

Chapter 1

Applications, Models, Problems, and Solution Strategies

Hai Liu¹, Amiya Nayak², and Ivan Stojmenovic²

¹*Hong Kong Baptist University, Hong Kong, P.R. China*

²*School of Information Technology and Engineering, University of Ottawa,
Ottawa, Ontario, Canada K1N 6N5*

Abstract

This introductory chapter describes various applications, scenarios, and models of wireless sensor and actuator networks. Problems at the physical, medium access, network, and transport layers as well as various tools needed to enable their functioning are identified. Various assumptions and metrics used in simulations and protocol descriptions are discussed. The chapter then describes ways of generating sensor and actuator networks based on widely accepted unit disk graph models. Finally, this chapter discusses solution approaches arising in sensor networks and advocates the use of localized protocols, where individual sensors and actuators make their decisions based on local knowledge.

1.1 WIRELESS SENSORS

We will elaborate first on wireless sensors; then on wireless sensor networks (WSNs) (with a single sink), their properties, models, and application types. Afterwards, we will add actuators to the model and discuss various combinations of sensors and actuators that can form a heterogeneous network with different levels of complexity.

Recent technological advances have enabled the development of small-sized (a few cubic centimeters), low-cost, low-power, and multifunctional sensor devices. There are different types of sensors. Sensors are normally specialized, but sometimes a few capabilities may be available in a single sensor. They may measure distance, direction, speed, humidity, wind speed, soil makeup, temperature, chemical composition, light, vibration, motion, seismic activity, acoustic properties, strain, torque, load, pressure, and so on.

Traditionally, sensors are attached to the environment and their measurements are sent to a base station (BS) with wired communication. There exists a large body of knowledge on such models and applications of sensors which have been studied by a huge community of researchers. During the last decade, a new vision of sensor nodes as autonomous devices with integrated sensing, processing, and communication capabilities has emerged. Attaching antenna for receiving signals and a transmitter enables wireless communication of sensors. Sensors also have a small processor and a small memory for coding and decoding signals, as well as for running simple communication protocols. They differ in their battery capacity; for example, some of them run on small batteries and last a day, whereas others have larger batteries attached that let them last up to a month with continuous operation. In some applications, a renewable power supply such as a solar panel is used. Further, some sensors are embedded into other devices and draw their required energy from them. Such sensors do not have energy limitations in their functioning. In some scenarios, sensors could be provided with a wireless single-hop access to infrastructure networks such as the Internet.

For some applications, sensors may be of a large size, especially if they are protected by boxes or lifted to a height that improves their communication and protection level. When collected data is not time critical, sensors may function in isolation. For example, seismological data or bird presence detected acoustically can simply be collected in the local sensor memory and downloaded when visited by humans. This book concentrates, however, on scenarios involving networks of wireless sensors.

1.2 SINGLE-HOP WIRELESS SENSOR NETWORKS

The majority of the existing applications for “wireless” sensors rely on a single-hop wireless network to reach a BS for further processing of the measured phenomena. That is, sensor measurements are sent directly, using a wireless medium, from sensor to BS. Most of these applications rely on sensors that are *embedded* into a different device. Also, the majority of applications for embedded sensors rely on single-hop wireless communication. For example, small sensors can be embedded into a traffic surveillance system to monitor traffic on congested roads or be used to monitor hot spots in a region or building.

Health care is one of the primary applications for wireless networks composed of embedded sensors. Sensors can be embedded into watches which, when attached to patients, monitor and analyze data such as pulse and blood pressure. In case of potential health risks, individual sensors send alarm messages to

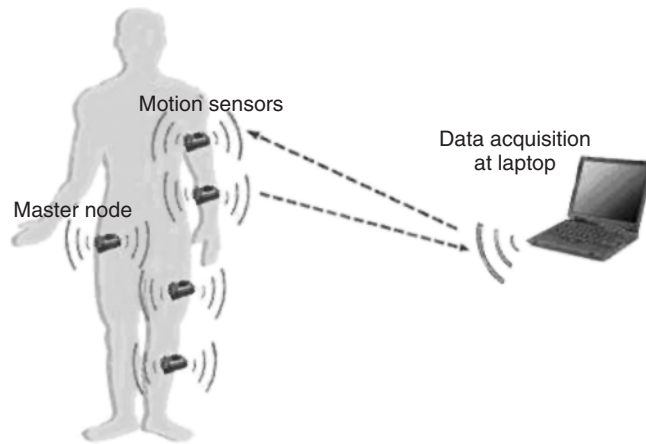


Figure 1.1 Monitoring limb movement in stroke patient rehabilitation.

a nearby control center via one-hop wireless communication. As these sensors are battery powered, they can benefit from intelligent sensor management that provides energy efficiency as well as quality of service (QoS) control.

For example, a wireless motion analysis sensor for stroke patient rehabilitation was studied in John *et al.* (2005). Wearable sensor motes with armbands were attached to stroke patients to monitor their limb movements and muscle activity during rehabilitation exercises. The sensor board consisted of a three-axis accelerometer, a gyroscope, and various electromyogram (EMG) sensors. It was able to capture a rich set of motion data used for studying the effects of various rehabilitation exercises on the patient population. The collected data was transmitted to a data acquisition or control center, such as a laptop or PC, via one-hop wireless communication. The system architecture is shown in Figure 1.1 below.

Some large-scale sensor networks may also be single-hop in terms of wireless communication needed for reporting. For example, the sink, or several sinks, could be mobile and move around the network. This would allow them to get close enough to the sensors so that report collecting could be done in a single hop. In other examples, embedded sensors could move toward a fixed sink. For example, sensors can be embedded into sea mammals to trace their locations over time. When a sea mammal approaches a fixed BS, reports can be downloaded.

1.3 MULTIHOP WIRELESS SENSOR NETWORKS

Nodes in the WSNs are generally randomly and densely deployed. For example, thousands of sensor nodes may be dropped from airplanes to monitor an interested area. Once deployed, these sensors are expected to self-configure into a wireless network. Since the energy budget of an individual sensor is very limited, the transmission range of sensors is also restricted. Thus, WSNs usually operate in a

multihop fashion. A large number of sensor devices can be organized in a multihop fashion to provide unlimited potential to “sense” the physical world. Reports from individual sensors are sent to other sensors, where they can be combined with other sensor readings or simply retransmitted to other sensors until a sink node that is capable of communicating with a user is reached. Therefore, individual sensor readings may need several wireless hops to reach a BS. Such WSNs have received significant attention in recent years. A WSN usually consists of a large number of low-cost and low-energy sensor nodes, which can be deployed on the ground, in the air, in vehicles, or inside buildings. Nodes in WSNs sense data, find routes, and forward sensing data to a sink or BS that is usually far away from the data source. Since sensors usually have a small size, low-battery capacity, nonrenewable power supply, limited processing ability, small buffer capacity, and a low-powered radio, WSNs pose new challenges to both industrial and academic communities.

Applications for WSNs have been envisioned for a wide range of areas. These include, but are not limited to, the following: environment monitoring (e.g., traffic, habitat, security, etc.), infrastructure protection (e.g., power grids and water distribution), fire prevention, agriculture, health care, chemical plumes tracking, building monitoring or control, warehouse management, smart transportation, and context-aware computing (e.g., smart homes and responsive environments) as well as industrial sensing, diagnostics and process control, biomedical sensor engineering, water and waste management, military applications, and so on.

Most of the scenarios considered contain a single sink (also called *base station*), which is normally static. The sink in a WSN collects information from sensors and then analyzes and processes the information for specific applications. The sink could be connected to the Internet via wireless or wired communications such that a remote user is able to inquire about data via the Internet (at any time or from anywhere). Single sink scenarios, or scenarios with multiple fixed or mobile sinks, have also been explored in literature. Sensors in WSNs are usually static. However, they can be mobile when attached to robots, soldiers, or vehicles.

1.4 EVENT-DRIVEN, PERIODIC, AND ON-DEMAND REPORTING

There are three types of applications for WSNs and each has its corresponding data communication modes: *event-driven*, *periodic*, and *on-demand reporting*. In the *event-driven* mode, sensors report the sensing data to the sink once a specified event (e.g., fire) has been detected. In the *periodic reporting* (or *time-driven*) mode, sensor nodes gather information from the environment at predetermined times and periodically send the data to the sink. In the *on-demand* (or *query-driven*) mode, users decide when to gather data. They send instructions to the WSN indicating that they wish to receive data and then wait for the required type of data to be sent in the requested format. Users may even specify the future reporting periods; subsequent reports would then be sent in periodic reporting mode.

Target or event detection and tracking is a typical example of applications in event-driven reporting. Its purpose is to detect, classify, and locate specific targets or events, as well as track the targets or events over a specified region. Once there is an event or a target emerging in the area, the sensor nodes around the target or event gather the required information and report back to the sink. One characteristic of event-driven reporting is its real-time requirement. This means that data transmission latency is one of the key problems in these applications.

Targets can be divided into two categories: targets in the first category are individual objects that usually have a small size when compared to the sensing area of the network. These targets emit noise, light, and seismic waves, such that nearby sensor nodes are able to detect and track them. A typical example is to deploy a sensor network to detect troops, such as tanks and soldiers, in a battlefield. Once a tank moves into a specific area, information on the tank such as its location and speed, will be gathered by the sensor nodes and reported to the BS via multihop communication. The targets in the second category are continuous objects, which spread in the sensing area of the network. An example is the use of WSNs to detect and track diffused poison gas or chemical/biochemical liquids.

Figure 1.2 shows a typical scenario of event-driven reporting in WSNs. Sensor nodes are deployed in the sensor field to form a wireless network. Once there is an event in the monitoring field, such as a fire, a nearby sensor node, say *A*, will detect the fire if the sensed temperature exceeds a predefined threshold. Then *A* either starts the routing process (reactive) or uses the route in its routing table (proactive, e.g., *A-B-C-D-E*), to report information of the event to the sink. The sink may then take appropriate actions immediately or store the data in the database for future statistical use.

Periodic reporting is different from event-driven reporting. Data gathered in periodic reporting does not require urgent delivery to the sink. Further, the data in the event-driven reporting usually comes from sensors in the vicinity of a target or event, whereas the data in periodical reporting is normally gathered from sensor nodes throughout the sensor field.

