog (or rain) in an outdoor area. This could have a serious impact for applications like climate monitoring.

### 13.4.3 Gray Zone

In sensor networks, like other communication networks, it is the higher layer performance that is more crucial than simpler lower layer metrics like transmission range. The question in most cases is what the packet transmission reliability (or packet success ratio) is. In most of the simulations or analysis, it is common to use the transmission range as a hard threshold. All communications within this distance are assumed to be 100% reliable and all packets that are received beyond the transmission range are assumed to be received incorrectly. If the sensor network is modeled as a graph, then this is also called the *unit disk model*.

Experiments in real sensor networks over the last few years have shown the presence of a significant *gray zone* where a model with such a hard threshold does not work well. Consider Fig. 13.3, which shows an illustrative plot (not from real data) of the packet success ratio as a function of distance between a sensor transmitter and receiver. Clearly, there is a close to 100% chance that the packet is received successfully when the

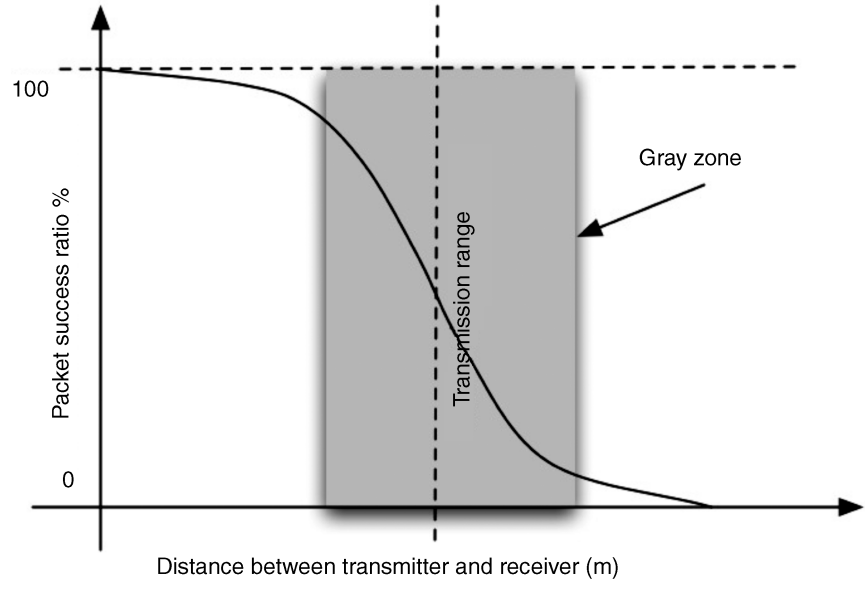


**FIGURE 13.3** Illustration of the gray zone.

transmitter and receiver are very close. As expected, a packet has no chance of successfully being delivered when the receiver is very far away from the transmitter. However, in between, the packet success ratio varies from something less than 100% to something greater than 10% or so. In most experiments, this region, where the packet success ratio is not really predictable, has been found to be of significant size. In many cases, it is found to be dominating. Unfortunately, unlike Fig. 13.3, it is not really easy to predict what the success ratio will be for a given transmitter–receiver separation, and this is very site specific.

***Example 13.5: Link Stability in the Sensor Network on Great Duck Island*** In the Great Duck Island sensor network [Sze02a], the stability of links was examined by looking at how long links existed and how often a parent node changes (the node to which data is delivered). Most links were found to have small life spans, but the stable links were used by the network to deliver most of the data. Up to 80% of the data was delivered using 20% of the links. While 55% of the time the parent node did not change, there were some occasions when the entire network topology would be modified.

Experiments have shown that the gray zone exists not only for sensor networks, but also for IEEE 802.11 WLANs. Some experiments have shown that the gray zone is less prevalent at higher data rates in IEEE 802.11 WLANs. The packet size impacts the size of the gray zone (usually, larger packets are more prone to errors). The transmission scheme and the receiver complexity and circuitry also impact the size of the gray zone, as it has an effect on how reliably the receiver can detect the transmitted data. The experimental performance of motes and IEEE 802.15.4 and IEEE 802.11 devices in several settings and scenarios can be found in the literature [e.g. Fan03, Ana05, Pet06].

Finally, the packet success ratio depends on interference. When sensor networks coexist with other technologies employing the same spectrum, the performance can drop significantly. Also, transmissions between a given pair of sensors can impact transmissions between another pair of sensors at the same time, depending on where the sensors are located.

### 13.4.4 Modulation Schemes

Some of the earliest sensors used on–off keying, as this modulation scheme is simple and easy to implement, both in the transmitter and receiver. The IEEE 802.15.4 PHY layer uses more complicated quasi-orthogonal sequences in a modification of orthogonal modulation. We discuss below the IEEE 802.15.4 PHY layer primarily from a conceptual standpoint. We ignore intricate details of framing and exact fields and formats in our treatment (see Chapter 10 for some of these details). Details of the IEEE 802.15.4 standard are available in [IEEE06]. The PHY layer in most standard technologies is responsible for functions other than simply transmitting and receiving data. Some of these functions are in support of the protocols. The 802.15.4 PHY layer, for example, also handles activation and sleep of the transceiver, detecting energy on the air for collision avoidance, providing some estimate of the quality of the link, and selecting appropriate frequency channels.

The standard defines *channel pages* and *channel numbers*. Channel pages are used to distinguish between the different possible modulation schemes. The 2.4 GHz band is associated with a single channel page (0) and has 16 channel numbers, while the 868 MHz

and 916 MHz bands are associated with three channel pages, 0, 1, and 2. The 868 MHz band has one channel number and the 916 MHz band has 10 channel numbers for a total of 3 and 30 channels respectively in these bands. However, note that, at a given time, only one possible (channel page, channel number) combination is possible. Thus, in reality, there is only a single channel in the 868 MHz band and 10 channels in the 916 MHz band. However, these channels can support different modulation schemes and, thus, different data rates. In the 868 MHz band, the channel is 0.6 MHz wide, in the 916 MHz bands it is 2 MHz wide, and in the 2.4 GHz bands it is 5 MHz wide.

In each channel, irrespective of the band, the bits that are received from the higher layers are first converted to symbols (if necessary), which are then mapped to the sequence of chips and finally the chips are modulated on a carrier. As an example, consider the 2.4 GHz bands. Here, an octet is broken into groups of 4 bits. Each group of 4 bits (there are 16 possible combinations) – now a symbol – is mapped to one of 16 quasi-orthogonal sequences of length 32 chips. The chips are then modulated using a 4-ary modulation scheme called MSK. MSK is a modified form of QPSK with a half-sine pulse shape. The modulation scheme is more bandwidth efficient than QPSK, with a rectangular pulse shape. The symbol rate is 62.5 kS/s (with 4 bits/S, the bit rate is 250 kb/s). Each symbol is converted into a 32-chip sequence. Consequently, the chip rate is 2 Mc/s.

In the 868/916 MHz bands, two base modulation schemes are allowed. If BPSK is used as the base modulation scheme, then the data rates supported are 20 kb/s and 40 kb/s respectively in the two bands. Each bit is first mapped onto a sequence of 15 chips (derived from a maximal length sequence) and each chip is transmitted using BPSK with raised cosine pulse shaping. The chip rates for the two bands are 300 kc/s and 600 kc/s respectively. With 15 chips/bit, this results in the corresponding data bit rates. An optional base amplitude-shift keying (ASK) modulation scheme allows the data rates to be increased by using a form of CDM (where bits are sent in parallel by spreading them using almost-orthogonal sequences).

