η represents the amplifier inefficiency factor, ω is the free-space path loss, d is the distance, and n is the environmental factor. Based on an environmental condition, n can be a number between 2 and 4, and η specifies the inefficiency of the transmitter when generating maximum power ωd^n at the antenna. Clearly, the distance-dependent portion of total energy consumption depends on the real-world transceiver parameters, θ, η, and the path attenuation ωd^n. If the value of θ overshadows $\eta\omega d^n$, the reduction in the transmission distance through the use of multihop communication is not effective.

In theory, the maximum efficiency of a power amplifier is 48.4 percent. However, practical implementations show that the power-amplifier efficiency is less than 40 percent. Therefore, θ is calculated assuming that $\eta = 1/0.4 = 2.5$. Using Equation (20.1), we can express the energy consumption of a transmitter and a receiver, E_T and E_R, respectively, by

$$E_T = \theta_T + \eta\omega d^n \tag{20.2}$$

and

$$E_R = \theta_R, \tag{20.3}$$

where θ_T and θ_R are the distance-dependent terms for the transmitter and the receiver, respectively. Although maximum output power and total power consumption are provided in the manufacturer's data sheet, θ can be calculated the following formula:

$$\theta = \theta_{TX} + \theta_{RX} = (E_T - \eta\omega d^n) + E_R. \tag{20.4}$$

Example. Table 20.1 shows values of E_T and E_R based on a manufacturer's data sheet and θ and $\eta\omega d^n$ calculated for a selected chipset. Although path-attenuation energy increases exponentially by the transmission distance, the data illustrates that the static power consumption, θ, dominates the path loss. Clearly, this causes total power consumption to remain constant as the transmission distance increases. Standards 802.11a, 802.11b, and 802.11g have multirate capabilities. Although, sensor nodes in general generate data in low rates, they can transmit the information using wireless high-speed modulation and techniques.

Table 20.2 shows the expected data rate for the 802.11g wireless technology. Although exploiting the multirate capabilities of wireless standards has never been proposed for sensor networks, this technique can decrease the transmission energy for smaller distances by switching to higher data rates and keeping the transceiver on for a

Table 20.1 Energy consumption parameters

IEEE Standard	Max. Output Power, ωd^n (dBm)	Total Power Consumption (W)	θ (W)	$\eta \times \omega d^n$ (W)
802.11a	+14	1.85 (E_{TX}) 1.20 (E_{RX})	2.987	0.0625
802.11b	+21	1.75 (E_{TX}) 1.29 (E_{RX})	2.727	0.3125
802.11g	+14	1.82 (E_{TX}) 1.40 (E_{RX})	3.157	0.0625

Table 20.2 Expected data rate of IEEE 802.11g technology

Rate (Mb/s)	Maximum Range	Rate (Mb/s)	Maximum Range
1	100.00 m	18	51.00 m
2	76.50 m	24	41.25 m
6	64.50 m	36	36.00 m
9	57.00 m	48	23.10 m
12	54.00 m	54	18.75 m

shorter period of time. In this case, the energy in terms of Joule/bit reduces discretely as transmission distance shrinks:

$$E = \frac{1}{R}\left(\theta + \eta\omega d^n\right), \tag{20.5}$$

where R is the rate in bits/sec. Figure 20.4 shows energy consumption using 802.11g technology at the constant rate of 1 Mb/s and the same technology with the multirate extension. Owing to large values of θ compared to the maximum output power, single-rate communication energy consumption remains constant as the transmission distance increases, whereas the communication energy consumption for multirate transmission decreases for shorter transmission ranges. However, this scenario does not follow the model of ωd^n. Meanwhile, the multirate communication necessitates the presence of a robust rate-selection protocol.

Multi-Hop Communication Efficiency

Considering the impact of real-world radio parameters and multirate communication, we should reevaluate the effectiveness of multihop communications. Since a multi-rate communication reduces energy consumption for shorter distances by switching to

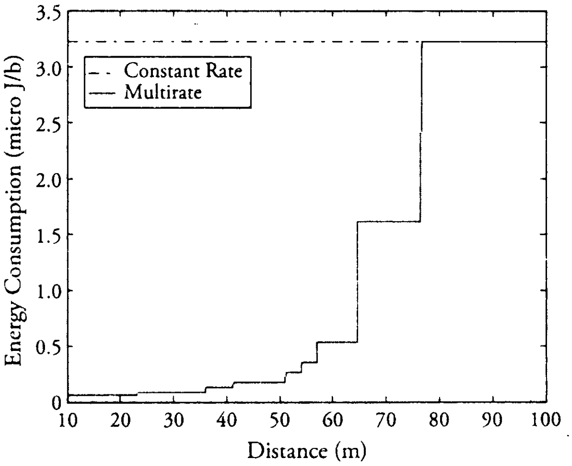

Figure 20.4 Energy consumption versus transmission distance for single-rate and multirate communication using 802.11g technology

higher data rates, multihop communication can conserve energy. The traditional objective of multihop communication is to divide the transmission distance into a number of hops, m, and to relatively conserve energy, considering Equation (20.3), by means of

$$E = m\left(\theta + \omega\left(\frac{d}{m}\right)^{n}\right). \tag{20.6}$$

However, if the division of transmission distance happens when the maximum range is less than 18.75 m for standard 802.11g, the data rate remains constant, and total energy consumption multiplies by the number of hops. Since sensor networks deal with two- or even three-dimensional spaces, multihop efficiency depends on the network scale and density.