Sensors report to the sink by applying *data gathering* and *data aggregation* operations. Data gathering refers to forwarding the measured data to the sink

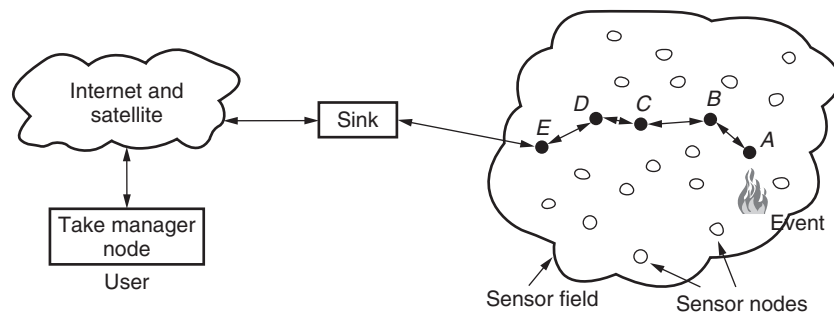


Figure 1.2 A scenario of event-driven reporting.

without further changes on the way toward sink. This is normally achieved via a *routing* task, that is, sending a message from a sender node (sensor) to a destination node (sink), using other sensors to forward the report. However, data collected by sensor nodes might be redundant, correlated, and/or inconsistent with data from other sensors. Data aggregation is used to combine data coming from different sensor nodes. This eliminates redundancy and minimizes the number of transmissions.

A general approach employed in data gathering and data aggregation is to construct a spanning tree which is rooted at the sink and connects all sensor nodes in the network. If one node fails, the topology will be reorganized into a new topology. Tree maintenance is usually an energy-demanding operation.

In data gathering or aggregation, data from each sensor is forwarded to the sink along the spanning tree. This is illustrated in Figure 1.3 where a WSN is deployed for agricultural applications. A large number of sensor nodes are scattered throughout a field to monitor the temperature, light levels, and soil moisture. The sink, located in the house, queries the sensors, which configure themselves. The reporting tree is constructed in the process and rooted at the sink. Data is periodically collected from all sensors in the field and sent to the sink. In data gathering operations, individual data from each sensor is forwarded along the tree without being combined with measurements from other sensors. Data aggregation may be applied too, for example, combining the readings from all of the sensors inside a zone and submitting a combined report via data gathering



Figure 1.3 Data gathering or aggregation in agricultural applications.

operations along the tree hop-by-hop toward the root. For instance, information on soil conditions in different zones of the same field might be needed to apply uneven amounts of fertilizer. Sometimes a single report from the whole field would suffice such as information on the current temperature. In this case, upon receiving the data from each child node in the tree, a sensor node aggregates the data with its own before delivery to its parent node in the constructed tree.

The traditional view of large-sized static sensor networks with one fixed sink has been challenged by their theoretical and simulation analysis discovering some bottlenecks in their performance. For example, it was reported that while using the same transmission range for sensors optimizes energy per report (without data aggregation), it also creates *energy holes* around the sink while the periphery is left with almost full energy (Olariu *et al.*, 2006). Moreover, data aggregation is often impossible. For example, sensors monitoring movements do not generate the same reports and sink instructions are also not aggregated. Therefore, the problems do not seem to have a resolution unless the model itself is changed: It should be either small scale (e.g., up to hundred nodes) or involve multiple *sinks*, *mobile sinks*, *mobile sensors*, and so on. However, this in turn complicates network layer protocols.

1.5 UNIT DISK GRAPH MODELING, HOP COUNT METRIC, AND PROBABILISTIC RECEPTION

Multihop wireless communication in networks of equal devices applying same and fixed transmission radii (i.e., a homogeneous network), has a simple modeling that is an excellent and extremely useful simplification of the complex physical layer. In the *unit disk graph* (UDG), two nodes communicate if and only if the distance between them is at most R , where R is the transmission radius which is equal for all nodes. A UDG is therefore determined by the positions of nodes and a fixed common transmission range R . To illustrate this, if we use $R/2$ as the radius of the disk of each node, two nodes are connected if and only if their corresponding disks intersect. An example of a UDG is shown in Figure 1.4 below. Unit disk graphs successfully model WSNs, wireless *ad hoc* networks (used in rescue, conference, and battlefield scenarios), vehicular network communications, and wireless networks of actuators (to be defined shortly). In combined networks, such as sensor and actuator networks, they can model communication of each component network separately by using different transmission radii for them.

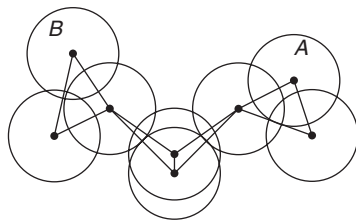


Figure 1.4 An example of a unit disk graph (with radius $R/2$).

Hop count can be used as a metric for routing in UDG if each node applies the same and fixed transmission power. It is defined as the number of hops from one node to another. Hop count between two adjacent nodes is 1. In Figure 1.4, the hop count between node A and node B is 4. In homogeneous networks, where nodes do not adjust transmission radius, the route with the smallest hop count from a source to a destination guarantees the minimal energy cost and the lowest transmission latency (assuming that the delay at each node is the same).

Although the protocols at the network layer are mostly designed with ideal UDG assumptions, experiments are normally carried out on simulators that implement more realistic physical and medium access control (MAC) layers. The UDG model is not realistic since variations in the received signal strengths are not considered. In fact, it has been pointed out that the impact of signal strength fluctuations is sometimes more significant than the impact of node mobility (Stojmenovic *et al.*, 2005a). Therefore, nondeterministic radio fluctuations cannot be ignored when designing robust protocols for sensor and *ad hoc* networks. In addition to distance, the received signal strength also depends on other factors such as environment and transmission medium.

Existing physical layer models, such as the combined Friis and two-ray ground model (Nadeem and Agrawala, 2004) and the lognormal shadowing model (Stojmenovic *et al.*, 2005b), require nodes to estimate the probability of receiving a bit or a packet based on either signal strength, distance between nodes, or merely by deriving statistics from a number of bits or packets recently sent between two nodes. The realistic physical model normally uses a function to represent the packet reception probability. For instance, the packet reception probability $p(x)$ in the shadowing model (Stojmenovic *et al.*, 2005b) depends on the length of the packet, and the distance x between two nodes. Suppose R is the distance so that the packet reception probability is $p(R) = 0.5$, the function $p(x)$ may have approximately the following values: $p(0) = 1$, $p(0.1R) \approx 1$, $p(0.5R) \approx 0.9$, $p(R) = 0.5$, $p(1.5R) \approx 0.25$, and $p(2R) = 0$. The values give a sufficient intuition on how to design physical layer-aware routing protocols. If a fixed signal-to-noise ratio (SNR) is assumed then the function $p(x)$ looks like the graph in Figure 1.5 (Kuruvila *et al.*, 2005). In this example, the probability for successful transmission at distance $d = 30$ is more than 0.95. If $d = 41$, the probability for successful transmission is around 0.5. This means that approximately half of the transmissions are successful. If $d = 50$, the probability for successful transmission decreases to around 0.05. Two nodes can still communicate as long as they make a sufficient number of attempts.

At the physical layer, the hop count metric may not properly reflect the real cost involved in a route. For example, suppose there are many long edges in the shortest path, in terms of hop count. Many retransmissions may be required between adjacent nodes on these long edges due to low probability of packet reception. Thus, the *expected hop count (EHC)* should be used instead. Expected hop count is defined as the expected number of messages between the sender and the receiver, including retransmission, acknowledgments and so on. Extended hop

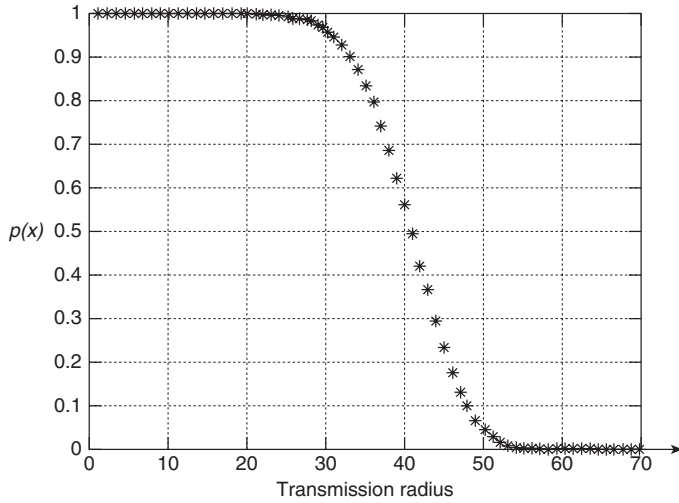


Figure 1.5 Packet reception probability versus transmission radius.

count measures can be also used to measure the cost of a route, by summing EHC values on each edge of the route.

Let S and D denote the sender and receiver, respectively. Suppose that acknowledgment is required for each successful transmission. Assume also that the sender repeats sending packets until it receives acknowledgment from the receiver. Let the distance between S and D be x , and the packet reception probability for transmission from S to D be $p(x)$. Thus, the probability that S does not receive any of the u acknowledgments from D is $(1 - p(x))^u$. That is, the probability that S receives at least one of the u acknowledgements from D is $1 - (1 - p(x))^u$. Therefore, $p(x)(1 - (1 - p(x))^u)$ is the probability that S receives an acknowledgment after sending a packet and therefore stops transmitting further packets. The total EHC between two nodes at distance x is:

$$1/[p(x)(1 - (1 - p(x))^u)] + u/[(1 - (1 - p(x))^u)],$$

where the first term is the message count and the second term is the acknowledgment count. In order to minimize the EHC, the value of u should satisfy $up(x) \approx 1$. That is, the best value of u for a given $p(x)$ is approximately $1/p(x)$ (Stojmenovic *et al.*, 2005b).