The receiver sensitivity in the 2.4 GHz bands is specified to be at least $-85$ dBm. For BPSK modulation in the 868/916 MHz bands, the receiver sensitivity should be at least $-92$ dBm or $-85$ dBm with ASK modulation.

## 13.5 THE MAC LAYER IN SENSOR NETWORKS

Like the physical layer, the MAC layer cannot make use of very complicated algorithms since they have to reside in each sensor and sensors are expected to be low-cost devices. In wireless sensor networks the shared medium is air, which makes the design of MAC protocols more challenging due to interference and lack of reliability. This section has to be considered in conjunction with Chapter 5, which discusses the MAC layer in detail. In this section we discuss MAC-related issues for sensor networks, followed by a description of the IEEE 802.15.4 MAC. A good tutorial on IEEE 802.15.4 is the article by Callaway *et al.* [Cal02] in the *IEEE Communications Magazine*. Then we present an overview of MAC protocols developed in the research literature that focus either on energy efficiency or latency. A good survey of MAC protocols for sensor networks is that by Demirkol *et al.* [Dem06], which also points to references to some of the MAC protocols discussed in this section, like sensor MAC (SMAC) and its variants, Sift, and the traffic-adaptive medium access (TRAMA) protocol. The design of MAC protocols typically considers metrics such

as throughput, bandwidth utilization, fairness in providing access to the medium, latency, scalability, and efficiency or control overhead for evaluating the MAC protocol. For sensor networks, the metrics for evaluating a MAC protocol are often different, and we explore these briefly next.

### 13.5.1 Issues in Medium Access for Sensor Networks

Unlike MAC protocols that are designed for fairness or efficiently utilizing the bandwidth, in sensor networks the two issues of importance are energy efficiency and latency.

***Energy.*** Energy efficiency is crucial because of the scale and application environments in which sensors are deployed. Sensors are expected to be of extremely small form factor and several thousands of them may be deployed in a sensor field. The sensor field may not be accessible to human beings; and even if accessible, it may not be practical to replenish batteries in several thousand sensors on a regular basis. For most applications, it is expected that sensor devices will be in sleep mode (as far as communication goes) for a majority of the time and will be awake only periodically or as events occur to transmit information. The low duty cycle of sensors must be exploited by MAC protocols.

Some of the common sources of energy waste in MAC protocols are collisions, unnecessary reception of data, idle listening, and control overhead. Collisions occur when two nodes transmit at the same time and both transmissions (whether intended to a receiver or not) arrive simultaneously at a receiver. The receiver cannot recover the transmission. Energy is wasted for the transmission and for the attempted reception. This is a bigger problem with carrier-sensing-based MACs (see similar discussion in Chapter 9 with respect to WLANs). Suppose all sensor nodes are identical and have identical transmission and reception ranges, as shown in Fig. 13.4. The transmission from sensor A can be heard by

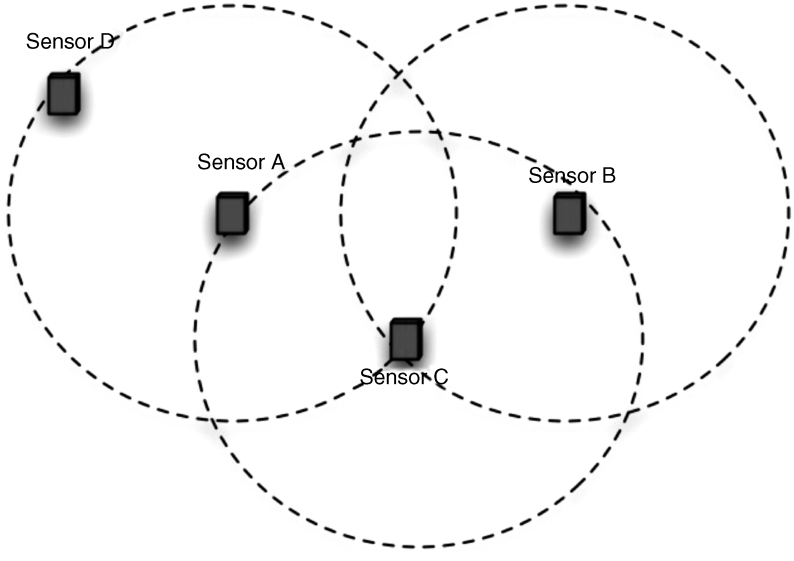


**FIGURE 13.4** Illustrating hidden nodes, collisions, and exposed nodes.

sensor C but not sensor B. So, when sensor A is transmitting a frame to sensor C, sensor B will not sense the channel as busy and sensor A is *hidden* from sensor B. If both sensor A and sensor B transmit frames to sensor C at the same time, then the frames will collide. This problem is called the hidden terminal problem. There is a dual problem called the exposed terminal problem. In this case, sensor A is transmitting a frame to sensor D. This transmission is heard by sensor C, which then backs off. However, sensor C could have transmitted a frame to sensor B and the two transmissions would not interfere or collide. In this case, sensor A is called an *exposed node*. Both hidden and exposed nodes cause a loss of throughput. Since sensor C is listening to sensor A's transmission to sensor D (which is of no consequence to sensor C), it is engaging in wasteful overhearing, which leads to energy waste. In most carrier-sensing-based MAC protocols, there is some amount of idle listening to ensure that the medium is free for access. This wastes energy as well. Finally, in trying to reduce the hidden node problem or in scheduling-based MACs, nodes exchange short control packets to announce their presence, impending transmissions, or schedules. These control packets do not contribute to the application objectives directly and waste energy.

***Latency.*** Latency is an important issue in several applications where the response from a group of sensors needs to be received at the sink (BS or gateway) quickly after the occurrence of an event so that appropriate action can be taken to mitigate the impact of an event. For example, intrusion detection requires alarms to be delivered without significant latency to the BS. If the vibration data in a section of a bridge changes such that it makes it necessary to clear vehicular traffic rapidly and close the bridge, such alarms must be received in a timely manner. In such cases, it is important that the MAC protocol is not a bottleneck for the delivery of data to the sink.