Example. Figure 20.5 shows an organization in which sensor nodes A, B, C, D, and E are placed d meters apart and tend to send their data packets to the *cluster head* (CH). Note that d is an application-dependent parameter and can be chosen based on the sensor's characteristics. Assume that standard 802.11g technology is used in an environment in which sensors are placed on average no more than 10 meters apart. Compare nodes' energy consumptions, using Figure 20.5.

Solution. With the choice of 10 meters for d in the 802.11g charts, if node B tries to use node A as a relay node and then sends data to the cluster head, the total energy

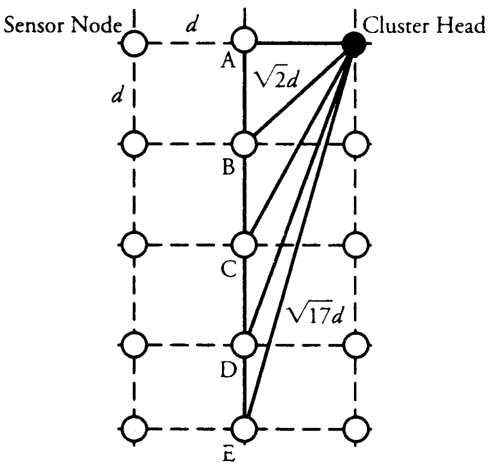

Figure 20.5 Cluster-head distances from sensor nodes A, B, C, D, and E in a two-dimensional model

of the chosen two-hop path is larger than the direct-transmission energy obtained as

$$E(\sqrt{2}d) = 0.0517 < E(d) + E(d) = 0.1034. \qquad (20.7)$$

Also, for nodes C and D, there is no multihop path that can lead to better energy consumption than the direct communication path:

$$E(\sqrt{5}d) = 0.0581 < E(\sqrt{2}d) + E(d) = 0.1034 \qquad (20.8)$$

and

$$E(\sqrt{10}d) = 0.0775 < E(\sqrt{5}d) + E(d) = 0.1098. \qquad (20.9)$$

But if node E first sends the data to the intermediate node D, total energy consumption will be less than the direct communication path:

$$E(\sqrt{17}d) = E(41.23) = 0.1789 > E(\sqrt{10}d) + E(d) = 0.1292 \quad (20.10)$$

Node E is 41.23 meters away from the cluster head. This shows that for nodes more than 41.23 meters apart, direct transmission is no longer the best-possible communication method.

Example. Continuing the previous example using 802.11g technology, set up an environment representing one cluster. The dimension of the field is 50 m × 50 m, and

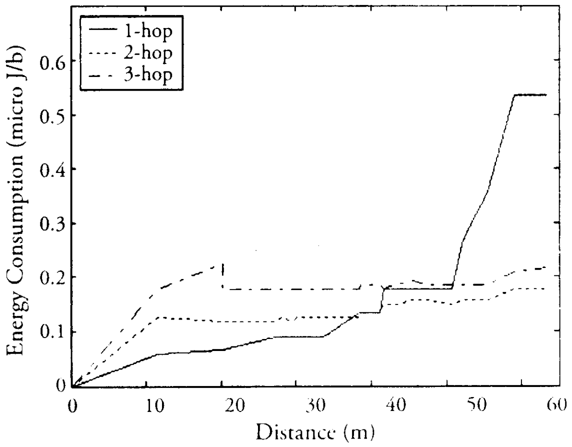

Figure 20.6 Communication energy versus distance from cluster head for 802.11g technology

25 nodes are randomly dispersed in the field. Compare the energy consumption of direct and multihop communication inside the cluster.

Solution. At this point, we assume that the cluster head is chosen randomly among the sensors. (The details of cluster-head selection algorithms explained in Section 20.3.) Figure 20.6 shows the energy consumption of a direct, minimum-energy two-hop and minimum-energy three-hop path, based on the distance between nodes in the cluster and the cluster head. For 802.11g technology, the direct transmission is the optimum choice for ranges less than 37 meters, which is almost the same as the result from analytical calculations (41 m). However, for ranges greater than 37 meters, the minimum-energy two-hop path can lead to significantly lower energy consumption.

20.3 Clustering Protocols

Clustering protocols specify the topology of the hierarchical nonoverlapping *clusters* of sensor nodes. A robust clustering technique is essential for self-organizing sensor networks. An efficient clustering protocol ensures the creation of clusters with almost the same radius and cluster heads that are best positioned in the clusters. Since every node in a clustered network is connected to a cluster head, route discovery among cluster heads is sufficient to establish a feasible route in the network. For a large sensor network, clustering can simplify multihop route discovery and limit the number of transmissions compared to a flat, nonclustered network.

20.3.1 Classification of Clustering Protocols

Clustering techniques can be either *centralized* or *decentralized*. Centralized clustering algorithms require each sensor node to send its individual information, such as energy level and geographical position, to the central base station. Based on a predefined algorithm, a base station calculates the number of clusters, their sizes, and the cluster heads' positions and then provides each node with its newly assigned duty.

Given the assumption that sensor networks might consist of thousands of nodes, it is impractical, if not impossible, for a base station to collect information about every node in the network prior to route setup. Therefore, centralized clustering is not an option for large sensor networks. Since a sensor node begins a clustering process without any knowledge about its location relative to the corresponding base station, a clustering algorithm should be able to form clusters without the help of the base station and knowledge of node positioning. Although location-finder devices can also be deployed to perform this task, they are often either costly or add too much overhead on the network.