1.6 ADJUSTABLE TRANSMISSION RANGE AND POWER METRIC

Sensor nodes have the capacity to adjust their transmission ranges without incurring any significant cost for the adjustment. A common transmission radius is normally preferred because medium access protocols currently considered to be *de facto* standards, such as Zibgee, require it for proper functioning. However,

finding a minimal common transmission radius is a nontrivial problem, especially for its maintenance. A possible compromise is that each sensor is made aware of its neighbors by using “hello” messages. But when the neighbor for forwarding is decided, the transmission power could be adjusted to reflect the distance. Here, we still assume the UDG modeling with adjusted transmission radius. In reality, there is an impact of the realistic physical layer at critical transmission distances, which will be discussed later.

A simple power consumption model is introduced in Rodoplu and Meng (1999). The total *power* needed to transmit and receive a message between two nodes at distance d is proportional to $d^\alpha + c$, where α is the signal strength attenuation factor which is normally between 2 and 5 depending on the transmission medium and the environment, and c is a constant that accounts for signal processing at the transmitter and receiver, as well as the minimal power to receive a signal properly. This model has been restated in Heinzelman *et al.* (2000). The energy consumption per bit is calculated as follows: $power = E_{trans} + \beta d^\alpha + E_{rec}$, where E_{trans} and E_{rec} are distance-independent terms which represent the overhead of the transmitter electronics and receiver electronics, respectively. For simplicity, E_{trans} and E_{rec} are often assumed to be the same (Chen *et al.*, 2004), and c from Rodoplu and Meng (1999) model is proportional to $E_{trans} + E_{rec} \cdot \beta d^\alpha$ is the distance-dependent term that represents the power consumption required to transmit one bit from a sender to a receiver over distance d . If a message contains k bits, the power consumption is then normally multiplied by k .

When the transmission range of the nodes is adjustable, power metric is used to measure the optimality of a routing algorithm in different ways. The simplest way is to measure power consumption at each hop and look for a route that minimizes the total power consumption (the sum of powers consumed at each hop). However, some nodes may be centrally positioned and used on many paths. Their energies can be depleted while the energies of some peripheral nodes may remain close to their maximum. The lifetime of a network may be measured in several ways, including the moment the first node spends all its energy or network partitioning (the moment a particular sensor is not able to deliver its report to the sink because of energy holes in the network coming from sensors left with no energy). Thus, the minimum energy metric routing may not maximize the network’s lifetime since some sensors may suffer early failure. An alternative is to maximize the lifetime of the network.

1.7 COST METRICS

A variety of metrics and their combinations can be used to design and evaluate communication protocols for WSNs. We have discussed so far hop count and power consumption metrics. A convenient metric that can be used to avoid nodes with low remaining energy on a routing path is called *reluctance* (Stojmenovic and Lin, 2001b). The remaining energy g at a sensor node can be normalized in the interval (0,1). The resistance f is then $f = 1/g$, meaning that the reluctance becomes huge when a sensor is close to depletion.

Some applications of WSNs, where real-time or multimedia data are involved in communications, require a guarantee on *QoS* metrics such as delay, throughput, and bandwidth. For example, sensor networks for fire detection require short latency to transmit emergency data to the sink. QoS routing is usually performed through resource reservation in individual nodes along the route.

In the sequel, the term *cost* will often be used to denote one of the mentioned metrics or a newly designed metric which is often a combination of several existing metrics. One example of a combined metric is *power * reluctance*, which can be used in designing routing paths to balance between finding routes with a low total sum of power metrics on route and also avoiding nodes with low remaining energy. The cost metric is therefore often used to find a trade-off among these parameters.

As another example, a conditional max–min battery capacity routing is studied in Toh (2001). If there is at least one route such that the residual energy of each node is greater than a specified threshold, the minimum energy metric is chosen. Otherwise, the route that maximizes the minimum residual energy is selected. In this algorithm, there exists a hidden cost for finding routes needed to elect the best one. The sender node needs global network knowledge to gather it, which requires communication overhead not accounted for in the selected metric of route efficiency. This point will be further discussed in the Section 1.15.

1.8 SLEEP AND ACTIVE STATE MODELING

Energy consumption is one of the key problems in WSNs. Several energy consumption models have been studied in the literature. The following discussion and the graph (Fig. 1.6) are based on the study by Barrenetxea *et al.* (2008). The graph shows energy consumption of a TinyNode sensor mote in different states. The experiment shows that the calculation of the sensor node's receiving costs depends on the assumption of the node's status. If it assumes that the radios of the nodes are always on, the energy consumption of the receiving costs is negligible since the cost for receiving packets has been included in the cost for keeping the radios on. More precisely, the energy consumption is equal to 2 mA when the radio is off but is equal to 16 mA when the radio is on for reception. This means that it takes about eight times more energy for listening compared to sleeping state. The total energy consumption for receiving depends on how long the radios need to be on to receive an incoming packet. Using the example in Figure 1.6 below, we suppose that transmitting a packet at 15 dB consuming 60 mA takes 5 ms. Receiving the packet takes at least 5 ms. However, it is not possible for the node to turn on exactly at the time the packet is sent. That is, to receive the packet, the node should turn on its radio for more than 5 ms (according to the used protocol). In Figure 1.6, the energy consumption of the radio is 15 mA. Therefore, if the total time the radio is on is more than 20 ms, the energy consumption of receiving a packet is more than the energy consumption of transmitting a packet.

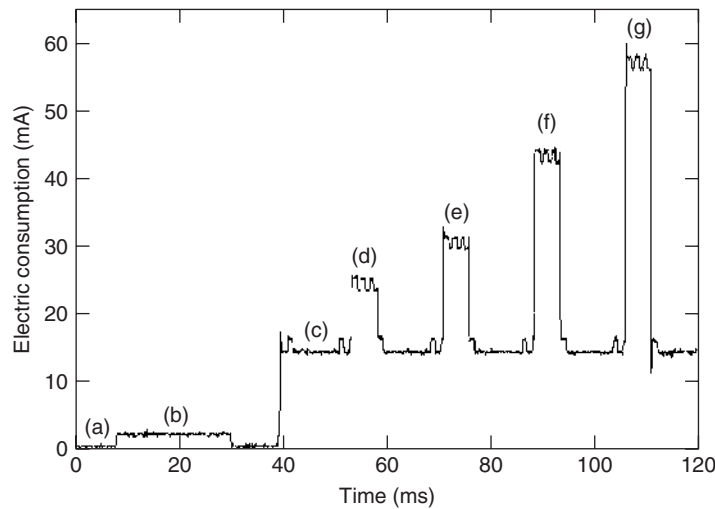


Figure 1.6 (a) CPU off; (b) CPU on; (c) radio on; (d) sending a packet at 0dB; (e) sending a packet at 5dB; (f) sending a packet at 10dB; (g) sending a packet at 15dB.

1.9 ARCHITECTURES FOR WIRELESS SENSOR AND ACTUATOR NETWORKS

Although WSNs have been employed in many applications, such as environment monitoring and health care, there are an increasing number of applications that require the use of actuators along with sensors. This occurs when the network system needs to interact with the physical system or the environment via *actuators* (also called *actors*). From the engineering aspect, an actuator is a transducer that accepts a signal and converts it to a physical action. Actuators transform an input signal into an action upon the environment. Typical examples of actuators are robots, electrical motors, and humans. Traditional sensor and actuator networks use wired communications among themselves; these networks have been well studied. The advent of small, intelligent, low-energy, and low-cost wireless sensing and actuation devices has the potential to significantly expand existing applications of wired sensor actuator networks. Wireless sensor actuator networks (WSANs) are emerging as the next generation of WSNs. The major difference between WSANs and WSNs is that WSANs are capable of changing the environment and physical world while WSNs cannot. Wireless sensor actuator networks are envisioned for applications that include disaster relief operations, intelligent buildings, home automation, smart spaces, pervasive computing systems, cyber-physical systems and nuclear, biological, and chemical attack detection (Xia *et al.*, 2007).

A WSAN usually consists of a group of sensor nodes that are used to gather information from the environment, and actuator nodes that are used to change the behavior of the environment. There are wireless links between the sensor

and actuator nodes. Sensor nodes sense and report the state of the environment while actuator nodes gather data from sensors and are able to act on the environment. Wireless sensor actuator networks are expected to be self-organized and potentially operate autonomously in unattended environments, with basic and minimal directives from the user that might be remotely connected to the scene. In typical applications of WSAWs, sensor nodes are static while actuator nodes, such as robots and humans equipped with vehicles, are mobile. However, some actuators such as sprinklers in fire detection systems, could be static. Sensor nodes also could be mobile in some scenarios (mobile sensors). For example, in sensor relocation problems, sensor nodes are required to move to locations of failed sensors for continued area coverage. Sensors and actuators can even be integrated into a single robot which is capable of sensing and moving. Compared to sensor nodes, actuator nodes usually have stronger capabilities in data processing, wireless communication, and power supply (Melodia *et al.*, 2007). Therefore, the number of sensor nodes deployed in a monitoring region may be in the order of hundreds or thousands while such size is not necessary for actuator nodes since they have higher capabilities and can act on larger areas.

Coordination is another aspect of WSAWs (Akyildiz and Kasimoglu, 2004). Unlike WSNs, where the sink performs the functions of data collection and coordination, sensor-sensor, sensor-actuator, and actuator-actuator coordination is required in WSAWs to achieve the overall application objective. *Sensor-actuator coordination* provides the path establishment for transmission of event data from sensors to actuators. This coordination may be also needed for some control traffic, such as communication locations of actors to sensors or helping sensors to learn their geographic position with higher precision. After receiving event data, actuators need to coordinate with each other to make decisions on the most appropriate way to perform actions. We refer to this process as *actuator-actuator coordination*. The coordination has additional aspects, such as fault tolerance, activity scheduling, and network design guidelines.