***Data Gathering Tree and Energy/Latency Issues.*** In many sensor applications, sensors deliver data towards a sink in a tree-like manner, as shown in Fig. 13.5. Sensors at the highest level (level 3 in Fig. 13.5) send their collected data to sensors at the next lowest level and so on till the data is delivered at the BS or sink. This causes additional problems in terms of

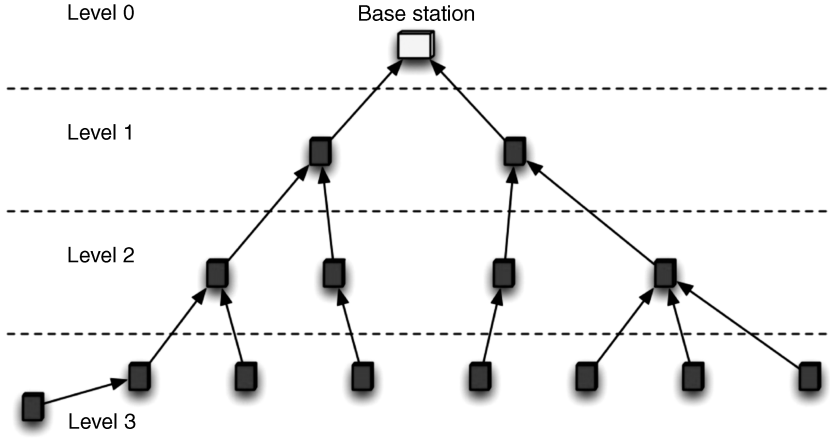


**FIGURE 13.5** Tree-like data delivery.

disproportionate energy consumption in nodes that are closer to the BS (e.g. those at level 1) that are required to transmit not only the data sensed by themselves, but also the data that is sent to them from nodes at higher levels. This also causes congestion closer to the BS, since more data needs to be transmitted there compared with the transmissions at the higher levels.

### 13.5.2 IEEE 802.15.4 Medium Access Control

As mentioned in Chapter 10, the IEEE 802.15.4 standard specifies FFDs and RFDs. An RFD can only communicate with an FFD, whereas an FFD can communicate with RFDs or other FFDs. A network with two or more devices is called a WPAN. A WPAN can be of two topologies: a star topology, where one FFD acts as the WPAN coordinator and controls other RFDs and FFDs in the network, or a peer-to-peer topology, where FFDs and RFDs exist and can communicate with one another as long as they are in range and have the correct functionality (e.g. two RFDs cannot communicate with each other directly). The formation of a network is part of higher layer issues discussed briefly in Section 13.6. However, each WPAN is assumed to have a WPAN coordinator which is an FFD.

The MAC protocol in 802.15.4 operates under a superframe structure (see Fig. 13.6). A superframe is defined as the period between two *beacons*, which are special management packets transmitted by the coordinator. Beacons synchronize the WPANs and provide information about the network. Within the time between two beacons, i.e. the superframe, sensor nodes can have an active period and an inactive period. The active period can be divided into a contention period and a contention-free period. The contention period is slotted. In each slot, nodes use CSMA/CA to access the channel. The process is quite simple. Each device waits for a random period to see whether the channel is idle. If it is idle, then it simply transmits. Otherwise, it backs off for another random period and tries again (see bottom of Fig. 13.6). This access suffers from the disadvantages mentioned previously

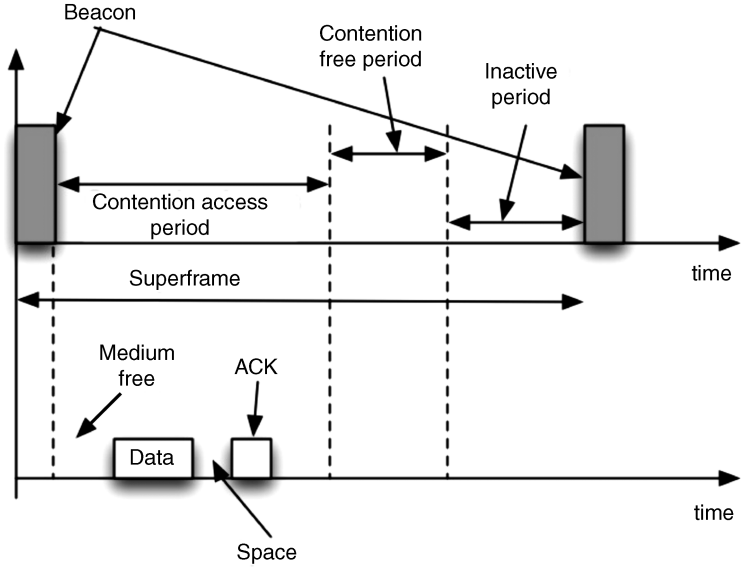


**FIGURE 13.6** Illustration of medium access in IEEE 802.15.4.

with carrier sensing. Scheduled access is possible through the use of the contention-free period, where the WPAN coordinator creates guaranteed time slots, which nodes can use without contention from other nodes. It is possible to have a WPAN without beacons, in which case nonslotted CSMA/CA is adopted by all nodes.

The standard considers "transactions" that are initiated by the low-power devices, which will otherwise have the choice to be in a low-power mode. In the star topology, nodes can send data to a coordinator or receive data from a coordinator. In the former case, the node simply sends data to the coordinator using CSMA/CA and gets an acknowledgment if requested. In the latter case, the node should first request data from the coordinator, get an acknowledgment for its request, followed by the data. Upon receipt of the data, it acknowledges it to the coordinator. Acknowledgements do not wait for the medium to be idle as in the case of data frames. Alternatively, the beacon can have information about whether or not there is pending data for a sensor node (see related discussion of SMAC and its variants next). Peer-to-peer transmissions occur in a similar manner, except that special steps may be necessary for synchronizing transmissions of two peer nodes.

In order to allow the MAC layer to process frames, there must be some time that elapses between successive frame transmissions. The time between receipt of a frame and the transmission of an acknowledgment is the smallest, while long and short IFSs are used to separate long or short frames.

The MAC layer in IEEE 802.15.4 is also responsible for starting and maintaining WPANs (scanning through the various PHY channels, recognizing IDs in the beacons), synchronization with the WPAN coordinator, association and disassociation with a WPAN, allocation of guaranteed time slots, and frame security (encryption and authentication).

We next briefly consider some MAC protocols that have been proposed in the literature, again from a conceptual standpoint. The reader is referred to references previously mentioned in the chapter for additional details.

### 13.5.3 Low-Duty-Cycle Medium Access Controls

As mentioned previously, energy waste in sensor networks occurs due to the fact that there are collisions, unnecessary overhearing, and wasted awake times when sensor nodes are listening to the channel but not receiving anything. This is especially a problem when the sensors actually transmit or receive data very infrequently (low duty cycles). Random access schemes that try to address this problem are described below.

Researchers at the University of Southern California proposed SMAC, which looks at this issue specifically rather than issues such as fair access to the medium (when all nodes get equal opportunities to transmit data). Several variations of SMAC have been proposed since, including *timeout* MAC (TMAC) by researchers in Delft University and dynamic SMAC by the University of Buffalo. The underlying mechanism behind SMAC and its variants is still CSMA/CA. The idea behind SMAC is that sensor nodes can form virtual clusters where they are loosely synchronized. This synchronization can help them coordinate their sleep cycles so that they do not have to be awake when they are unlikely to receive any transmissions. Note that this is a distributed approach compared with the approach taken in IEEE 802.15.4, which requires nodes to ask for data from the coordinator or receive information about pending data in the beacons. Nodes that are in the vicinity of two virtual clusters may have to follow two different sleep schedules in SMAC. SMAC also supports message passing, where a sensor node occupies the medium

until it has completed transmitting a message (which may be fragmented into frames). This may make the medium unfair, but avoids the unnecessary waiting times in typical CSMA protocols. Moreover, SMAC supports the use of RTS and CTS control frames that ameliorate the hidden node problem by announcing the impending data transmissions in the vicinity of *both* the sender and the receiver nodes, like the IEEE 802.11 protocol. The RTS/CTS frames also indicate the size of transmission, allowing nodes to sleep longer if necessary. TMAC introduces a variable sleep cycle for nodes, compared with SMAC, to improve upon the energy consumption in the presence of variable loads. Latency, which could be high in SMAC, especially for multi-hop transmissions (e.g. in Fig. 13.5, the time taken for a packet to reach the sink may be very high if the sleep schedules of the nodes in different levels are arranged in a worst-case manner), can be reduced by halving the sleep schedule of some nodes which experience high latency and yet maintain synchronization with other nodes.