Decentralized clustering techniques create clusters without the help of any centralized base station. An energy-efficient and hierarchical clustering algorithm can be such a way whereby each sensor node becomes a cluster head with a probability of p and advertises its candidacy to nodes that are no more than k hops away from the cluster head. Given the limited transmission range of wireless sensor nodes, a hierarchical structure with an arbitrary number of levels has its limitations. As the number of hierarchical levels grows, the distance between upper-level cluster heads may increase to the point that they are no longer able to communicate with one another. The *Low-Energy Adaptive Clustering Hierarchy* (LEACH) algorithm and the *Decentralized Energy-Efficient Cluster Propagation* (DEEP) protocol are two examples of the decentralized clustering protocols and are explained next.

20.3.2 LEACH Clustering Protocol

The *Low-Energy Adaptive Clustering Hierarchy* (LEACH) protocol is a decentralized clustering algorithm that does not offer a complete energy-optimization solution, as it has no strategy for specifying cluster-head positioning and distribution. LEACH is an application-specific protocol architecture that aims to prolong network lifetime by periodic reclustering and change of the network topology.

LEACH is divided into *rounds* consisting of a clustering phase and a steady-state phase for data collection. At the start of each round, a sensor node randomly chooses a number between 0 and 1 and then compares this number to a calculated threshold called $T(n)$. If $T(n)$ is larger than the chosen number, the node becomes a cluster head for the current round. The value $T(n)$ is calculated using the following formula:

$$T(n) = \begin{cases} \frac{p}{1-p(r \operatorname{mod}(1/p))} & \text{for } n \in G \\ 0 & \text{otherwise} \end{cases}, \tag{20.11}$$

where p is the ratio of the total number of cluster heads to the total number of nodes, r is the number of rounds, and G is a set of nodes that have not been chosen as cluster heads for the last $1/p$ rounds. For the first round (r=0), $T(n)$ is equal to p, and nodes have an equal chance to become cluster head. As r gets closer to $1/p$, $T(n)$ increases, and nodes that have not been selected as cluster head in the last $1/p$ rounds have more chance to become cluster head. After $1/p - 1$ rounds, $T(n)$ is equal to 1, meaning that all the remaining nodes have been selected as cluster head. Thus, after $1/p$ rounds, all the nodes have had a chance to become a cluster head once. Since being the cluster head puts a substantial burden on the sensor nodes, this ensures that the network has no overloaded node that runs out of energy sooner than the others.

After cluster heads are self-selected, they start to advertise their candidacy to the other sensor nodes. When it receives advertisements from more than one cluster-head candidate, a sensor node starts to make a decision about its corresponding cluster head. Each node listens to the advertisement signals and chooses the candidate whose associated signal is received with higher power. This ensures that each sensor node chooses the closest candidate as cluster head. The LEACH algorithm is distributed, as it can be accomplished by local computation and communication at each node, rather than the transmission of all the nodes' energy level and geographical position to a centralized point. However, cluster heads are chosen randomly, and there is no optimization in terms of energy consumption.

20.3.3 DEEP Clustering Protocol

The *Decentralized Energy-Efficient Cluster Propagation* (DEEP) protocol that establishes clusters with uniformly distributed cluster heads. This protocol balances the load among all the cluster heads by keeping the clusters' radii fairly equal. This protocol is completely decentralized, and there is no need for any location-finder device or hardware. The protocol starts with an initial cluster head and forms new cluster-head candidates gradually by controlling the relative distance between a pair of cluster heads and the circular radius of each cluster. Owing to the balanced load among cluster heads, periodic reclustering is not necessary, and operational expenses caused by frequent reclustering are therefore eliminated.

An efficient path-selection algorithm for nodes that are placed more than ℓ meters away from a cluster head is necessary in order to find the optimum two-hop or three-hop path. Although direct transmission to a cluster head can eliminate the overhead created by the route set-up packets, its efficiency is questionable, owing to the limited transmission range. In order to avoid the frequent control signal transmission and extra power consumption associated with that, a cluster head can be placed at the center of the cluster, with sensor nodes positioned closer than ℓ meters around it. In this case, cluster members can send the data packets directly to the cluster head without the need for any route set-up protocol, while efficiency has already been achieved through the choice of cluster shape and cluster size.

In order to explain the details of this algorithm, control signals and protocol parameters need to be introduced:

- Control signals: (1) cluster-head declaration signal or (2) cluster-head exploration signal
- Membership search signal with control parameters: declaration range (d_r), exploration range (d_{r1}, d_{r2}), minimum number of members (m_n), E_{rc1}, and E_{rc2}.

Protocol-control parameters are application-specific choices and can be defined prior to network deployment. DEEP forms clusters by starting with an initial cluster head that can be chosen prior to network deployment. This initial cluster head starts the cluster set-up phase by propagating cluster-head declaration signals within the range of d_r. This means that the cluster-head candidate chooses an appropriate data rate and signal output power so that it can reach nodes that are less than d_r away from the sender.