There are two basic architectures for data processing in WSAWs described in Akyildiz and Kasimoglu (2004). One is called *automated architecture*, where sensor nodes sense the environment and report the data to actuator nodes which then initiate appropriate actions based on the received data. This architecture is shown in Figure 1.7a. The second architecture is called *semiautomated architecture* where sensor nodes route sensing data back to the sink which may then issue action commands to actuator nodes. This architecture is shown in Figure 1.7b. Semiautomated architecture is similar to the architecture of traditional WSNs. Therefore, current protocols and algorithms for traditional WSNs can be easily adopted in this architecture. The advantages of automated architecture are as follows. First, since sensing data is reported to actuators which are closer than the sink to sensors, communication latency is minimized. Second, in semiautomated architecture, transmitting the sensing data to the sink usually causes fast energy depletion of nodes which are around the sink. In automated architecture, sensing data is reported to actuators and different actuators may be triggered based on different events. Hence, the communication load can be more evenly distributed

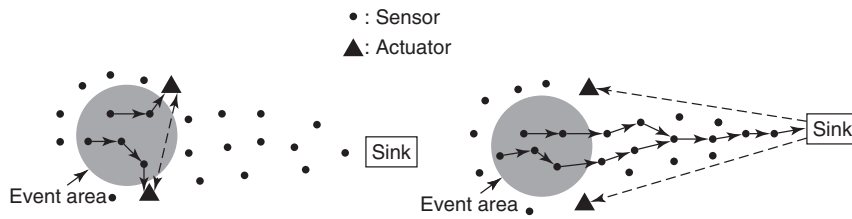


Figure 1.7 (a) Automated architecture and (b) semiautomated architecture.

among all nodes and it results in a longer lifetime of networks. Therefore, the automated architecture is able to provide low communication latency and longer network lifetime, which are desirable in most applications of WSANs.

The third architecture is proposed in Stojmenovic *et al.* (2007) and will be referred to here, as the *cooperative architecture*. In this architecture, sensor nodes transmit sensing data to actuator nodes via a single-hop or multiple hops. The actuators analyze the data and may consult the sink(s) before taking any action. That is, actuators may use their peer-to-peer network to make decisions and take action, possibly informing the sink about the action taken, or could inform the sink and wait for further instructions from the sink. A user (task manager) controls the network via the sinks. One or more of the actuators may also play the sink role. In fact, sinks can be treated as special kinds of actuators, although a better interpretation might be to associate them with BSs that communicate directly with the user. The architecture is illustrated in Figure 1.8, where one sink is linked to one of the actuators while the other actuators can reach the sink in a multihop actuator-actuator structure. Usually, actuators are more powerful and have a larger transmission radius than sensors. In extreme cases, actuators are able to directly reach all sensors in the network. Sensors route their data to any actuator. This task is known as the *anycasting* problem if a sensor is aware of the geographic positions of all actuators and itself. Sensors may start reporting

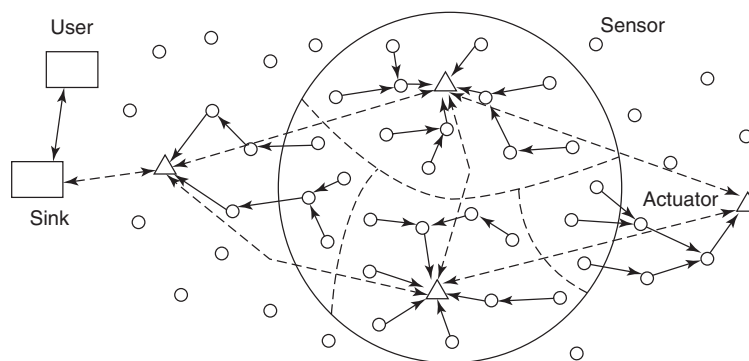


Figure 1.8 Cooperative architecture of sensor-actuator networks.

toward one selected actuator, but another actuator could be the ultimate receiver after some dynamic changes of the message destination. Alternatively, routes can be created by flooding from all actuators and memorized by sensors for reporting back toward the nearest sink.

The sink monitors the overall network and communicates with the task manager node and the sensor or actuator nodes. If necessary, the sink issues commands to actuators which forward them to sensors located inside the area that needs to be monitored (the circle in Figure 1.8). After sensors detect an event occurring in the environment, the event data is locally processed and transmitted to the actuators, which gather, process, and eventually reconstruct the event data. Coordination is one of the primary characteristics of WSNs.

Ruiz-Ibarra and Villasenor (2008) proposed a taxonomy for cooperation mechanisms in wireless sensor and actor networks. Wireless sensor actuator network frameworks consist of an architecture network (automated or semi-automated), a coordination level (sensor-sensor, sensor-actor, actor-actor), node mobility (fixed or mobile), and a network density (dense or sparse). Collaborative procedures include routing protocol, synchronization, localization, aggregation, clustering, encryption, power control (for event reporting and task execution), and QoS. The performance criteria include optimization criteria (metrics for reaching stated objectives), scalability, complexity order, and reliability (security, robustness). The application requirements consist of real-time constraints, event frequency, and concurrent events.

Some other architecture may be envisioned for futuristic applications. For example, Figure 1.9 shows a vision of merged *ad hoc* sensor and actuator networks in military applications. Sensors are placed in the field to detect minefields and firing locations, for target tracking, detecting chemical and biological attacks, and can be also attached to soldiers and vehicles. Vehicles, soldiers, and airplanes can also serve as actuators in the network.

1.10 SIMPLE MODELS AND APPLICATION OF WIRELESS SENSOR AND ACTUATOR NETWORKS

Current applications of WSNs rarely use theoretical models described in the previous section. There exists a gap between theoretical achievements and practice. We describe here several simple models for wireless sensors and actuators from recent literature, which do not really fall within presented classification. In all cases, the wireless communication is a single-hop one, direct communication between sensor and actuator or between two actuators.

A classical *star topology* of WSNs was studied in Korber *et al.* (2007). In the star topology, the BS serves as a network controller and as a gateway to upper layers. The BS may have a wired bus and a wireless radio interface. The TDMA (time division multiple access) technique is employed in the MAC layer. Each sensor is integrated into an actuator to form a sensor-actuator module. These modules are able to reach the BS in one-hop wireless communication. A time and frequency slot is allocated for each sensor and actuator, such that communication

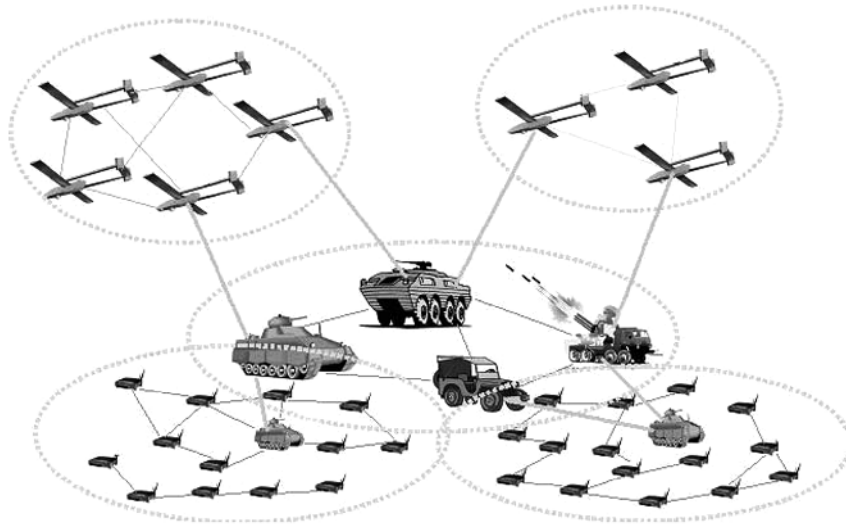


Figure 1.9 An example in military applications.

collisions between the sensors and actuators can be avoided. There is a trade-off between QoS, for example reliability and real-time communications, and the lifetime of nodes in WSANs. However, in many applications of WSANs, it is preferable to guarantee real-time communications and defined timing behaviors. The star topology is a good solution to satisfy the real-time requirement. The authors Korber *et al.* (2007) also argue that single-hop wireless communication is important for reliable industrial applications.

There is an application of WSANs in bull breeding paddocks, which is used to control the aggressive behavior of bulls. This application is described in Wark *et al.* (2007). Fighting between bulls during the breeding season may result in serious injuries to the bulls, which are high value animals. Therefore, it is critical for the breeding industry to protect these high value animals without human intervention. The idea is to deploy sensors and actuators in the cattle collars, which serve to detect and control the bulls' behavior. A hardware platform is capable of integrating a wide range of sensors and actuators, and it consists of an Atmega 128 processor and an 8MB flash memory. The onboard radio transceiver and hardware platform, along with the integrated stimuli board which acts as an actuator, are mounted inside a specially designed collar. The integrated sensor is able to estimate the dynamic states of the bulls from location and velocity observations. Once aggressive behavior in a bull is detected, the actuators initiate a stimuli on the bulls.

A sensor and actuator network in smart homes for supporting elderly and handicapped people was studied in Dengler *et al.* (2007). The primary goal was to monitor domestic systems such as air conditioning, lights, and heating, as well as to control the basic functions of the home entertainment and security systems. In the experiment, a test area which included "a living area" and "a kitchen" was

used to represent the smart home environment. The sensor network consisted of three BTnodes, an autonomous wireless communication and computing platform based on a Bluetooth radio, and a microcontroller. Each BTnode was equipped with BTsense v1.1a sensor boards and proper sensors for light, motion, and temperature detection. Each sensor measured data at intervals of 30 s. If a sensor sent more than five unacknowledged emergency calls, the mobile robot was programmed to move directly to the sensor node. Another application for smart home monitoring has also been described (Li, 2006).

An example of the use of WSAWs to monitor environments is the fire detection system. A group of sensor nodes are placed in a building or an area of interest. In the event of a fire in the monitoring region, the sensor nodes that are close to the origin of the fire report the location and intensity of the fire to water sprinkler actuators. On receiving alarm messages from sensor nodes, the water sprinkler actuators analyze the intensity of the fire and take appropriate actions before the fire becomes uncontrollable.