Asynchronous MAC protocols for low-duty-cycle sensor nodes rely on a mechanism called "low-power listening," where sensor nodes sample a (periodically) transmitted preamble to see whether there is data that is intended for them. If so, the nodes wake up; else they return to sleep. The low-power listening reduces the power to detect intended transmissions; but there are other disadvantages, where nodes should stay awake to listen to the preamble whether or not the preamble indicates a transmission towards them, and latency is a problem, as data transfer can occur only after the preamble is completed even if a receiving node is awake at the beginning of the preamble. As in the case of SMAC, latency accumulates over multiple hops. The Berkeley MAC from the University of California at Berkeley and X-MAC from the University of Colorado are examples of asynchronous MAC protocols that rely on low-power listening. The X-MAC protocol employs a shorter preamble to reduce both energy consumption and latency.

### 13.5.4 Low-Latency Medium Access Controls

A second group of MAC protocols for sensor networks considers latency as the issue of interest.

Of these, so-called *event-driven* MACs are still based on carrier sensing and random access. When an event occurs, it is likely that sensors that detect the event will start a flurry of transmissions, many of them spatially correlated, that attempt to deliver data towards a sink, all at times very close to one another. In many applications, it is not necessary that all of this data be delivered reliably and correctly to the sink. It is sufficient if a subset of sensors from a spatially correlated set delivers data reliably and correctly to the sink. This observation is exploited in *Sift*, a MAC protocol developed at the Massachusetts Institute of Technology. Again, fairness in channel access is a sacrifice for improvement in latency performance. In typical CSMA/CA MAC protocols, it is common for nodes to pick a slot within a contention window to transmit. The slot is picked randomly, from a uniform distribution, so that there is roughly fair access for all nodes. The node with the earliest slot gets to transmit, while others back off and wait for their chance to transmit in succession. The contention window expands with time in the case of collisions to alleviate the problem. In Sift, the probability of picking a slot is picked from a nonuniform distribution within a fixed contention window, which changes depending on whether or not a node observes a transmission in the earlier time slots. If no transmissions are seen, then a node increases the probability with which it will transmit in the coming time slots. Sift has a much better latency performance than standard IEEE 802.11-based CSMA/CA.

Finally, we mention TDMA-like MACs that attempt to schedule transmissions of sensors to prevent collisions and achieve gains in latency. The TRAMA protocol, which is itself a modification of node activation multiple access (NAMA) to better suit sensor networks, considers a two-hop neighborhood of sensor nodes to schedule transmissions. The schedules ensure that there are no collisions between transmissions (which waste energy). The schedules also indicate the intended receivers, allowing other nodes to go to sleep when no packets are expected for them. In the *neighbor protocol* phase, sensors exchange information about their two-hop neighborhood. In the *schedule exchange protocol* phase, nodes exchange traffic information and schedules. Then, an *adaptive election algorithm* picks the transmitters and receivers to ensure collision-free transmissions. The first two phases make use of a contention access period (like CSMA) to enable exchange of information.

## 13.6 HIGHER LAYER ISSUES IN SENSOR NETWORKS

In previous sections we provided an overview of sensor devices and the lowest two layers (namely the PHY and MAC layers) of sensor networks. However, sensor networks are *application-oriented networks* and the operation of the higher layers plays an important role in realizing application objectives. For a sensor network to be operational, it is necessary for nodes to self-organize into a network to establish a service and have protocols in place that enable routing of sensed and processed data from source sensor nodes to destinations (sinks). Instead of transmitting all of the sensed data towards the sink, sensor nodes can save communication and energy costs by performing in-network processing and aggregation of data. How sensor nodes should be deployed to *cover* a region that needs to be sensed such that areas do not suffer gaps nor the resulting deployment cause partitioned networks is a challenging issue. Many applications need sensors to have some ability to determine their location (also called localization – see Chapter 12 for more details). Several MAC and routing protocols, as well as applications, need synchronous operation of sensor networks, requiring some protocols for enabling synchronization between nodes and handling clock drifts. Finally, it is necessary to keep sensors and their information secure.

A large number of research papers have been published over the last decade addressing these issues and, in some cases, the impact that these issues have on one another. For example, some MAC protocols, not discussed in this chapter, consider joint MAC and routing. As another example, there are protocols that look at routing correlated data towards a sink. It is extremely difficult to survey all of these papers in a short chapter devoted towards background information. Thus, the objective of this section is to provide the reader with an idea of some of the issues related to higher layers in sensor networks and to provide information about some common protocols and research efforts briefly. The references described next provide pointers to many surveys and tutorials that address these topics in greater detail.

The chapter on sensor networks in Siva Rama Murthy and Manoj [Srm04] discusses many of the topics considered in this chapter, such as routing, establishment of the network, and coverage. Recent developments in node clustering for sensor networks are described in Younis *et al.* [You06]. A treatment of Zigbee is presented by Wheeler [Whe07], including random addressing and routing. The paper by Woo *et al.* [Woo03] describes MintRoute, the routing protocol used in many real implementations of sensor networks (such as the structural health monitoring network described by Kim *et al.* [Kim06]). Al-Karaki and

Kamal [Alk04] provide a comprehensive survey of routing protocols for sensor networks. A survey of coverage and connectivity issues in wireless sensor networks is available in Ghosh and Das [Gho08], and a thorough treatment of topology control is presented by Santi [San05]. Our treatment of the issues of coverage is heavily influenced by the latter two papers. A survey of synchronization protocols for sensor networks is available in Sivrikaya and Yener [Siv04]. The issue of localization in ad hoc/sensor networks has been discussed in many papers [e.g. Sav01, Bul00, Nic03]. Stinson [Sti03] is a good reference on encryption/authentication schemes and the AES. Details of the security process in IEEE 802.15.4 are available in [IEEE06].

### 13.6.1 Establishing the Sensor Network

When sensor nodes are first deployed in a sensor field, some type of organization is essential to ensure that the network operates smoothly. In this section we provide a quick overview of some of the approaches implemented or proposed for sensor networks.

In the case of the IEEE 802.15.4 standard (along with the ZigBee higher layer protocols), this is accomplished through the use of WPAN coordinators. As mentioned in Chapter 10, two different topologies are possible. In the star topology, nodes communicate directly to a WPAN coordinator and all communications go through the coordinator. In many ways, this is like nodes communicating to an AP in WLANs. In the peer-to-peer topology, devices can communicate directly with one another as long as they are in radio range.

