path routing, in sensor networks.

4.3 Classification of Routing Protocols in WSNs [Njamal04]

Routing in sensor networks is very challenging and different from contemporary wired/wireless networks, such as Ethernet and MANETs [Rwheinzelman99, Jkulik02]. The popular IP-based protocols cannot be applied to sensor networks because it is infeasible to build a global addressing scheme for the deployment and maintenance of thousands of tiny sensor nodes having limited resources. Many new algorithms have hence been developed for routing and forwarding data in sensor networks.

4.3.1 Proactive and Reactive Routing

Routing protocols can be *proactive*, *reactive*, or *hybrid* depending on how the route is found. Proactive protocols attempt to continuously evaluate the routes within the network so that all routes are computed before they are needed. In other words, when a packet needs to be forwarded, the route is already available and can be immediately adopted. Reactive protocols, on the other hand, invoke a route determination procedure only on demand. Hence, some sort of search procedure has to be employed to identify a route prior to data forwarding. Hybrid routing protocols attempt to integrate the above two ideas to take the advantages of both.

The advantage of proactive schemes is that there is little or no delay in determining a route whenever a route is needed. On the other hand, reactive protocols have to start a route discovery process to identify proper path information when a route is needed, which means that the time for determining a route can be quite significant. This leads to increased latency for packet delivery, and may not be applicable to real-time communication. However, proactive schemes are likewise not appropriate for the ad hoc networking environment whereas network topology changes fast and constantly. Such network dynamics may result in continuous route evaluation and maintenance, which use a large portion of the network resources. Particularly, when the changes are more frequent than the route requests, the routing information from the continuous evaluation process may not be necessary and never be used.

4.3.2 Flat and Hierarchical Routing

Based on the network structure, routing protocols in WSNs can also be broadly divided into *flat* routing and *hierarchical* routing. In *flat* routing schemes, equal

roles and functionality are typically assigned to each node. *Flat* routing protocols distribute information as needed to any node that can be reached, or receive information. *Hierarchical* routing protocols often group nodes together by function into a hierarchy or cluster. By assigning different roles to different type of nodes or performing traffic aggregation to reduce redundancy, a hierarchical protocol allows WSNs to make best use of the heterogeneous nodes' capability. In many hierarchical routing protocols, each cluster designates a cluster-head (CH) node to aggregate and relay intercluster traffic. These CH nodes may become the bottleneck, potentially resulting in network congestion and single point of failure. In addition, maintaining the hierarchy or cluster can be costly in terms of energy or bandwidth consumption for small- to moderate-sized WSNs, which indicates that flat schemes are favorable in this case. On the other hand, *hierarchical* routing protocols are often better suited to large WSNs due to their scalability.

In fact, there are many other ways to classify routing protocols based on different criteria, such as protocol operation, network flow, energy, and QoS awareness. In the remaining part of this chapter, we will focus on four typical categories: data-centric protocols, hierarchical routing protocols, location-based routing protocols, and multipath routing in WSNs.

4.4 Data-Centric Routing Protocols in WSNs

In many applications of WSNs, the physical area covered by the sensors and the number of deployed sensor nodes can be enormous. Typically, the meaningful data traffic is generated due to the sensors' response to a query from the users (e.g., sink or BS) or actively reporting a detected event. In either case, multiple sensors having the data of interest will initiate the data transmission, which may result in significant redundancy and resource wastage. Certainly, if sensor nodes are as reliable as the IP routers and globally addressable, the redundancy issue will be trivial to resolve. However, it is infeasible (if not impossible) to assign a unique identifier to each sensor node and make each sensor node globally addressable like the IP router in the Internet. Accordingly, *data-centric routing protocols* are proposed for WSNs. In the data-centric routing scheme, the sink sends queries to specific regions and waits for answers from these regions. These queries are described in a high-level language. As data is being requested through queries, attribute-based naming is necessary to specify the properties of the data of interest.

SPIN and Directed Diffusion are among the earliest data-centric protocols [Jkulik02, Rwheinzelman99, CIntranagonwiwat00], which consider data negotiation between nodes to eliminate redundant data and save energy. These two protocols motivated the design of many other protocols that followed similar concepts. Examples of such data-centric routing protocols are Rumor Routing [Bdavid02], Minimum Cost Forwarding Algorithm (MCFA) [Fye01], Gradient-Based Routing (GBR) [Cschurgers01], COUGAR [Yyao02], Energy-Aware

Routing [Rcshah02], etc. Rumor Routing is mainly intended for contexts in which geographic routing criteria is not applicable. The Rumor Routing protocol uses a set of long-lived agents to create paths that are directed toward the events they encounter. MCFA uses the information about the direction of routing. It gets rid of the unique ID and routing table; instead each node in MCFA maintains the least-cost estimate from the node itself to the BS. COUGAR considers the whole network as a distributed database system and uses declarative queries for query processing from network layer functions. It also utilizes in-network data aggregation for more energy saving. Energy-Aware Routing uses a set of suboptimal paths occasionally to increase the lifetime of the network. These paths are chosen by means of a probability function, which depends on the energy consumption of each path.