1.11 GENERATING CONNECTED WIRELESS SENSOR AND ACTUATOR NETWORKS

In this section, we will discuss the population of connected wireless sensor and/or actuator networks, and how to generate one particular sample from the population for the purpose of simulating them and evaluating performance of proposed communication protocols. Typically, in literature, connected *random UDG* is employed in generating wireless sensor and/or actuator networks. It is generated by placing a group of nodes in a specific area pattern, such as a rectangle or a circle. The positions of N nodes are randomly determined (e.g., by selecting their two or three coordinates at random) and are independent from each other. The desired network topology is achieved once the generated topology passes the connectivity test (usually by running centralized Dijkstra's shortest path algorithm). The expected node *degree* (average number of neighbors per node) is the number of remaining nodes, $N - 1$, times the probability that any node will be placed within the node's transmission area. This probability can be approximately calculated by dividing the transmission area by the total area. Thus the expected degree, d is $\approx (N - 1) \times \pi r^2 / A$, where A is the area of the region of interest and r is the transmission radius. That is, $r \approx \sqrt{dA / (N - 1)\pi}$. The exact average degree D for the generated graph is only an approximation of the desirable average degree d used in the graph generation.

It was observed that simulations in existing literature were using transmission radius r as the independent variable, while the corresponding average degree d was normally not even reported (Stojmenovic and Lin, 2001a). This has led to reporting simulation data for only dense graphs, for example, and hiding delivery problems for sparse graphs. To achieve the desired and accurate network density d , and use it as a parameter in simulations to study performance networks ranging from sparse to dense, Stojmenovic and Lin (2001a) proposed a method to control the average number of neighbors d by adjusting the corresponding common

transmission range R . First, N nodes are generated at random. All $(N-1)N/2$ edges are sorted into a list in nondecreasing order. The transmission range R is set to be the length of the $(Nd/2)$ th edge in the sorted list of all distance lengths. There is an edge between the two nodes if and only if their Euclidean distance is not greater than R . The generated network is then tested for connectivity.

The two generation algorithms, with approximate and accurate average degree as parameters, suffer from two problems. Since sparse networks have a high probability of being partitioned, they may generate a lot of disconnected topologies and take a long time before a connected UDG is obtained. Although it is reasonable to assume that the positions of nodes are independent in some scenarios, some networks may have characteristics different from random UDGs. For example, actuator nodes in WSNs create their own network to facilitate coordination and enhance data communications and actuation performance. This imposes certain restrictions on their locations with respect to each other. In some applications, the position of a newly deployed node may depend on the positions of other nodes that are already in the region of interest. For example, laptops of attendees in a conference form a multihop *ad hoc* network. When a new attendee with a laptop enters the conference venue, a good choice is to sit not very far from the others, so that the network service is available. At the same time, the new attendee may try to avoid overpopulated areas in order to have an acceptable throughput.

Fast generation of several types of wireless *ad hoc* networks where new node placement is dependent on other nodes' placements, has been studied (Onat *et al.*, 2008). Two classes of algorithms: rejection-acceptance and center node-based algorithms were proposed. In center node-based algorithms, for example minimum degree proximity algorithm (MIN-DPA), a center node is chosen among the already placed nodes before the placement of a new node. The new node is placed around the center node. In rejection- or acceptance-based algorithms, a random candidate position is selected for each node during each round. Then, the position is either accepted or rejected depending on some constraints.

There are several constraints for placement of nodes. In the *proximity constraint*, a new node is placed at a minimum safe distance from any other existing nodes and should be closer than the approximate transmission radius from at least one of the existing nodes. In the *maximum degree constraint*, a new node position is rejected if its placement would make one or more of the existing or new nodes have a degree exceeding the maximal one set by a threshold. For sensor networks, *coverage constraint* is important. A candidate sensor position is accepted only if it sufficiently increases the overall coverage area. In the extreme case of aiming at full coverage, a candidate sensor is accepted if its coverage area is not fully covered by already placed nodes.

The basic idea of MIN-DPA is to place each new node around the center node that has the smallest degree in the current graph. The first step of the algorithm is to determine an approximate radius r , which is used to estimate degrees in the process. One of the existing nodes with the minimum degree is selected as the center node. A new node is uniformly and randomly placed

within the transmission range of the center node (subject to possible boundary constraints). The procedure continues until all the nodes are placed. At the end, the approximate transmission radius r used in the generation process is replaced by the transmission radius R corresponding to the desired average degree d , using the “edge sorting” algorithm (Stojmenovic and Lin, 2001a) described above. After the placement, the connectivity of the topology is checked. Simulation results show that generating a connected graph using MIN-DPA is significantly faster than using UDG, especially for sparse networks.

In MIN-DPA, the position of a new node affects the degree of already placed neighboring nodes. MAX-DPA (maximum degree proximity algorithm) imposes a maximum degree constraint for all nodes in the network. In round i , a random position is chosen for node i , and accepted only if none of the existing or new nodes exceeds the maximum degree allowed d_{\max} . After all the nodes are placed, the transmission radius is adjusted and the connectivity of the topology is checked as in the case of MIN-DPA.

1.12 GENERATING MOBILE WIRELESS SENSOR AND ACTUATOR NETWORKS

A lot of research has been conducted on fixed sensor networks, where connectivity is normally demanded. However, a number of applications require mobile sensors. Mobile sensors and mobile actors may not preserve connectivity and often the application itself involves sporadic connectivity. Examples include vehicular networks where cars can be seen as sensors carrying information or actuators with possible actions such as changing speed, lanes or roads. People or wild animals can also act as actuators.

The model based on social network theory (Musolesi and Mascolo, 2007) views networks as collections of disconnected clusters. Each cluster is a connected network and nodes may move occasionally from one cluster to another (social movement), according to attractive “virtual forces” from other clusters. This is illustrated in Figure 1.10.

Some applications are based on harsh environmental conditions, such as underwater sensor networks of seals. In this application, batteries are impossible to change and could be lost since seals change their fur periodically. Networks can be very sparse. Seals can meet in clusters but then, they rarely meet at sea. There is no human pattern of day or night behavior.

It is worthwhile to mention that a collection of real mobility traces in various wireless networks is maintained at <http://crowdad.cs.dartmouth.edu/>.

1.13 PROBLEMS AT PHYSICAL, MAC, AND TRANSPORT LAYERS

Since a WSAN can be treated as a union of a WSN and an actuator network (mobile *ad hoc* network), current problems and challenges with sensor networks and *ad hoc* networks also exist in WSANs.

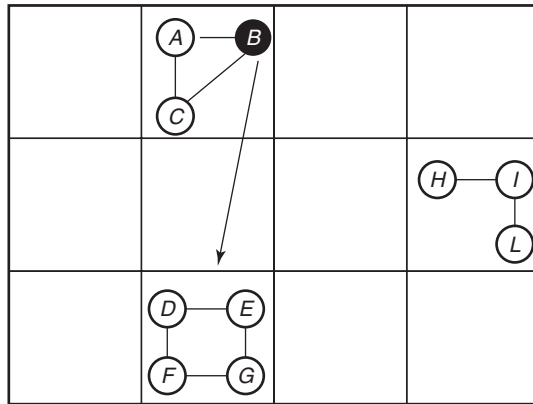


Figure 1.10 Social clustered network and social movement.

At the *physical layer*, it is required to process signals, deal with the hardware failure of sensor nodes, manage limited bandwidth and limited power, control sensing range and transmission range, and select antennas and operating channels. Energy scavenging and nontraditional power sources are surveyed in Roundy *et al.* (2005). Sensors use wireless communication, where RF noise and multipath fading causes severe packet losses. It is easy to eavesdrop and to launch spoofing or Denial-of-Service attacks. Infrared and optical lines of sight are alternatives that are considered. Nodes are at physical risk because they can be defective, lost, damaged, compromised or can have expired. Wireless communication implies limited bandwidth and in most cases also limited power (unless rechargeable battery, solar power or other energy supply alternatives are feasible). Wireless communication also implies one-to-all communication where messages sent by one node are simultaneously received by all neighbors within transmission radius. Smart omnidirectional antennas can also be considered (especially for actors) but this makes sensors more complex. Small processing power limits processing time and reduces the choices available for security solutions, data compression, and error control techniques. Routing table sizes are small due to reduced memory size and also reduced usefulness of such tables. Sensors are normally assumed to be all on the same frequency (or perhaps two frequencies, one being used for some control messages), since otherwise lots of communications sent on a “wrong” frequency can cause significant loss of energy and also the inability to find neighbors in a timely manner, during the time neighbors are active. Thus, frequency-hopping solutions like Bluetooth are currently not considered feasible for WSNs.

The *MAC layer* primarily aims at energy-efficient and collision-free communications. Medium access in WSNs is currently envisioned via IEEE 801.15.4 standard (IEEE Standard 802, 2003) and its extension known as Zigbee. ZigBee network specification (ZigBee Alliance, 2004) is one of the first standards for *ad hoc* and sensor networks. ZigBee is a specification for a suite of high-level communication protocols using small, low-rate, and low-power digital radios for wireless personal area networks (WPANs). The technology is intended to

be simpler and cheaper than other WPANs, such as Bluetooth. Two network topologies are allowed by the standard, both relying on the presence of a central coordinator. In the peer-to-peer topology sensors (devices) may communicate directly, while in the star-shaped topology they must communicate through a coordinator. In a typical ZigBee network, the network addresses of nodes are organized in a hierarchical manner, such that one node can easily identify the addresses of its tree neighbors, for example its parent and children. A coordinator buffers all the packets for its associated devices, such that devices can go to sleep mode and wake up only when they need to retrieve data from the coordinator. Moreover, the coordinators route all packets for the devices. One can observe that the roles of coordinators and devices are similar to the roles of actuators and sensors, respectively. Sensors are time synchronized and follow a joint sleep–active schedule. They are active at the same time, followed by longer sleep periods. At the beginning of active periods, they compete for upcoming slots to send messages. While sensors are all active, Zigbee medium access works similar to the popular WiFi standard based on IEEE 802.11. Time is slotted. A station that has a message to transmit will first wait for a few slots of interframe separation. It generates a random number x in a certain interval (e.g., $[0,31]$) and uses carrier sense to determine whether or not the channel medium was used in each slot while waiting for retransmissions. The station will wait for x idle slots (without transmissions from any station) and afterwards will transmit the full message without further verification of a possible collision. Some other proposals that deviate from random access-based Zigbee were also considered. For example, Z-MAC (Rhee *et al.*, 2005) combines TDMA and CSMA. It switches MAC to CSMA and TDMA when contention is low and high, respectively.