In the star topology, as shown in Fig. 13.7, an FFD (see Chapter 10 for information on FFDs and RFDs), when deployed, becomes the WPAN coordinator. This is accomplished simply by the device, by transmitting a beacon and announcing itself as the coordinator if it does not hear any other device when it powers up. In the peer-to-peer topology, a similar mechanism works and there is a WPAN coordinator (usually the first FFD to power up). However, devices may communicate directly with one another where allowed. If two or

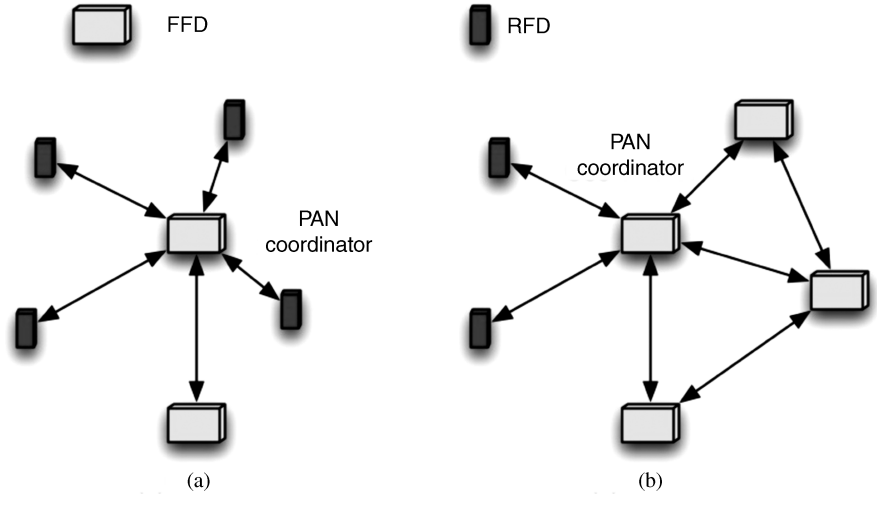


**FIGURE 13.7** Star and peer-to-peer topologies in IEEE 802.15.4.

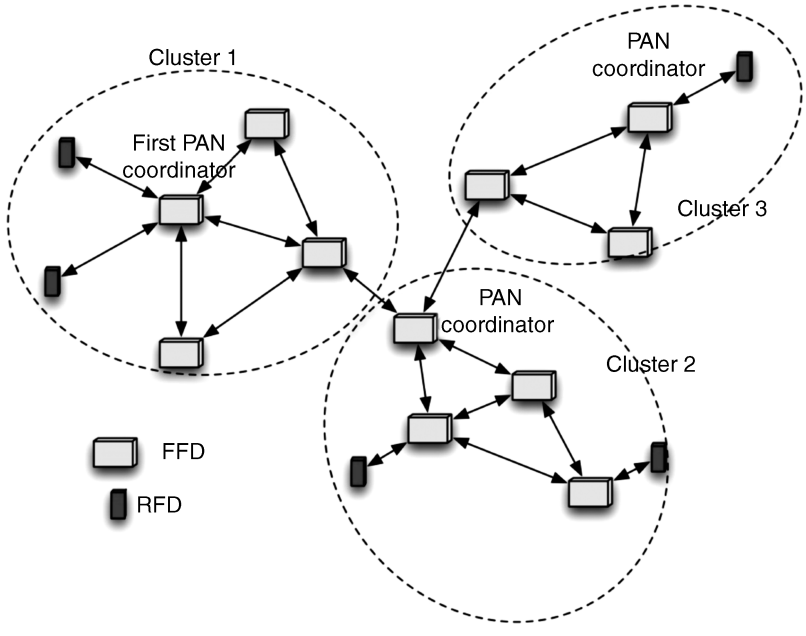


**FIGURE 13.8** Cluster tree formation in IEEE 802.15.4 WPANs.

more FFDs attempt to become coordinators, then some contention resolution beyond the scope of the IEEE 802.15.4 standard becomes necessary.

A *cluster tree* is a generalized form of the peer-to-peer topology in IEEE 802.15.4 networks. Figure 13.8 shows an example of a cluster tree with three clusters. Here, the assumption is that most of the devices are FFDs (although RFDs can connect to clusters as leaf nodes). The first FFD that announces itself (or a device with more power or capabilities) becomes the overall WPAN coordinator. It transmits beacon frames and provides other nodes (and other coordinators) synchronization. As the network grows, the overall WPAN coordinator instructs another device to become a WPAN coordinator of its own cluster. Clusters can develop in this way into a large network. The standard specifies resolution of conflicts between WPAN IDs, transmitting beacons, etc.

Clustering is an approach that is also suggested in the research literature for enabling communications between sensors in a field and the BS. Groups of sensors create clusters with cluster heads. These cluster heads are responsible for communicating information from sensor nodes in their clusters to the BS, perhaps through other cluster heads. While the proposed approaches in the research literature are not very specific as to the functionality of the cluster heads (e.g. they are not like WPAN coordinators), in order to distribute the load across sensors in the clusters, cluster heads are periodically changed by an election process. This improves the number of nodes in the network that are still alive after a given time. One popular protocol that changes cluster heads periodically is low-energy adaptive clustering hierarchy (LEACH; see Section 13.6.2). Several other clustering approaches have been suggested (e.g. picking nodes with highest degree as cluster heads to improve connectivity, or picking them among nodes that provide redundancy of coverage).

A second approach to initialization and establishment of the network is a flat architecture consisting of levels (see Fig. 13.5, for example). A BS could broadcast a signal containing its ID which is received by all sensors in the field. This is because the BS is powerful and can transmit at a high power. Next, all sensors respond to this broadcast with their IDs. The BS only receives the responses from nodes that are in level 1 (those that can directly communicate with the BS using a single hop). The BS broadcasts a list of these nodes, enabling nodes that are in level 1 to recognize their level. Next, nodes that are not in level 1 transmit their IDs. Level 1 nodes that receive these IDs forward them to the BS, which broadcasts a list of the new nodes as belonging to level 2, and so on. This way, nodes can determine where they belong and how many hops from the BS they are away.

One of the issues with sensor networks is the challenge in providing addresses to sensor nodes, as they may be several hundred or thousands in number, and careful deployment of addresses in nodes is quite challenging. The ZigBee Pro higher layer standard that operates on the IEEE 802.15.4 lower layers handles this issue by allowing new nodes that join a network to randomly pick 16-bit addresses. With more than 60 000 addresses, the expectation is that collisions will be negligible.

### 13.6.2 Routing

Routing in sensor networks has received special attention in the research literature. The number of routing protocols that have been proposed makes it impossible for a survey in a short overview chapter like this one. Moreover, many of these routing protocols are not actually implemented in real sensor networks as of now making them mostly of academic interest. The objective of this section is to provide a short summary with pointers for further reading.

Routing in ad hoc networks is typically classified into proactive and reactive types. In proactive routing protocols, nodes exchange information periodically and maintain routes to all possible destinations in the network (i.e., they have information about the network topology). In reactive or on-demand routing, nodes only try to find routes to those destinations to which they are interested in transmitting data. This may be accomplished just before data has to be sent by sending route requests to neighboring nodes that forward these on till the destination is reached. In sensor networks, often times, the data is directed towards a sink from many source sensor nodes. On occasion, traffic may be directed between sensor nodes in the network. The situation is exacerbated by constraints on energy, limited memory of nodes, and potential for links to break due to nodes dying or sleeping.