At this point, sensor nodes that receive the declaration signal accept the corresponding cluster head as a leader. They can then estimate their relative distance to the candidate by looking at the received signal's energy level. Once they know the relative distance to the cluster head, they can conserve energy by adjusting the transmission speed to the appropriate value and switching to sleep mode. Now, the initial cluster-head candidate propagates the cluster-head exploration signal within the range of d_{r2}, as shown in Figure 20.7. All the sensor nodes in this range can listen to the exploration signal, but only nodes that have never played the role of a cluster head and verify the following inequality are chosen as new candidates:

$$E_{rc1} < E_r < E_{rc2}, \tag{20.12}$$

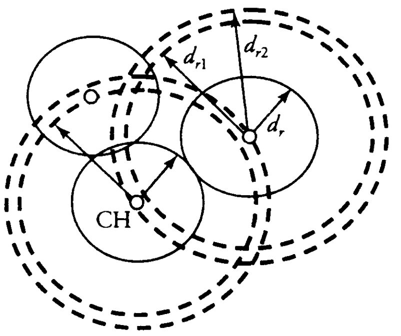

Figure 20.7 Initial cluster head starts the advertisement process. New cluster-head candidates send the exploration signal within the range of d_{r2} to continue the process of cluster establishment.

where E_r is the received signal energy. Note that E_{rc1} and E_{rc2} are fixed protocol parameters that can be precalculated and stored in the sensor-node memory, using the following formula:

$$E_{rc1} = P_{out} - \omega d_{r1}^n \tag{20.13}$$

and

$$E_{rc2} = P_{out} - \omega d_{r2}^n, \tag{20.14}$$

where P_{out} is the constant output power of the cluster-head exploration signal, and ω and n are parameters that can be determined based on the environmental conditions of the deployment area. This way, any of these nodes can consider itself a candidate. This ensures that new cluster-head candidates are positioned between d_{r1} and d_{r2}, away from the initial cluster head.

After a new cluster-head candidate is assigned, it sends a declaration signal within the range of d_r to find new cluster members. If two candidates can hear each other's declaration signal, they are too close to each other to be considered cluster-head candidates. Therefore, one of them is eliminated through a negotiation phase. Whenever it receives a declaration signal, a cluster head informs the sender of the message, using an acknowledgment message. A cluster head that receives the acknowledgment sends a dissolution message and informs all nodes within the range of d_r about its elimination. A node that receives a declaration signal from more than one candidate chooses the candidate whose associated signal is received with a higher power.

At this point, all confirmed cluster heads propagate exploration signals and search for new cluster-head candidates. Nodes that have already been chosen as cluster head or member ignore the cluster-head exploration or declaration signals. Therefore, this advertisement process terminates automatically when all the nodes in the field belong to a cluster. At this point, the algorithm might have produced some clusters with a very small number of members. Therefore, a cluster whose total number of members is smaller than the minimum number of members, m_n, is dissolved, and all its members, including its cluster head, initiate a *membership-search signal*.

After this process is completed, nodes listen to the responses from the local cluster heads and choose the closest cluster head, based on the received signal power. At the end, if the timeout has been reached, a sensor node that has not received any control signal sends a *membership-search signal* and chooses the closest cluster head as leader. The following algorithm summarizes the core segment of the DEEP protocol.

Begin DEEP Clustering Algorithm

1. **Initialize:** Initial cluster head finds cluster members by sending "cluster-head declaration."
2. Initial cluster head finds new cluster-head candidates by sending "cluster-head exploration signal."
3. **Repeat:** Cluster-head candidates that are placed on the (d_{r1}, d_{r2}) ring find cluster members.
4. Nodes that receive more than one cluster-head declaration choose the closest cluster head, based on the received signal energy.
5. Cluster-head candidates that receive a cluster-head declaration signal negotiate with the sender, and one of them gets eliminated.
6. Confirmed cluster heads send "cluster-head exploration" signals to find new cluster-head candidates (Go to step 4).
7. **Finalize:** If the number of members in a cluster is less than m_n, all the members find new clusters by sending the membership-search signal.
8. At the end, a node that has not received any control signal sends the membership-search signal. ■

DEEP has several advantages over other clustering protocols. With DEEP, a sensor node can either select itself as a cluster head by receiving a cluster-head exploration signal or join a cluster by receiving a cluster-head declaration signal. After the execution of the protocol, all the sensor nodes are covered and belong to only one cluster. This clearly shows that this protocol is completely decentralized. In addition, for the execution of

DEEP, there is no need for any location-finder hardware, such as the *global positioning system* (GPS) or a position-estimation protocol that puts extra overhead on sensor nodes.

DEEP can control a cluster-head distribution across the sensor network through protocol-execution methodologies. For example, cluster-head candidates should receive the cluster-head exploration signal with a certain amount of energy; if they can hear the declaration signal of each other, one of the candidates is eliminated. Communication cost is low through proper selection of protocol parameters, such as declaration range, exploration range, and minimum number of members.

With DEEP, *intracluster* communication is controlled by cluster heads, and nodes transmit their data directly to cluster heads. Therefore, no additional control signal is associated with route selection and maintenance inside the cluster. Also, owing to the uniform distribution of cluster heads, communication cost of a direct transmission path between a pair of neighboring cluster heads is almost identical across the sensor field. This is one of the most important protocol characteristics contributing to convenient deployment of an *intercluster* routing protocol.