At the *transport layer*, data gathering and data aggregation is scheduled in order to reduce traffic, increase reliability, and provide QoS control. Most of these problems are well-studied in WSNs and *ad hoc* networks. Note that traditional end-to-end reliability in wired networks is not applicable in wireless networks since link failure is possible due to the mobility or energy depletion of nodes. Thus, the QoS issues in WSNs usually call for reliability of communications rather than bandwidth and/or delay. Individual sensor measurements are not reliable and need to be combined with readings from other sensors to achieve collective reliability. The primary task in the transport layer in WSNs is, in fact, to achieve a sufficient level of reliability in the sensor network reports to the sink while minimizing their energy and bandwidth resources.

Wireless sensor networks and WSANs have a number of additional issues, often of a cross-layer nature, each of which is covered in literature and is important for overall network functioning. They will not be investigated in this book, which instead concentrates on basic network layer problems. The reader is advised to consult some other sources, such as handbooks on sensor networks (Stojmenovic, 2005), which cover authentication, key management, security issues, operating systems, databases, path exposure, target location, classification, tracking, data gathering and fusion, localization (position determination), time synchronization, and calibration.

1.14 PROBLEMS AT THE NETWORK LAYER

Current problems at the network layer can be classified into three categories: topology control, routing, and coordination. Their coverage follows.

1.14.1 Topology Control

A well-organized network topology can not only prolong the lifetime of a network, but also enhance data communications. Topology control problems can be subdivided into neighbor discovery problems and network organization problems. Neighbor discovery problems are defined as problems in detecting and discovering neighbors which are located within the transmission range. In the network organization problems, each node chooses its neighbors and constructs local topology by either adjusting its transmission power or setting its status, such as sleep and active modes. There are some protocols that achieve desired network topology by movement control on nodes. For example, the localized mobility control protocol in Das *et al.* (2007) constructs a biconnected network from a connected network through the movement of nodes. In this and a number of other protocols, topology control is used to create fault-tolerant networks for reliable communication protocols.

The most important topology control, especially for power-critical sensor networks, is to place as many possible sensor (and similarly actor) nodes into a sleep mode as possible. All nodes that are not essential for communication or area coverage can be placed in sleep mode for prolonged periods, synchronously or asynchronously. This is in addition to synchronized sleep–active state changes of currently active sensors for power efficiency at the MAC layer. There are a number of studies on energy efficiency at the MAC layer, which are also based on topology control. S-MAC (Ye *et al.*, 2004) divides nodes into clusters based on fixed common sleep schedules to reduce control overhead and enable traffic-adaptive wake-up. T-MAC (Dam and Langendoen, 2003) extends S-MAC by adjusting the length of the waking time of the nodes based on the communication of neighboring nodes. B-MAC (Polastre *et al.*, 2004) employs an adaptive preamble sampling scheme to reduce the duty cycle and minimize idle listening.

Some topology control schemes aim at selecting certain nodes from the network to create a *backbone* that can be used in several ways. A backbone is connected if the network of solely backbone nodes remains connected (after selecting them from the originally connected network). Some backbone structures are used to improve the efficiency of data communication protocols. For example, routing or broadcasting remains successful if intermediate nodes are selected only from connected backbones since each nonbackbone node has a neighbor from the backbone. Another possible application of the backbone set is to place the remaining nodes into sleep mode.

Clustering and *connected dominating sets* (CDSs) are two basic techniques used to generate the backbone for wireless sensor and *ad hoc* networks. The

clustering process divides the nodes of a network into several clusters. In each cluster, there is a *clusterhead*, which is responsible for the coordination and data communication between nodes in the cluster. The selection of clusterheads is done via global nomination or local election, according to a certain protocol. Communications within a cluster could be one hop or multihop. The backbone could contain only clusterheads or may include some gateway nodes to enable connectivity.

Dominating sets are another technique for backbone creation. A subset of the vertices of a graph is called a *dominating set* if every vertex in the graph is either in the subset or is adjacent to at least one vertex in the subset. A CDS requires also connectivity among the backbone nodes. In the example of Figure 1.11 below, subsets $\{1, 2, 3, 5, 6, 10\}$ and $\{4, 7, 8, 9\}$ are dominating sets. The subset $\{4, 7, 8, 9\}$ is also CDS while the former one is not.

Topology control may also be applied to select certain existing edges of the network while ignoring others. This leads to subgraphs that may have useful properties. For example, the Gabriel graph is a planar subgraph of UDG which can be used to guarantee delivery in position-based routing without relying on any memorization (Bose *et al.*, 1999).

1.14.2 Data Communication

In data communication problems, such as routing, QoS routing as well as multicast, broadcast, and geocast, the primary goal is to fulfill a given communication task successfully between nodes in the network. It requires, at the same time, the minimization of communication overhead and power consumption.

Routing is one of the critical issues in almost any type of network. It is used to find a route from a source to a destination in the network. In WSANs, each of the source and destination nodes could be either sensor or actor. In *QoS routing*, selected routes should satisfy the QoS criteria such as delay and/or bandwidth for real-time and multimedia-rich data communications.

In a *multicasting* task, the same message needs to be routed from a source node to a fixed number of k known destinations. *Broadcasting* is a special case

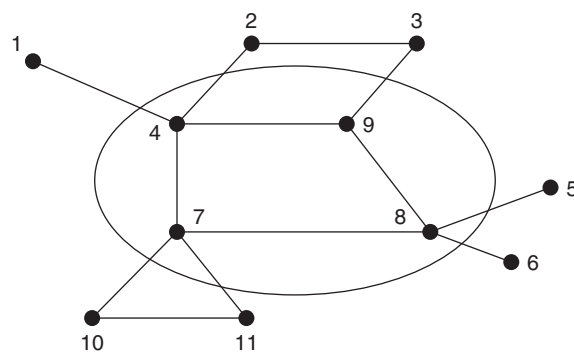


Figure 1.11 An example of a connected dominating set.

of multicasting where $k = N$, the number of nodes in the network. That is, in the broadcasting task the message is to be sent from one node to all the other nodes in the network. In a sensor and actuator network, multicasting is usually applied from a sensor node to send the same report to a fixed set of actuators. In some applications, such as monitoring a certain area, geocasting operation is carried out. In *geocasting*, the source sends messages to all the nodes located in a particular geographic region. For example, an actuator may request all sensors located in a certain region to report sensed movements.

Other basic data communication primitives may be needed in particular scenarios. For example, sensors monitoring certain movements may be required to continuously send video images of the object. At the same time, the actuator or sink may need to improve the quality of the delivered image by improving the communication links along the route. In this scenario, the traffic is large enough in volume and duration to warrant nodes expending energy on movement in order to forward the large traffic in a more efficient manner. Such a mobility control routing task and an algorithm were proposed in Liu *et al.* (2007). The primary goal was to move toward a route with minimal total power consumption while preserving communication during mobility, and achieving this with small total movement distance of actors or mobile sensors involved in routing.

1.14.3 Coordination

Wireless sensor actuator networks require coordination not only among sensors or actuators, but also between them. To facilitate the coordination, the first problem is to achieve proper actuator selection. Sensors need to know where and how to send reports to the closest actuator. Actuators may flood the network with their position or merely their IDs. Sensors receiving such information may rebroadcast them (so that more sensors learn about the same actuator) or ignore them if, for example, a closer actuator was already identified. Such flooding type algorithms from multiple actuators have been discussed in Ingelrest *et al.* (2006). Similar to flooding of route discovery in routing, sensors may learn their paths toward the nearest actuator, in terms of hop count or other metric distance. If position information is available and used in routing, the path may be found dynamically. When a sensor has a data report for actuators, it needs to efficiently find the best actuator to deliver the report, without flooding all the actuators. *Georouting* to the geographically closest actuator is an option, but it is often not the optimal one if, for instance, there is a void area between the sensor and that actuator. Instead, it may be more efficient to try to find a route to any of the actuators. Routing may start toward the physically closest actuator, but the destination actuator may be changed during the path search. This task is known as *anycasting*. The actuator selection problem also exists in the clustering stage of sensor and actuator networks. Each sensor needs to find available actuators and decide to join the cluster which is dominated by an actuator.

Sensors usually report to the sink or actuators via hop by hop transmission. However, mobile actuators are able to move to collect reports periodically via a

predesigned route and thus minimize power consumption of sensors. For example, the U.S. Marine Corps used an unmanned airborne vehicle (UAV) to drop several sensor motes to detect vehicles traveling through an isolated desert area (Hill, 2003). The motes organize themselves to construct a network to monitor moving vehicles and record tracking information. The UAV plane comes back later and retrieves each node's tracking data.

There are several coordination problems associated with *location service* in WSAWs. In location service problems, actuators need to provide and maintain (if they move) position information for sensor nodes and sensors need to maintain position information for the nearest actuator or neighboring actuators. Sensor-sensor, sensor-actor and actor-actor coordination may be used to provide position information to some sensors, using position information of nearby actors and possibly some "landmark" sensors in the field, as well as collaborative processing of neighborhood graphs.

In the *sensor relocation* problem, mobile actuators or mobile sensors move to replace failed sensors. Similar problems include coordinated movement of sensors to place themselves strategically around the point of interest, for efficient monitoring (*focused coverage* problem).

1.15 LOCALIZED PROTOCOLS AS THE SOLUTION FRAMEWORK

On the basis of the information required to run algorithms, existing algorithms or protocols in wireless networks can be roughly classified into two groups: *globalized protocols* and *localized protocols*. In globalized protocol, one or more nodes (usually the central node like a BS) need(s) to gather global information ranging from detailed information such as the whole network topology to simple information of global nature, such as the maximum degree in the network, to execute the protocol. However, in localized protocol, each node makes protocol decisions based solely on some local knowledge available. To be more precise, local knowledge in wireless networks is based on information from neighbors within k hops from a certain node, where k is usually a small integer like 1 or 2. Some protocols, for example beaconless georouting, do not require any information from neighbors. Another such example is the probabilistic flooding scheme (Ni *et al.*, 1999) where each node makes its own decision about possible retransmission using a predefined fixed probability.