In the ZigBee Pro protocol, multiple routing algorithms are employed depending on the type of traffic that is involved. For traffic between any two nodes in the network, a version of the AODV protocol is employed. This is a reactive protocol that is suitable for occasional traffic between two nodes, especially if they are close to one another in the network topology. A proactive approach is selected for data from many sensor nodes to the BS or gateway. The BS periodically broadcasts its presence to nodes that are one hop away and these nodes can inform their neighbors, and so on. When sensors send data to the BS, packets carry the source routes (i.e. the nodes through which they have reached the BS). The BS thus does not need to store routes to thousands of devices. Instead, it can simply send a response back to a sensor node using the source route embedded in the received packet. Using source

routes increases the overhead in packets, but this has been used as a compromise between complexity and overhead.

One of the early routing protocols developed for sensor networks is MintRoute, which is proactive and maintains information about the next hop that can take a packet towards the BS. It recognizes the fact that, in a dense network, there will be a few good links to one-hop neighbors and many weak links to other nodes and develops appropriate metrics to account for the best route towards the BS. We note that neither of these two routing schemes considers energy efficiency or optimizations of any kind for sensor networks.

Several routing schemes have been defined with the recognition that sensor networks are data oriented – in that sensors have to sense and collect data for delivery to a sink. It is likely that such data is spatially correlated and it may be possible to reduce the energy consumption by eliminating redundancies through processing of the data en route to the BS. In the *sensor protocol for information via negotiation* (SPIN), sensor nodes perform some metadata negotiation before transmitting the data towards the sink. The nodes can also take into account the remaining energy they have to decide which node transmits the data. *Directed diffusion* aggregates data along a route towards the BS. The BS queries sensor nodes with *interests* (e.g. data from nodes that see a particular range of values). The interest is transmitted through the network by sensor nodes. Depending on the interest, a gradient can be set up for different flows from leaf nodes to the BS. The flows with the strongest gradients are reinforced to prevent flooding. The protocol has been evaluated using random sources in the sensor field and a circular *event region*, which generates the event of interest. Both SPIN and directed diffusion save energy because of elimination of redundancies. In both cases, aggregation occurs only when routes intersect (or when nodes are close together). Among other routing protocols of interest is the COUGAR data-centric protocol that assumes that the entire network is a huge distributed database and abstracts query processing from networking functions to identify nodes that need to deliver data. In-network data aggregation can provide further savings.

The routing protocols considered so far are flat routing protocols that assume that all sensors are involved in routing in the same manner. As described previously, one of the methods for establishing a sensor network is to create clusters with cluster heads. This can make routing more scalable by introducing a hierarchy into the network architecture. Several routing schemes have been specified with clustering in view. The earliest, called LEACH, also lets cluster heads aggregate data. The assumption that cluster heads can directly send data to the BS is a weakness of LEACH. In the *threshold-sensitive energy-efficient sensor network* (TEEN) protocol, cluster heads inform nodes in their cluster the hard and soft thresholds for the sensed quantity, which indicate the interest in the data. This reduces unnecessary transmissions to the cluster heads. TEEN supports changing cluster heads and thresholds as necessary.

Location-based routing protocols have received attention in the literature. These protocols assume that sensor nodes are aware of their locations (see Chapter 12) and can be addressed accordingly using their location information. Location awareness can be combined with a hierarchy. For example, in the *geographic adaptive fidelity* protocol, nodes form virtual grids based on their location. They elect one node in the grid to be awake while the others go to sleep. The node that is awake will be responsible for sensing and communicating.

Other routing protocols consider maximizing battery life, using multiple paths for routing, QoS-based routing that balances data quality and energy consumption, mobility, and multiple BSs.

### 13.6.3 Coverage, Connectivity, and Topology Control

As mentioned in the introduction, one of the key issues in sensor network deployment is the interaction between coverage, connectivity, and topology control. Loosely speaking, coverage implies that sensors have been deployed so that they do not miss sensing events of interest in the geographical area deployed. Assuming that the area is "covered," sensors must now be able to communicate such that there are always paths available to report the sensed data to the BS or sink (connectivity). Topology control seeks to optimize the "connectivity" of the network by changing the transmission range of nodes while reducing energy consumption. This may have the by-product of reducing contention and collisions in the network. In this section, we provide a brief overview of the concepts related to coverage, connectivity, and topology control without going into the details of algorithms or approaches investigated in the research literature. Much of this work is still in the theoretical domain, with few actual applications employing the results from this research in real sensor networks.

We start with the observation that the coverage required by different applications may be different. The amount of coverage expected to monitor habitats or the temperature in an area may be quite different from that required to detect intrusions in a given area. Even the definitions of coverage can be quite different. *Blanket coverage* assumes that it is possible to deploy nodes in a static arrangement such that there is a maximum detection rate of events in the sensor field, while *barrier coverage* minimizes the chance that an event goes undetected. *Sweep coverage* envisions moving sensor devices in the sensor field so as to balance the detection rate and missed detections in a given unit area.

It is generally believed that the quality of sensing by a sensor device degrades with distance from the sensor, in a manner similar to how the signal strength drops with distance. In fact, this degradation is modeled in a similar manner: the sensitivity of the sensor drops as the $m$th power of the Euclidean distance between the sensor and the target point to be sensed. Like the unit disk model for communications (see Section 13.4), a threshold determines when the sensing is essentially nonexistent. A circle with a sensing radius $R_s$ thus defines the coverage of a given sensor. Like the gray zone, this model can be enhanced to have a zone where the sensing coverage is probabilistic, at closer distances the sensing is perfect, and at farther distances there is no sensing (see Fig. 13.9). We note that specific models for specific types of sensor (e.g. temperature or seismic) do not appear to be readily available.

While it is not trivial to quantify the coverage provided by a particular sensor deployment, in order to obtain an idea of how good or how poor the coverage is, *paths* within the sensor field with certain coverage are employed. Let us suppose that sensors are deployed (say randomly) in a field. Each sensor has the ability to sense an event, which drops with distance from the sensor. Along a given path through the field, the sensitivity to detect an event changes.

A path that results in the least probability of detection represents the worst-case scenario. Two optimization problems that determine the *minimal exposure path* and the *maximal breach path* have been defined to quantify the worst-case path. In the former case, exposure is defined as the path integral of a sensing function (which depends on the distance from the

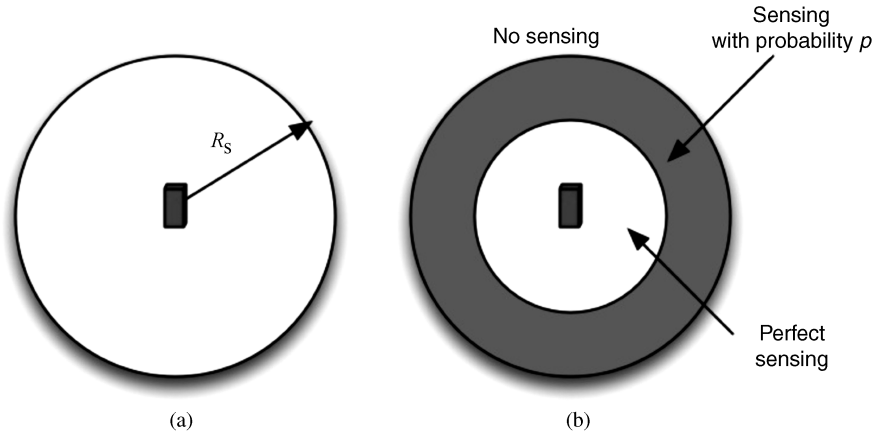


**FIGURE 13.9** (*a*) Unit disk sensing and (*b*) sensing with a gray zone.

closest sensor or distances from a group of sensors that contribute a certain sensitivity to a given point) during a specific time interval. The path that results in minimal exposure represents one metric of the worst-case coverage in the sensor network. The maximal breach path uses a simpler idea. It corresponds to the path in the sensor field where each point on the path is at the maximum possible distance from the closest sensor. The maximal breach path can be constructed as follows. First draw line segments joining pairs of sensors in the deployed sensor field. Then draw lines that bisect these line segments and use the intersections of the bisectors as vertices and the bisectors as edges of polygons that tessellate the sensor field. This creates a Voronoi diagram where all points inside the polygons are closer to some sensor compared with points along the edges. The maximal breach path must lie along the edges of the polygons, but the amount of breach depends on how far a given edge is from the sensors within the polygon.