20.3.4 Reclustering

In order to prevent overutilization of some sensor nodes, clustering technique should ensure that the cluster-head responsibility rotates among all sensor nodes. To achieve this, reclustering is performed periodically in LEACH. However, every round of reclustering requires several control-signal exchanges among self-elected cluster heads and sensor nodes. The reclustering process in DEEP is based on one small shift in the initial cluster head. When the current period of cluster setting is finished, the current initial CH chooses the nearest node that has never acted as an initial cluster head. This newly chosen initial cluster head starts the clustering process and creates a totally different cluster-head constellation.

20.4 Routing Protocols

After clusters with well-distributed cluster heads have been established in a network, energy-conscious routing is essential in order to set communication routes among cluster heads in a two-level hierarchical system. Similar to computer networks, routing protocols in sensor networks can be classified as either *intracluster* or *intercluster*. This section looks at both categories.

The fundamental concept behind them is much the same as the concept behind intradomain and interdomain routings (see Chapter 7). Assuming that every node in a cluster can act as a relay node, there could be a large number of possible routes from a source to a sink. Because of the limited transmission range associated with low-power wireless technologies cluster-head packets cannot reach the base station unless other cluster heads act as relay nodes. Two major approaches can be used for routing and selecting the best path in a sensor network, as follows:

1. *Centralized routing*, whereby the routing decision is made by a single command center
2. *Distributed routing*, whereby the routing decision is made in a distributed fashion by several entities

Distributed routing algorithms are further classified as *proactive* or *reactive*. With proactive routing algorithms, such as link-state routing and distance-vector routing, nodes keep a routing table that contains next-hop information to every node in the network. Reactive routing protocols set a route to the desirable destination only when it is needed. Note that none of the ad hoc network protocols explained earlier consider energy consumption.

Another group of on-demand reactive routing protocols address the exclusive issues of wireless sensor network. For example, *directed diffusion* introduces a concept of "interest" propagation whenever a node wants to send data or a source needs to ask for it. With this type of protocol, flooding the network with interest signals establishes a path from a sink to every possible source (spanning tree).

20.4.1 Intracluster Routing Protocols

A routing algorithm within a cluster can be either *direct* or *multihop*. In a direct routing algorithm, the cluster head as the destination for all cluster nodes is located in the center of the cluster, so all nodes can communicate with the cluster head directly, as shown in Figure 20.8. Note that in this figure, two nodes cannot reach the destination, as they are located far from it. The number shown in each node indicates the level of energy the corresponding node has.

In a multihop routing algorithm, a node can face multiple hops in order to reach the destination. If a multihop algorithm is used for the centralized clustering procedure, the algorithm aims to choose the appropriate next neighbor for each node, using a central command node. Typically, a central command node collects the information about direct paths' costs and geographical positions of the nodes and finds the best path.

Figure 20.9 shows a routing implementation. Sensor nodes are usually scattered in the field. A packet from a node is routed to a neighboring node that exhibits the

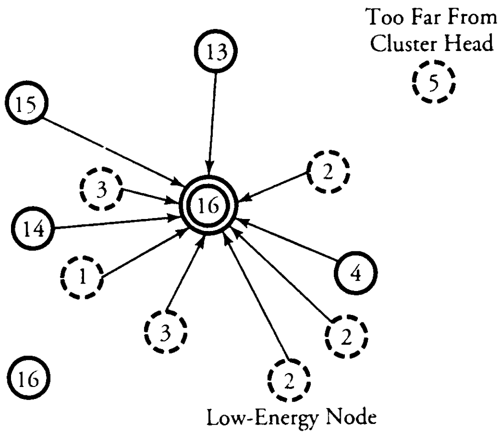

Figure 20.8 Direct routing in a cluster. The number associated with each node indicates a normalized value of the remaining energy in that node.

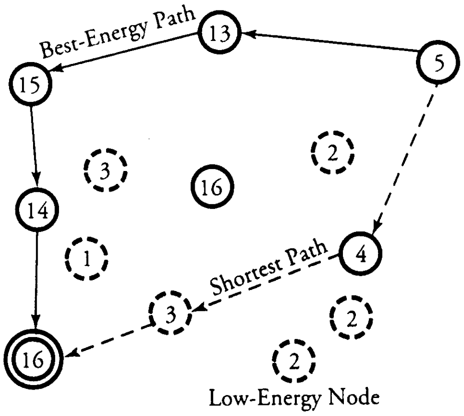

Figure 20.9 Multihop routing in a cluster in which the number associated with each node indicates a normalized value of the remaining energy in that node

highest amount of energy. The energy is an indication of the node's battery level. The number associated with each node indicates a normalized value of the remaining energy in that node. Figure 20.9 shows two paths from a node to a cluster-head node. One path involves the shortest distance in terms of hop counts; the other one uses the highest-energy route. The challenge here is to find the best path that suits the rapid and secure deployment of data. Data is routed to the cluster head as shown in Figure 20.1, where the cluster head may communicate with the base station via radio frequencies.

The least-cost algorithm and the best-energy path can be modeled and compared for a network to provide a behavioral benchmark. The model can determine all possible paths available between a given source and destination. The energy of each node and hence all possible least-cost paths are computed regularly by cluster heads and propagated to all cluster nodes for database updating. During the phase of path finding to a cluster head, the routing algorithm accepts a failing (low-energy) node and finds the least-cost path, taking the failing node into account. The inputs for route process then include source node, destination node, failing nodes, and all other nodes.