In the rest of this section, we describe three typical data dissemination schemes—flooding/gossiping, SPIN, and Directed Diffusion—in detail with a focus on their key ideas and performance issues.

4.4.1 Flooding and Gossiping

Flooding is a classical and straightforward mechanism to disseminate data in WSNs, which takes advantage of the broadcasting nature of the wireless medium. To deliver a particular packet from the source to the destination node with flooding, the source node broadcasts the data to all the neighbors. Upon receiving the packet, each neighbor will broadcast a copy of the packet to its neighbors. This process continues until the packet arrives at the destination or the packet is dropped. Flooding is very easy to implement, but it has a major drawback of increasing the network load with redundant traffic. In classical flooding, a node may blindly broadcast whatever it receives, regardless of whether or not the neighbor has already received a copy from another source. This leads to the *implosion* problem [Rwheinzelman99, Jkulik02]. Figure 4.1 shows the implosion problem where the same message goes to node C, from nodes A and B, thereby creating redundancy [Rwheinzelman99]. In Figure 4.1, a WSN with four nodes, A, B, C, and D, is shown. Assume that the data needs to be sent from node D to node C using flooding. Node D broadcasts the data (a) to its neighbors, which are nodes A and B. The nodes A and B forward the same data (a) to node C. Here the issue is that node C receives the same data twice. This *implosion* results in multiple copies of the same data packet floating around the network, and a node may receive multiple copies of the data information.

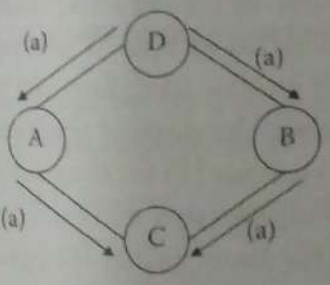

Figure 4.1 Implosion problem. (From Heidemann, R.W. et al., Adaptive protocols for information dissemination in wireless sensor networks, *ACM Mobicom '99*, Seattle, WA, August 1999, 174–185; Joanna, K. et al., *Wireless Netw., ACM*, 8(2/3), 169, March–May 2002.)

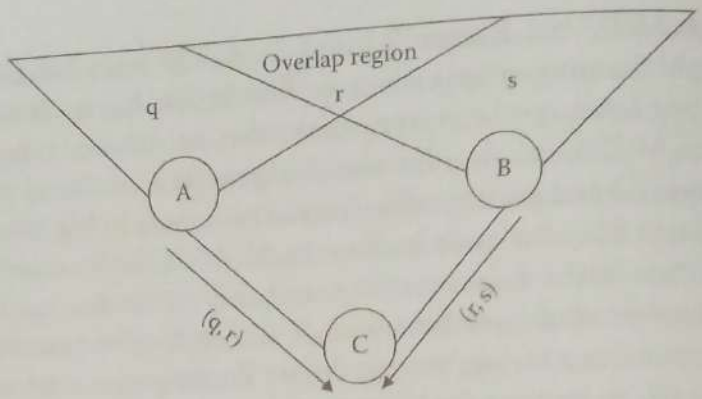

Figure 4.2 Example of overlap region. (From Heidemann, R.W. et al., Adaptive protocols for information dissemination in wireless sensor networks, *ACM Mobicom '99*, Seattle, WA, August 1999, 174–185; Joanna, K. et al., *Wireless Netw., ACM*, 8(2/3), 169, March–May 2002.)

Sensor nodes often cover overlapping geographic areas and gather overlapping pieces of event data. The sensed data received by the neighbors of the nodes would contain some part of the data that is redundant, which is known as *overlap* [Rwheinzelman99, Jkulik02]. Figure 4.2 shows an example of the overlapping issue. Node A in Figure 4.2 senses the data in the regions q and r. Similarly, node B senses the data in the regions r and s. Assume that the data sensed in the regions q and r is (q, r) and the data sensed in the regions r and s is (r, s). After sensing the data, the nodes A and B send the data (q, r) and (r, s) to node C. Obviously, the redundant copy of data (r) received at the destination node C is unnecessary.

The *implosion* and *overlap* issues lead to additional traffic in the network, which is unnecessary. The limited resources, such as energy and bandwidth, in WSNs will be wasted by this naïve flooding process. Hence, many studies have brought up techniques such as probability and packet ID to control the redundancies generated from the flooding process. For example, after assigning a unique ID for the packet, a sensor node can remember the IDs for the packets it broadcasted earlier. Then, the node can ignore the broadcast requests when it sees the same packet ID again. Similarly, a node may ignore a broadcast request according to a certain probability distribution. However, such techniques still cannot totally eliminate the flooding redundancies and may have considerable negative impacts on the network performance. To avoid the problem of flooding redundancy, gossiping takes a step further by just selecting one random node to forward the packet rather than broadcasting. In other words, in gossiping, the receiving node sends the packet to a randomly selected neighbor. The received packet is forwarded to another next-hop neighbor, which is also picked randomly to forward the packet and so on. However, the random selection of next-hop neighbors can cause delays in the propagation of data through the network.

to its neighbors. The interested neighbors possess the new data, can send a request message to the advertiser. Then the advertiser will reply with the data to the requested nodes. Unlike the classical flooding and gossiping protocols, which are blind to the resources' consumption in the network, SPIN uses a resource manager to become resource aware and resource adaptive in the process of data dissemination. The major goal of SPIN's metadata negotiation is to resolve the classical flooding problems, such as redundant information passing, overlapping of sensing areas, and resource blindness, and, thus, achieve better energy efficiency.