In a similar manner, it is possible to define paths that have the best coverage, and they are referred to in the literature as the *maximal exposure paths* and *maximal support paths.* The former corresponds to a path that has the largest exposure path integral in a manner similar to the minimal exposure path. The maximal support path corresponds to a path that has points closest to sensors on it.

Research literature evaluating the possibility of *moving* sensors to obtain the best coverage after initial deployment also exists. If mobile robots are employed, a gradient to repel and attract sensors has been proposed to distribute sensors in a field. The Voronoi diagram can also be used to determine coverage holes and fill them by repelling sensors that are close to one another or to the edge of a sensor field.

One question that arises is whether coverage automatically implies connectivity in the network. Connectivity is often measured by whether or not the graph of communication links that exist between pairs of nodes is connected or partitioned. A result from the research literature indicates that, given a communication radius that is two times the sensing radius (assuming unit disk models for both sensing coverage and communications), the network is connected if it also provides complete coverage over the given area. Coverage here assumes that at least one sensor can sense any point in the given region. Similar results exist for situations where $n$ sensors are required to cover any point in the given region.

Topology control ensures connectivity of the network by changing the transmission range of nodes (by increasing the transmit power). While this is not practical in the real sensor networks of today, one can anticipate benefits of this approach in the future. Two types of topology control technique are considered: homogeneous and nonhomogeneous. In the homogeneous case, all sensors must have the same transmit and receive ranges, and the question is what range is the smallest to provide connectivity in the network. In the nonhomogeneous case, sensors are allowed to have different transmission ranges (which may result in some asymmetric links).

Related to the idea of topology control is the idea of developing optimal sleep schedules. In many applications, many more sensor nodes are deployed than necessary to cover the region and keep the network connected. In such cases, it is not necessary to keep all the sensors active all the time. Developing optimal sleep schedules, where a subset of nodes sleeps while others are active, such that the network is still covered and connected, is another problem that has been considered in the research literature.

### 13.6.4 Synchronization

Synchronization is an important function in sensor networks. As already described in Section 13.5, many MAC protocols rely on synchronization between nodes. Synchronization is also necessary for localization schemes that rely on time of flight of a signal. When nodes perform data aggregation, it is necessary for them to be synchronized to recognize the age of data. This is also necessary for some applications at the BS, which needs temporal information related to sensed data to make decisions. Synchronization can be used to save energy in the network by having correct sleep schedules and transmissions.

It is not easy to synchronize all sensor nodes to a common clock for several reasons. The devices are supposed to be inexpensive and may not have accurate clocks. The clock drifts between different sensors can be high. There are several sources of error in computing a common time, such as the time at which the synchronization message is composed may be different from the time at which it was sent due to medium access delays, the propagation delay is unpredictable, and the time required for a receiving node to process the synchronization message may be hard to determine. The challenges only increase in multi-hop networks.

In sensor networks, one can think of local synchronization and global time synchronization. In the case of local synchronization, the argument is that nodes should run without any synchronization for most of the time. When sensed data need to be transmitted, nodes can temporarily synchronize in a relative manner, perhaps to the first node that initiates synchronization. This makes sense in low-duty-cycle networks where communications is infrequent and the importance of aggregation and ordering is mostly in local groups of sensors. Implementations of local synchronization through the reference broadcast synchronization (RBS) protocol have shown synchronization on the order of a few microseconds on mote-class sensor devices. Global synchronization schemes have also been proposed for sensor networks. For example, the timing-sync protocol for sensor networks (TPSN) uses a root node and levels (similar to Fig. 13.5) for achieving network-wide synchronization. The way it operates is as follows. The root node sends a level discovery message. Its immediate neighbors form level 1. Their immediate neighbors that cannot reach the root form level 2, and so on. Then a two-way message exchange occurs between level 1 nodes and the root node with time stamps of local times of transmission and

reception. These allow the clock drift and propagation delay to be computed by level 1 nodes, which can then synchronize themselves to the root node. Level 2 nodes then synchronize themselves with level 1 nodes, and so on.

In IEEE 802.15.4, the beacon from the WPAN coordinator can be used for synchronization purposes.

### 13.6.5 Security

Security in sensor networks is an important topic, although there are opinions that suggest that work in this area may be of academic interest only. In this section, we briefly discuss the security features provided in the IEEE 802.15.4 standard and briefly mention some of the research work describing security threats in sensor networks.

***Security Threats.*** Sensor networks are especially vulnerable to security attacks for the following reasons: (a) the transmission medium is air, and so it is easy to obtain remote access to the medium using powerful antennas for eavesdropping, jamming, or injecting malicious traffic; (b) sensors may be deployed in huge numbers and are supposed to be of low cost, with the implication that the possibility of some of the sensors being captured, tampered with, and compromised being very real. It is possible for adversaries to deploy their own sensors into a sensor field, but this requires physical protection of the sensed region. There are numerous security threats that have been considered in the research literature of wireless sensor networks, making it difficult to consider them all together. Instead, it is easier to group the threats into categories. While there are overlaps between them, we can classify these threats into the following categories.

*Physical Layer Threats:* At the physical layer of the communications protocol stack, common threats against sensor networks are disruptions to communications through jamming and node disabling. By jamming, an attacker may disrupt reliable communications by transmitting signals that interfere with the radio signals of sensor nodes. This may result in partitioning of the network, lower reliability of the sensed data because of the lack of availability of data from certain sensed regions, and ultimately result in the battery exhaustion of nodes repeatedly transmitting data till they are acknowledged or receiving bogus data. Jammers can be classified as those that may be outsiders employing a constant radio signal, deceptive jammers that inject regular packets into the network, random jammers that alternate between sleep and awake states, and reactive jammers that cleverly disrupt communications upon sensing channel activity. Experimental studies indicate that packet delivery ratios are adversely impacted by all of these types of jammer. Jamming may adversely impact sensor nodes at the edges of a network or those that are towards vulnerable physical areas.

*Eavesdropping Threats:* One of the most common threats in wireless sensor networks is information leakage, where an adversary may obtain the sensed information by simply passively eavesdropping on the radio signals being transmitted by sensor nodes. Eavesdropping is especially problematic even with encryption, because of the potential for sensor nodes possessing keys to be compromised or captured by adversaries. One model for computing the eavesdropping vulnerability is based on the adversary interested in

predicting the behavior or aggregate output of the sensor network. In addition to information leakage, radio transmissions may reveal the location of sensors and the sink node and allow other kinds of analyses on the traffic patterns. An adversary may also be able to poll sensor nodes actively for information if there is no authentication of queries in the network.