20.4.2 Intercluster Routing Protocols

Intercluster protocols are not typically different from the multihop ones for intradomain cases. Interdomain protocols are available for

- Intercluster energy conscious routing (ICR)
- Energy-aware routing (EAR)
- Direct diffusion

ICR uses interest flooding similar to directed diffusion and EAR to establish routes between the base station and sensor nodes but differs from EAR and directed diffusion in some aspects.

Intercluster Energy-Conscious Routing (ICR)

ICR is a destination-initiated reactive routing protocol. This means that a destination, local base station (LBS), initiates an explicit route-discovery phase, which includes the propagation of an *interest* signal that floods throughout the network and establishes energy-efficient routes. Based on the application, which can be either periodic data collection or event driven, the interest signal can include the *type* and the *period* of the desired data shown in Figure 20.10. For an application requiring information from specific locations, the interest signal also includes the position of the required information.

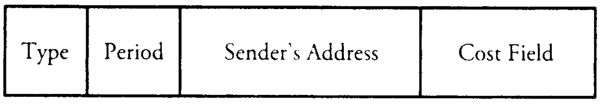

Figure 20.10 Interest-signal structure in a packet

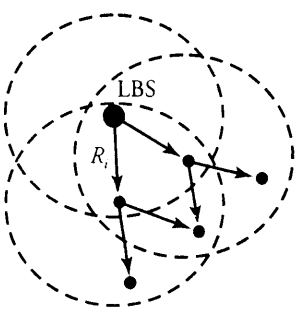

Figure 20.11 LBS starts route discovery by generating interest signals.

If the LBS requires some periodic data collection, it sets the *period* in which nodes send the specific *type* of information. Monitoring and surveillance applications are examples for the data-collection paradigm. If it requires sensor nodes to detect one specific event, an LBS includes the *type* of the event in the interest signal. Following the route-discovery phase, sensor nodes switch to sleep mode and wait for the specific event. In case of event detection, non-cluster-head nodes send the data directly to the associated cluster head, which uses the previously established route to send the information back to the LBS. In short, ICR occurs in two phases: *route discovery* and *data acquisition*.

In *route discovery*, the local base station initiates route discovery by sending an interest signal within the range of R_i. The value of R_i should be both high enough to keep the cluster-head network connected and low enough to prevent unnecessary energy consumption and interest generation. Owing to even distribution of cluster heads achieved by a clustering protocol, R_i can be chosen slightly bigger than the average distance between a pair of adjacent cluster heads. The LBS should adjust its output power and data rate of the interest signal to limit its transmission range to R_i. Also, the cost field is set to zero before interest propagation starts.

Since the distance between a pair of neighboring cluster heads is approximately the same across the network, the communication energy consumption associated with two distinct adjacent cluster heads is also the same. Therefore, the cost, or weight, of a multihop path is defined exclusively by the number of hops. In addition, the remaining energy in the cluster heads along the path affects the route-selection decision. The total-cost function C is defined as

$$C = \alpha h + \beta \sum_i \frac{B_M}{B_{ri}}, \tag{20.15}$$

where h is the hop number and B_{ri} represents the remaining energy in the battery of node i, B_M shows the maximum battery capacity of a sensor node, and α and β are normalization factors. The second part of the cost field favors the paths that include nodes with higher energy. To update the cost field, each intermediate cluster head calculates the inverse of its remaining battery power plus 1 (increment in the number of hops) and adds the outcome to the existing cost value.

Each intermediate cluster head that receives the interest signal saves the interest in its memory, including the address of the nodes that sent the message. Then the node should update the cost field of the outgoing interest signal and send it within the range of R_i. All the cluster heads within this range around the sender can hear the incoming signal. If it receives an interest signal that currently exists in memory but the sender's address is different, a cluster head compares the cost field of the received signal with the cost field of the previously saved message. If the incoming interest signal includes a cost field smaller than the previously saved message, the node replaces the old interest entry, updates the cost field, and propagates the packet, since the new signal represents a shorter, or more energy-efficient, path. If the new interest signal represents a path with a higher number of hops, the node should destroy the packet.

The *data-acquisition phase* occurs after each cluster head collects the requested information from sensor nodes and compresses it into a packet with fixed length, searches for the neighbor's address in memory, and relays the packet to that neighbor. In order to reduce the diffusion of spare data bits in the network, relay nodes can receive the data packets, each of length L, from N nodes and aggregate them into one single packet of length L. This reduces the number of data bits forwarded by the relay node from NL to L. To enable data aggregation during the data-collection period, cluster heads that are closer to the base station—that is, the cost field of the saved interest message includes fewer hops—should wait for their neighbors to send their data packets and then compress the incoming information with their own data and send the packet with the fixed length to the relay neighbor.

Comparison of ICR and EAR

ICR is different from EAR in two aspects. In EAR, sensor nodes save and propagate most of the incoming interest signals and eliminate only the ones with a very high cost field. However, in ICR, every time that the cost field of the incoming interest message is higher than the previously saved one, the packet gets destroyed. This puts a limit on the generation of interest messages.

In EAR, in order to ensure that the optimal path does not get depleted and that the network degrades evenly, multiple paths are found between a source and a destination. Each node has to wait for all the interest signals to come and then calculates the average cost between itself and the destination. Based on the average cost, each path is assigned a probability of being chosen. Depending on the probability, each time one of the paths is chosen, ICR assumes that data aggregation is executed among cluster heads, and no packet moves along the chosen path independently. This means that during the data-collection period, each cluster head aggregates the data from its N adjacent cluster heads and has to forward only one compressed packet rather than N distinct packets. After the execution of the routing protocol, a spanning tree is established that is rooted in the base station and connects all the cluster heads to the base station. Therefore, only the least-cost, or the optimum, path is a final, established route for each cluster head. This way, the degradation of the optimal path for each packet is prevented.