Shortest (weighted) path routing is a typical globalized protocol which computes the best route between the source and the destination. Both well known shortest weighted path algorithms, Dijkstra's and Prim's, require all nodes to know full network topology (all nodes and all edges between them) to make proper decisions. Globalized shortest path routing has huge setup costs because of the large number of messages exchanged in order for the source to gather the network topology. Global information is costly to gather and maintain in wireless sensor, actuator, and *ad hoc* networks (unless the network is static and small scale or single-hop). The dynamic nature of these networks (possible

mobility and changes in activity status, arrival, and departure of nodes) requires localized protocols so that communication overhead needed for gathering global information is avoided. When the network size becomes large, performance of globalized protocols is degraded dramatically compared to localized algorithms. If scalability is an important protocol requirement in wireless networks then localized protocols are the best choice. In localized protocols, decisions are made based only on information from neighbors and natural additional information. For example, greedy routing (Finn, 1987) is a localized protocol which uses only position information of one-hop neighbors and the destination. In greedy routing, the current node on the route selects the neighbor that is closest to the destination.

In many cases, the global nature of described protocols appears hidden. One example is the well-known Bellman–Ford algorithm, used for Internet routing, which also finds shortest path routes. Neighboring nodes periodically exchange their routing tables that contain the costs and forwarding neighbors for each destination. Routing tables are then updated after each such exchange. While this appears to be a localized algorithm because of exchange between neighbors as the only apparent communication, a repeated application of that operation involves the arrival of important information from distant nodes in summary form. Therefore the algorithm is globalized. Some other algorithms are claimed to be localized in the literature although the message complexity for each node is not bounded (it is often $O(d)$, where d is the number of neighbors). While the required information appears again to be strictly localized, the information gathered may still be of global nature in the process and the algorithm is not localized.

Localized protocols can be further classified based on the amount of information required and the overhead in the construction and maintenance phases. The amount of required information is related to the message complexity which is defined as the average number of messages exchanged per node in the protocol. In the construction phase, some localized protocols may need extensive message exchanges among neighbors. The amount of messages may then be comparable to that in the collection and use of global information. In the maintenance phase, some localized protocols (in the construction phase) may require message propagation and recomputation of the entire network for a change only in one part of the network, such as maintenance of a minimal spanning tree (MST) and a typical clustering structure. Thus, localized protocols can be further classified into *local localized* (message complexity in the maintenance phase remains low) and *quasi-local localized* (local changes may trigger global updates). Mobility of nodes and status changes between active and sleep modes require localized algorithms, preferably local localized.

Note that the expressed criticism for globalized algorithms does not mean that they are not useful for protocol design in scalable sensor and actuator networks. For example, globalized algorithms can be often applied with limited local knowledge (e.g., two hops), leading to winning protocols in several cases. This will be elaborated later in this book.

1.16 IMPLEMENTATION OF SENSOR MOTES

Many different versions of wireless sensor devices (also called *motes*) have been designed and built by various companies and institutions. The size of these motes varies from the size of a box of matches to the size of a pen tip. The smallest sensors are known as *smart dust*. We now describe several representative sensor motes.

The MICA mote is built by Crossbow in the United States. It consists of the Atmel Atmega 103L processor which is capable of running at 4MHz, has a 128kB flash memory, a 512kB serial flash, 4kB SRAM and a 4kB EEPROM. The MICA mote is powered by two AA batteries and the lifetime is up to 1 year under very low duty cycles. The mote operates at 916MHz or 413MHz and the transmission rate is 40 kbps with a transmission range of 100 feet. The MICA2 mote is the next generation commercial mote by Crossbow. It has the same processor and memory as the MICA mote but the radio transceiver operates on 433MHz or 868/916MHz with a transmission rate of 38.4 kbps. The outdoor range of the MICA2 mote is up to 500 feet. Both the MICA mote and MICA2 mote use TinyOS, an open-source embedded operating system developed at University of California, Berkeley, to control the mote and its attached sensors. The MICA2 motes accept the same sensor boards as the MICA mote.

Intel developed a mote in which the original modular design of the Berkeley motes are maintained while the data processing and battery life are improved. The Intel mote consists of a powerful ARM processor, SRAM, and flash memory. Optimal sensor boards and an optional power regulator are available. It is also based on TinyOS. The software stack includes an Intel Mote-specific layer with Bluetooth support and platform device drivers, as well as a network layer for topology construction and multihop routing. Security features, such as authentication and encryption, are also provided.

Sun SPOT (Sun Small Programmable Object Technology) is a sensor developed by Sun Microsystems (it appears to be the recommended choice in 2008). It can be battery- or USB-powered and is built upon the IEEE 802.15.4 standard. Sun SPOT is able to host a wide number of add-on boards with USB, TWI, SPI, I2S, RMII, USART and SD/MMC interfaces (<https://spot-espot.dev.java.net>). The most significant feature of Sun SPOT is that it runs Squawk Java Virtual Machine (VM) without an underlying OS. This VM acts as both, an operating system and software application platform. Moreover, Sun SPOT is a completely open source technology based entirely on Java technology. The open source release of the Sun SPOT platform includes hardware architecture, software, and the VM. The Squawk VM is the only open source research VM that is Java Logo certified.

Besides sensor motes, there exist integrated sensor and actuator nodes, such as robots, which are designed by several robotics research labs. For example, low-flying helicopter platforms provide ground mapping and air-to-ground cooperation of autonomous robotic vehicles (Thrun *et al.*, 2003). Autonomous battlefield robots sponsored by the Defense Advanced Research Projects Agency are able to detect and mark mines, and carry weapons. The robots developed by

Sandia National Lab may be the world's smallest autonomous robots. They are only 0.25 cubic inch and weigh less than an ounce (Akyildiz and Kasimoglu, 2004).

1.17 EXPERIMENTS ON TEST BEDS

Wireless sensor networks have been implemented on test beds in applications for environmental monitoring, business, military, health care and so on. A pioneering work is the use of a WSN for habitat monitoring on Great Duck Island (Mainwaring *et al.*, 2002). MICA motes are adopted as sensor nodes. The MICA Weather Board provides sensors which are able to monitor changing environmental conditions with the same functionality as a traditional weather station. The MICA Weather Board includes temperature, photo resistor, barometric pressure, humidity, and passive infrared sensors. Thirty-two motes were deployed on the island for 4 weeks.

A recent experiment on environmental monitoring used a flock of micro air vehicles (MAVs) to sense weather phenomena (Allred *et al.*, 2007). Each MAV may be equipped with temperature, pressure, humidity, wind speed or direction and/or other sensors. The MAVs are able to provide detailed mapping of hurricanes, thunderstorms and tornados, and also return data to ground stations. These data are useful in improving storm track predictions and in the understanding of storm genesis and evolution. In the experiment, the MAV is designed to keep the weight and the maximum speed of the airplane under 500 g and 20 m/s, respectively. The CUPIC autopilot board is employed. It contains a CPU, pressure sensor, radio, rate gyro and GPS device that send navigation information to the CPU. The cost of the entire airplane is less than \$600. Five MAVs are employed in the experiment. An XBee Pro Zigbee class 2.4GHz radio is used to support both air-to-air and air-to-ground wireless communications. The MAVs are always operated at an altitude of less than 150 m to avoid potential conflict with larger airplanes and to maintain communication with remote control pilots.

A heterogeneous architecture for light monitoring and control was studied in Li (2006). It consists of six to eight light-sensing nodes and several actuator nodes that are connected to dimmers. A sensing network and an actuation network operate separately but are joined at a central gateway. The sensors, which match required conditions, reply to the gateway. The condition could be "sensors for which the light reading is higher than 4000" or "sensors located in the living room". The gateway then sends command messages to actuator nodes to control the lights.

The energy company BP employed motes on the Loch Rannoch, a big oil tanker, to predict failures of onboard machinery. One hundred and sixty motes were placed near some of the ship's equipment to measure vibrations in the ship's pumps, compressors, and engine as an indicator of potential failure. The system initiates an alert if unusual vibration or motion is detected. The experiment

demonstrates that motes with relatively low cost are able to help protect expensive machinery (Steel, 2005).

Wireless sensor networks could be integrated into other networks, such as Internet and 3G networks. A video surveillance system which is composed of 3G, Internet, and WSNs was studied in Tso *et al.* (2007). The system consists of five components: a sensor network with a sink, a 3G phone-controlled patrol robot, a 3G handset, a laptop connected to the Internet, and a central gateway. The sensor network is used to detect abnormal events or intruders and report the sensing data to the central gateway via the sink. The central gateway analyzes the data and, if necessary, automatically sends an SMS notification to a user. The user can dial and instruct the robot to patrol on-site at a specific location to retrieve the real-time video via a 3G phone or Laptop. In the experiment, four sensors are deployed in the corners of the inspected room. The abnormal event is artificially set to the change of light intensity.

1.18 EXPERIENCES WITH THE DEVELOPMENT OF SENSOR NETWORK SYSTEMS

In Tanenbaum *et al.* (2006), authors argue that building sensor systems is a challenging task by discussing several considered scenarios. Monitoring Mexico's borders will be slow and costly for sensors as well as humans nearby. Sensors dropped in enemy territory need to be close to each other (sensing range about 10 m), therefore having a soldier watching the same area might be much more productive. Sensors detecting fire in a forest may not be able to deliver the report because of lifetime issues. Sensors need to be lifted for increased radio range. Sensors placed to monitor certain small regions can be easily activated with false alarms (e.g., by intentionally sending animals nearby).