*Threats Impacting Routing:* In networks, it is important for nodes to know where to send data packets so that they reach the destination in an efficient way. Such routing protocols in sensor networks are still evolving, since attempts to directly use routing protocols designed for mobile ad hoc networks in sensor networks have faced challenges due to the scalability and energy requirements of sensor networks. Geographical and geometric routing that makes use of the knowledge of the Euclidean coordinates of sensors is proposed as an efficient means of routing data to the destination. However, it is likely that, in general, routing in sensor networks faces the same threats as those in mobile ad hoc networks. Such threats include location disclosure, replay of old routing information, disruption by fabricating routing information, and route table poisoning. In addition, wormhole, blackhole, and Sybil attacks are possible. In blackhole attacks, malicious nodes advertise themselves as closer to the destination, thereby making themselves part of most routes. They can then disrupt network operation by dropping packets or get information by eavesdropping. In Sybil attacks, a single malicious node claims to be more than one node. This way, it could claim a disproportionate amount of resources and also perform blackhole attacks. If there are collaborating nodes, then they may create a wormhole (a tunnel) between them and create the impression of a false network topology.

*Threats Impacting Position Information:* The position of a sensor node has importance in several applications. For example, temperature variations over a given area may have to be accurately characterized, in which case the position location of the sensor reporting the temperature reading needs to be known to a certain accuracy. Such position information may be used for routing or even in security measures. Further, the location of a sensor monitoring a critical quantity may itself need to be kept secure (location privacy). Malicious nodes can interfere with the reporting of position location information in many ways. They can fabricate the position information or interfere with the support infrastructure used by sensors to determine their own positions. In the latter case, there are many different approaches for determining the position of sensor nodes, such as using beacons from nodes at known positions, determining the number of hops a node is away from a reference node, and so on. Malicious nodes can interfere with such position-determining activity.

*Threats Impacting Data Aggregation and in-Network Processing:* Data aggregation and in-network processing is an important feature of data-intensive sensor networks. Because sensor nodes collect a huge amount of data and sometimes only aggregate information is necessary at the sink (e.g. average value or the sum of the sensed quantity), intermediate nodes can process the received data (in-network processing) or fuse data and forward those values. This reduces the communication costs and delays in the network. However, such functionality makes it extremely easy for malicious nodes to introduce false values that corrupt the processed or fused values. If a malicious node is responsible for fusing or

aggregating data, then the problem could be worse. If a Sybil attack is launched, then a node can claim multiple identities and further skew the aggregated data by creating multiple false reports.

*Threats Against Time Synchronization:* Sensor networks often require nodes in the network to be time synchronized for many reasons, such as data fusion, scheduled transmissions for saving power, tracking duplicate sensed data, and so on. Time synchronization can be achieved using reference broadcasts or sender–receiver synchronization. It is possible to disrupt the sensor network operation by misleading different nodes about the time at which they have to perform operations like sensing or transmissions of packets.

*Miscellaneous Threats:* If sensor nodes are compromised, then they can disrupt a sensor network in many ways. For instance, a compromised node may not follow the medium access protocol and hog the medium. If a node assumes several identities, as in the case of a Sybil attack, it could access the medium more often than it should normally have fair access. These may both deny access to radio resources by legitimate sensor nodes. In many types of sensor network, a sink node is used to collect data after a query to many sensor nodes in the networks and for other types of network maintenance. In some cases, mobile sink nodes are employed to poll sensors or collect data from a set of static sinks. Compromise of sink nodes can lead to damages or disruption of a sensor network.

***Security in IEEE 802.15.4.*** In IEEE 802.15.4, no attempt is made to address the many different vulnerabilities or threats that exist for the sensor network applications described above. However, cryptographic protection of communications over links is part of the standard. The cryptographic operations that are part of the standard provide for confidentiality, integrity, and authentication of the communicated data using encryption and message authentication codes. The standard assumes that secrets that are to be shared between sensor nodes is an issue that is beyond its scope. So it is necessary to employ additional key establishment and key management schemes with IEEE 802.15.4 sensor devices. Keys may be pairwise or shared by a group of nodes. The rest of this section assumes that, somehow, pairs of communicating nodes share keys with each other (group or pairwise).

Protection in IEEE 802.15.4 can be adopted on a per frame basis with message authentication (includes integrity and replay protection) and optional encryption of contents for confidentiality. This enables deploying security as needed without expending energy for cryptographic operations that are not necessary. The size of the message authentication code can be varied (32, 64, and 128 bits), offering various levels of protection. Similarly, encryption may or may not be enabled. It is also possible to send frames without any protection.

The most general form of protection in IEEE 802.15.4 involves the *c*ounter mode with *c*ipher-block-chaining *m*essage authentication code (CCM) operation of a block cipher. This operation is used in the IEEE 802.11i standard for wireless local area networks as well. The block cipher specified in the IEEE 802.15.4 standard is the AES, an encryption scheme that was standardized by NIST in 2001. This scheme works as follows. A counter is incremented and encrypted with an encryption key. The resulting output stream is XOR-ed with

the data to provide confidentiality. The data is broken into bocks of 128 bits. Each block is XOR-ed with the previous block's ciphertext and then encrypted using an authentication key. The first block is XOR-ed with an initial vector. The final encrypted block is truncated to the appropriate number of bits to form the message authentication code. A receiving node can locally perform the same operations to decrypt the data or to compare the received message authentication code with the locally computed message authentication code to see if the message has been modified or fabricated.

## QUESTIONS

1. What is habitat monitoring? Why are sensor networks suited for this application?
2. Name two applications other than habitat monitoring that sensor networks are used for.
3. What are the four classes of sensor devices?
4. What are the differences between a base station class sensor device and a mote class sensor device?
5. What challenges in size reduction and mobility are expected in the future for sensor devices?
6. Why is it beneficial to employ unlicensed spectrum for sensor networks? What are the disadvantages?
7. What is the unit disk model for communications in sensor networks? Why is this model not correct?
8. Explain the concept of the grey zone for links between sensor devices.
9. What is the difference between a channel page and a channel number in IEEE 802.15.4?
10. What issues are more important for sensor networks at the MAC layer than traditional wired networks?
11. Why are sensors closer to the base station more likely to have depleted batteries than those farther away from the base station?
12. What is the difference between a full function device and a reduced function device in IEEE 802.15.4?
13. What is the primary objective of low duty cycle MACs?
14. What techniques are suggested to reduce energy waste in accessing the medium sor sensor networks?
15. What are event-driven medium access control protocols?
16. Differentiate between the star topology and the cluster-tree topology in IEEE 802.15.4 based sensor networks.
17. Name three types of routing protocols for sensor networks. What objectives do they try to meet?
18. Differentiate between blanket coverage and barrier coverage in sensor networks.
19. What is a minimal exposure path?
20. Why is synchronization important in sensor networks?
21. What is a blackhole attack?
22. What security services are provided by IEEE 802.15.4? Which of them are optional?
23. Describe the most general form of data protection in IEEE 802.15.4.