20.5 Case Study: Simulation of a Sensor Network

This section presents a case study that shows the implementation of DEEP and ICR for a wireless sensor network spread over an area. The network is used for monitoring and protecting the area. The basic objective is to deploy a large number of low-cost and self-powered sensor nodes, each of which acquires and processes data from a hazardous event and alerts a base station to take necessary action. In this scenario, 3,000 sensor nodes are randomly distributed in a field of 550 m × 550 m. Therefore, the density of sensor nodes is about one per 10 m × 10 m area, which is the maximum detection range for the hazard sensors.

MAC assigns a unique channel for every node and prevents possible collisions. With this assumption, we extracted the MAC layer from our simulations, and data packets were sent directly from the network layer of one node to the network layer of the neighbor. We simulated the DEEP algorithm, using parameters d_r, d_{r1}, d_{r2}, and m, and put the initial cluster head at the center of the field.

20.5.1 Cluster-Head Constellation and Distribution of Load

Figure 20.12 shows the result of the simulation with parameters $d_r = 30$ m, $d_{r2} = 80$ m, $d_{r1} = 78$ m, $m = 14$. Based on the results obtained from Section 20.2, the distance of 30 meters is an initial choice for d_r. In order to avoid overlapping between clusters, the value of d_{r1} and d_{r2} should be more than twice the value of d_r. Since the average distance between sensor nodes in this application is 10 m, 80 m is a fair choice for d_{r2}. The width of the (d_{r1},d_{r2}) ring should be large enough to accommodate new cluster-head candidates and small enough to avoid cluster-head candidates that are too close to each other. We chose an initial value 2 m for the ring width.

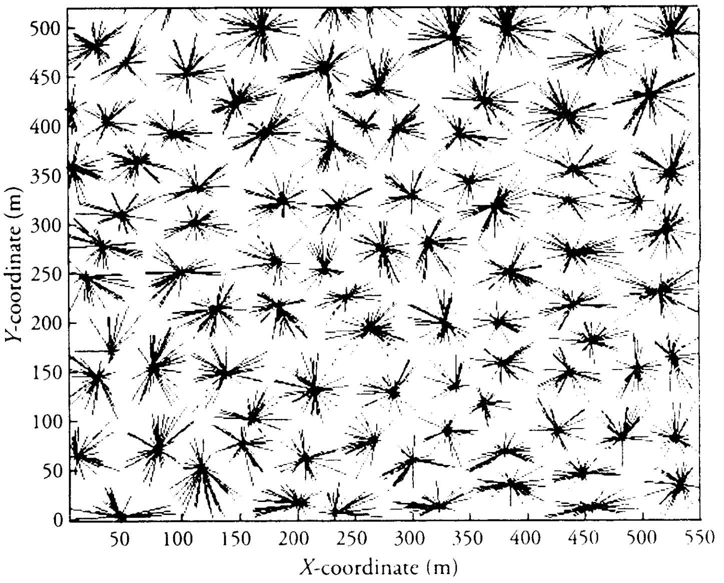

Figure 20.12 Simulation results on distributed clusters whose sensor nodes are directly connected to their associated cluster heads. The initial cluster head is put in the center of the sensor field, the simulation starts by advertising its candidacy, and cluster heads are gradually dispersed across the network.

In order to balance the load among cluster heads, DEEP controls the cluster-head distribution rather than the number of cluster members. Although cluster heads that manage more members should execute more signal processing for the sake of data aggregation, digital processing consumes much less energy than wireless transmission, and no overutilized cluster head is using this protocol.

Figure 20.13 demonstrates the cluster-head distribution achieved using LEACH and DEEP. Because of the random selection of cluster heads in LEACH, some of the cluster heads are too close to each other; others, too far. This type of cluster-head selection causes a lot of burden on some cluster heads and quickly drains their batteries. It can be shown that compared with LEACH, DEEP is capable of minimizing energy consumption associated with reclustering overheads more efficiently by reducing the number of necessary rounds.

20.5.2 Optimum Percentage of Cluster Heads

In order to determine the optimum cluster-head density and compare the performance of the routing protocol on both DEEP and LEACH, we used a 1,600-node network.

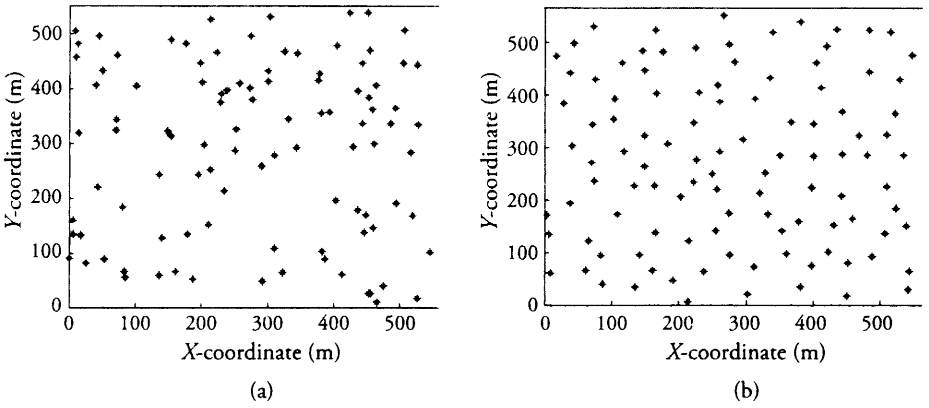

Figure 20.13 Comparison of cluster-head constellation between (a) LEACH and (b) DEEP. DEEP generates well-distributed cluster heads across the network.