In Barrenetxea *et al.* (2008), an efficient and cheap out-of-the-box environmental monitoring system is described. It is a time-driven network where sensors report environmental data (wind speed and direction, soil moisture, temperature, humidity, radiation, precipitation, etc.) to a sink, which in turn relays data to a publicly available database server. A sensing station consists of a four-legged skeleton containing a sensor box (containing a sensor mote as well as primary and secondary batteries) and a solar panel. Close to 100 such stations were deployed in the largest system. The communication stack consists of application, transport, network, and MAC layers as well as a radio medium. The application layer only queries sensors and batteries, and passes data to the transport layer. The transport layer does not include any congestion avoidance mechanism. It creates data or controls packets with 4 bytes of network header (containing hop count, sender ID, cost to sink, and sequence number) and 24 bytes of application payload. The network layer passes packets to the MAC layer. The MAC layer manages the radio and sends or receives packets. It is based on a simple backoff mechanism without carrier sense. The neighborhood is managed by beacon messages initiated from the sink. Each sensor updates its cost (only hop count was used) to reach the sink. The link quality is estimated by the ability of a neighbor to

receive a data packet and to forward it. The time synchronization is achieved by a similar flooding initiated from the sink. Its frequency is decided to just offset the time-drift in sensors. The power management is resolved by duty cycling, where all nodes are synchronously sleeping and waking up, and with messages not starting before maximum time-drift following wake-ups. Routing is opportunistic, that is, a message is sent to any neighbor with a smaller hop count and decided at random at forwarding time. This ensures load balancing.

REFERENCES

- AKYILDIZ IF, KASIMOGLU IH. "Wireless sensor and actor networks: research challenges". *Ad Hoc Netw* 2004;2:351–367.
- ALLRED J, HASAN AB, PANICHSAKUL S, PISANO W, GRAY P, HUANG J, HAN R, LAWRENCE D, MOHSENI K. "SensorFlock: an airborne wireless sensor network of micro-air vehicles". *Proceedings of SenSys'07*; 2007. pp. 117–130.
- BARRENETXEA G, INGELREST F, SCHAEFER G, VETTERLI M, COUACH O, PARLANGE M. "SensorScope: out-of-the-box environmental monitoring". *The ACM/IEEE 7th International Conference on Information Processing in Sensor Networks (IPSN 2008)*; St. Louis (MO); 2008 April 22–24.
- BOSE P, MORIN P, STOJMENOVIC I, URRUTIA J. "Routing with guaranteed delivery in Ad Hoc wireless Networks". *Proceedings of 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL M99)*; 1999. pp. 48–55.
- CHEN WP, HOU JC, SHA L. "Dynamic clustering for acoustic target tracking in wireless sensor networks". *IEEE Trans Mobile Comput* 2004;3(3):258–271.
- DAM TV, LANGENDOEN K. "An adaptive energy efficient MAC protocol for WSNs". *Proceedings of ACM Sensys*; 2003.
- DAS S, LIU H, KAMATH A, NAYAK A, STOJMENOVIC I. "Localized movement control for fault tolerance of mobile robot networks". *Proceedings of the First IFIP International Conference on Wireless Sensor and Actor Networks (WSAN 2007)*; Albacete, Spain; 2007 Sept 24–26.
- DENGLER S, AWAD A, DRESSLER F. "Sensor/Actuator networks in smart homes for supporting elderly and handicapped people". *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*; 2007.
- FINN GG. "Routing and addressing problems in large metropolitan-scale internetworks". *Technical Report ISI/RR-87-180*; Information Sciences Institute (ISI); 1987.
- HEINZELMAN WR, CHANDRAKASAN A, BALAKRISHNAN H. "Energy-efficient communication protocol for microsensor networks". *Proceedings of 33rd Hawaii International Conference System Sciences*; 2000.
- HILL JL. "System Architecture for Wireless Sensor Networks" [PhD dissertation]; UC Berkeley; 2003, pp. 155.
- IEEE Standard 802. "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)"; 2003.
- INGELREST F., SIMPLOT-RYL D, STOJMENOVIC I. "Routing and broadcasting in hybrid ad hoc and sensor networks". In: Wu J: editor. *Theoretical and algorithmic aspects of sensor, Ad hoc wireless and peer-to-peer networks*. Auerbach Publications (Taylor & Francis Group); 2006. pp. 415–426.
- JOHN M, FULFORD-JONES TRF, BONATO P, WELSH M. "A wireless, low-power motion analysis sensor for stroke patient rehabilitation". *Abstract 143281, Biomedical Engineering Society (BMES) 2005 Annual Fall Meeting*; Baltimore (MD); 2005 Sept 28–Oct 1.

- KORBER HJ, WATTAR H, SCHOLL G. "Modular wireless real-time sensor/actuator network for factory automation applications". *IEEE Trans Ind Inform* 2007;3(2):111–119.
- KURUVILA J, NAYAK A, STOJMENOVIC I. "Hop count optimal position-based packet routing algorithms for ad hoc wireless networks with a realistic physical Layer". *IEEE J Sel Areas Commun* 2005;23(6):1267–1275.
- LI S-F. "Wireless sensor actuator network for light monitoring and control application". Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC 2006); 2006. pp. 974–978.
- LIU H, NAYAK A, STOJMENOVIC I. "Localized mobility control routing in robotic sensor wireless networks". Proceedings of the 3rd International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2007), LNCS 4864; Beijing, China; 2007 Dec 12–14.
- MAINWARING A, POLASTRE J, SZEWCZYK R, CULLER D, ANDERSON J. "Wireless sensor networks for habit monitoring". Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications; 2002. pp. 88–97.
- MELODIA T, POMPILI D, GUNGOR VC, AKYILDIZ IF. "Communication and coordination in wireless sensor and actor networks". *IEEE Trans Mobile Comput* 2007;6(10):1116–1129.
- MUSOLESI M, MASCOLO C. "Designing mobility models based on social network theory". *Mobile Comput Commun Rev* 2007;11(3).
- NADEEM T, AGRAWALA A. "IEEE 802.11 Fragmentation-aware energy-efficient ad-hoc routing protocols". Proceedings of the First IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS); 2004. pp. 90–103.
- NI S, TSENG Y, CHEN Y, SHEU J. "The broadcast storm problem in a mobile Ad Hoc Networks". Proceedings of ACM/IEEE MOBICOM'99; 1999. pp. 151–162.
- OLARIU S, SIMPLOT-RYL D, STOJMENOVIC I. "Localized communication and topology protocols for Ad Hoc networks: a preface to the special section". *IEEE Trans Parallel Distrib Syst* 2006;17(4):289–291.
- ONAT FA, STOJMENOVIC I, YANIKOMEROGLU H. "Generating random graphs for the simulation of wireless Ad Hoc, actuator, sensor, and internet networks". *Pervasive and mobile computing: Elsevier*, to appear.
- POLASTRE J, HILL J, CULLER D. "Versatile low power media access for wireless sensor networks". Proceedings of ACM Sensys; 2004.
- RHEE I, WARRIER A, AIA M, MIN J. "Z-MAC: a Hybrid MAC for wireless sensor networks". Proceedings of ACM Sensys; 2005.
- RODOPLU V, MENG TH. "Minimum energy mobile wireless networks". *IEEE J Sel Areas Commun* 1999;17(8):1333–1344.
- ROUNDY S, FRECHETTE LG. "Energy scavenging and non-traditional power sources for wireless sensor networks". In: STOJMENOVIC I, editor. *Handbook of sensor networks*: Wiley; 2005.
- RUIZ-IBARRA E, VILLASENOR-GONZALEZ L. "Cooperation mechanism taxonomy for wireless sensor and actor networks". IFIP Conference on Wireless Sensor and Actor Networks; Ottawa; 2008 Jul 14–15.
- STEEL D. "Smart dust". ISRC Technol Brief; 2005.
- STOJMENOVIC I. *Handbook of sensor networks: algorithms and applications*: Wiley; 2005.
- STOJMENOVIC I. "Energy conservation in sensor and sensor-actuator networks". In: WU S-L, TSENG Y-C, editors. *Wireless Ad hoc networking: personal-area, local-area, and sensory-area networks*: Auerbach Publications, T&F; 2007. pp. 107–133. Chapter. 4.
- STOJMENOVIC I, LIN X. "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks". *IEEE Trans Parallel Distrib Syst* 2001a;12(10):1023–1032.
- STOJMENOVIC I, LIN X. "Power-aware localized routing in wireless networks". *IEEE Trans Parallel Distrib Syst* 2001b;12(10):1–12.

32 Chapter 1 Applications, Models, Problems, and Solution Strategies

- STOJMENOVIC I, NAYAK A, KURUVILA J. "Design guideline for routing protocols in Ad Hoc and sensor networks with a realistic physical layer". *IEEE Commun Mag* 2005a;43(3):101–106.
- STOJMENOVIC I, NAYAK A, KURUVILA J, OVALLE-MARTINEZ F, VILLANUEVA-PENA E. "Physical layer impact on the design and performance of routing and broadcasting protocols in Ad Hoc and sensor networks". *Comput Commun* 2005b;28(10):1138–1151.
- TANENBAUM AS, GAMAGE C, CRISPO B. "Taking sensor networks from the lab to the jungle". *IEEE Comput* 2006:98–100.
- THRUN S, DIEL M, HAHNEL D. "Scan alignment and 3-D surface modeling with a helicopter platform". *Proceedings of International Conference on Field and Service Robotic (FSR'03)*; 2003.
- TOH CK. "Maximum battery life routing to support ubiquitous mobile computing in wireless Ad Hoc networks". *IEEE Commun Mag* 2001;39(6).
- TSO FP, ZHANG L, JIA W. "Video surveillance patrol robot system in 3G, internet and sensor networks". *Proceedings of SenSys'07*; 2007. pp. 395–396.
- TULLY A. "Pervasive tagging, sensors, and data collection". *Foresight Intelligent Infrastructure Systems Project*; 2005.
- WARK T, CROSSMAN C, HU W, GUO Y, VALENCIA P, SIKKA P, CORKE PI, LEE C, HENSHALL J, PRAYAGA K, O'GRADY J, REED M, FISHER A. "The design and evaluation of a mobile sensor/actuator network for autonomous animal control". *Proceedings of 6th International Conference on Information Processing in Sensor Networks (IPSN 2007)*; Cambridge; 2007. pp. 206–215.
- XIA F, TIAN YC, LI Y, SUN Y. "Wireless sensor/actuator network design for mobile control applications". *Sensors* 2007;7:2157–2173.
- YE W, HEIDEMANN J, ESTRIN D. "Medium access control with coordinated adaptive sleeping for WSNs". *IEEE/ACM Trans Network* 2004;12(3):493–506.
- ZigBee Alliance. "Network specification", Version 1.0; 2004.