Nodes were randomly distributed in a field 400 m × 400 m. In this scenario, sensor nodes send the information directly to their associated cluster head. Each cluster head compresses the data and waits for neighbor cluster-heads' data packets. Then, the cluster head compresses all the received data packets into a packet with fixed length and sends it to the relay neighbor. The relay neighbor address has been saved in node memory through the propagation of the interest signal. In-network data aggregation performed by cluster heads helps to reduce the amount of data dispersed in the network.

20.6 Other Related Technologies

Other sensor-network based technologies use low-power nodes. The one discussed in this section is the *ZigBee technology*.

20.6.1 Zigbee Technology and IEEE 802.15.4

The *ZigBee technology* is a communication standard that provides a short-range low-cost networking capability that allows low-cost devices to quickly transmit small amounts of data, such as temperature readings for thermostats, on/off requests for light switches, or keystrokes for a wireless keyboard. Other ZigBee applications are in professional installation kits for lighting controls, heating, ventilation, air conditioning, and security. Even the Bluetooth short-range wireless technology found in laptops and cellphones lacks the affordability, power savings, and mesh-networking capabilities of ZigBee.

ZigBee comes from higher-layer enhancements by a multivendor consortium called the Zigbee Alliance. IEEE standard 802.15.4/ZigBee specifies the MAC and physical layers. The 802.15.4 standard specifies 128-bit AES encryption; ZigBee specifies how to handle encryption key exchange. The 802.15.4/ZigBee networks run in the unlicensed frequencies, 900 MHz and 2.4 GHz band, based on a packet radio standard and support many cordless telephones, allowing data to be sent over distances up to 20 meters.

ZigBee devices, typically battery powered, can transmit information much farther than 20 meters, because each device within listening distance passes the message along to any other device within range. Only the intended device acts on the message. By instructing nodes to wake up only for those split-second intervals when they are needed, ZigBee device batteries might last for years. Although this technology is targeting for manufacturing, health care, shipping, and homeland defense, the ZigBee Alliance is initially keeping its focus small.

20.7 Summary

This last chapter focused on wireless sensor networks. Some applications of sensor networks are target tracking, environmental monitoring, system control, and chemical or biological detection.

The protocol stack for sensor networks concerned with power factor. The sensor network protocol stack combines two features: power-efficiency and least-cost-path routing. Thus, the protocol architecture integrates networking protocols and power efficiency through the wireless medium and promotes cooperative efforts among sensor nodes. The protocol stack consists of the physical, data-link, network, transport, and application layers, power-management, mobility-management, and task-management planes. The internal structure of an intelligent sensor node consists of three units for sensing, processing, and storage, respectively, and communications capabilities.

An energy model for a transceiver of a node can be developed. The energy consumption, E, for all components of a transceiver in watts can be modeled by $E = \theta + \eta\omega d^n$, where θ is the distance-independent term that accounts for the overhead of the radio electronics and digital processing, and $\eta\omega d^n$ is the distance-dependent term in which η represents the amplifier inefficiency factor, ω is the free-space path loss, and d is the distance.

Two *clustering protocols* in sensor networks are the *Low-Energy Adaptive Clustering Hierarchy* (LEACH) algorithm and the *Decentralized Energy-Efficient Cluster Propagation* (DEEP) protocol. DEEP is based on the idea of controlling the geographical dimensions of clusters and the distribution of cluster heads. Because of the balanced load among cluster heads, there is no need for frequent re-clustering, but after current cluster heads are out of energy, the protocol can rotate the cluster-head position among all the sensor nodes. Also, the identical distance between a pair of neighboring cluster heads leads to the ease of route set-up deployment. After establishing well-distributed cluster heads and clusters in a network, energy-conscious routing is essential in order to set communication routes among cluster heads. *ZigBee technology*, which is based on the IEEE 802.15.4 standard, is a related technology that uses low-power nodes.

20.8 Exercises

1. Assume that the maximum line of sight for each node in the sensor network shown in Figure 20.9 is exactly 1 mile. The normalized maximum energy at each node is 20. Assume that 1 mile is scaled down to 1 inch on this map, measured between two centers of circles.
    (a) Apply the cost function presented in Equation (20.15) for the best-energy path.
    (b) Apply the cost function for the shortest path shown in the figure.
2. *Computer simulation project*. Consider a field of 100 × 100 square meter, on which 50 sensor nodes are randomly distributed. Apply DEEP, and use a mathematical tool, such as Mathlab, to simulate the clustering process each at the size of five sensor nodes. Use randomly one of the four available levels of energy to each node.
    (a) Sketch all clusters and the locations on cluster heads.
    (b) Show on the same chart how cluster heads are connected to regular nodes.
3. *Computer simulation project*. Continue the project from exercise 2, and now assume that an event is detected by a sensor at the closest cluster to one of the field's corners. Sketch the intercluster routing to a base station situated at the opposite corner from the event cluster.
4. *Computer simulation project*. Continue the project from exercise 3, now using LEACH.