4.4.2.1 Design of SPIN

The design of SPIN is motivated by application-level framing (ALF) [Ddclark90]. Using ALF, the network protocols choose transmission units that are meaningful to applications. In other words, the packetization is best done in terms of application data units. Hence, SPIN designs metadata to ensure common naming data in both the transmission protocol and the application. Instead of sending the actual data, sensor nodes send metadata to interested neighbors in the form of an advertisement. The metadata must be smaller than the actual data for SPIN to be energy efficient. If the actual data is distinguishable, then the corresponding metadata should also be distinguishable. Similarly, two pieces of data that happen to be indistinguishable should have the same metadata. Generally, the format of metadata depends on the particular application [Rwheinzelman99].

Another important aspect of SPIN is that it uses a resource manager to monitor the available resources in the node and make the corresponding decision whether or not to participate in a particular data dissemination. Applications probe the resource manager before transmitting or processing data. The nodes using SPIN calculate the energy and resources available by means of polling the resource system. Hence, the routing decisions in SPIN are made by combining the knowledge of not only topology information but also application data layout and the status of resources available at each node.

There are three different types of messages in SPIN, which are new data advertisement (ADV), data request (REQ), and DATA message. When a node has a DATA message to share, the node can advertise this fact by transmitting an ADV packet containing the metadata of the message. A node that is interested in the details of the message based on the received metadata packet can send an REQ packet to the advertiser. Then, the requesting node will receive the DATA message containing actual details of the message with a metadata header from the advertiser.

4.4.2.2 Different Types of SPIN

The above SPIN philosophy is tuned to accommodate different WSN application and network scenarios. Four SPIN protocols are proposed in [Rwheinzelman99]: SPIN-PP, SPIN-BC, SPIN-EC, and SPIN-RL:

- SPIN-PP—for point-to-point transmission media. Assume that there is plentiful energy and packets are never lost in the network.
- SPIN-BC—for broadcast transmission media. Assume that there is plentiful energy and packets are never lost in the network.
- SPIN-EC—an energy-conserving version of SPIN-PP.
- SPIN-RL—a reliable version of SPIN-BC.

4.4.2.2.1 SPIN-PP

SPIN-PP employs three stages of message exchange for networks using point-to-point transmission media, which allow nodes A and B to communicate exclusively with each other without interfering with other nodes. The three stages correspond to the three messages described above. The protocol starts with a node sending an ADV message to its neighbors to advertise the data it intends to disseminate. In the next stage, the neighbors check whether they are interested in the advertised data after receiving the ADV packet. If a node determines to possess a copy of the data, the node sends back an REQ message to the node that sent the ADV message. Then, in the final stage, the actual data in the form of DATA message is delivered from the advertiser to the requester. Based on the received new DATA message and its own data in the memory, a node could perform some aggregation or redundancy-reducing processes prior to re-advertising the aggregated metadata to the neighbors.

4.4.2.2.2 SPIN-EC

SPIN-EC adds an energy-conserving scheme to the SPIN-PP protocol. When a node receives a new data, it will consult the resource manager before initiating the SPIN protocol and advertising the new metadata. The SPIN protocol will be started if and only if it turns out that the node has enough energy to complete all the stages of the protocol. Otherwise, it simply refrains from participating in the protocol. Similarly, upon receiving an advertisement, a node does not send out a request if it does not have enough energy to transmit the request and receive the corresponding data.

4.4.2.2.3 SPIN-BC

SPIN-BC is developed for broadcast transmission media. In SPIN-BC, the nodes use a single channel to broadcast the data to all the nodes in the receiving range. SPIN-BC employs the one-to-many communication scheme for delivering the same

data to multiple sensor nodes in one transmission. Similar to SPIN-PP, SPIN-BC also operates in three stages. There are three primary aspects in which SPIN-BC is different from SPIN-PP.

1. In SPIN-PP, one transmission can only target one specific node. Hence, a node has to send the advertised metadata to every neighbor in a separate transmission. However, taking advantage of the broadcast transmission media, every node within the transmission range could receive the same data in SPIN-BC.
2. Unlike SPIN-PP, SPIN-BC does not allow nodes to respond to the ADV packets immediately. In SPIN-BC, upon receiving the ADV packet, the nodes check whether they already possess the data advertised. If a node does not possess the data, the node sets a random timer. When the timer expires, the node broadcasts an REQ message to the original advertiser if the node does not receive the advertised data yet. Then the node advertising the metadata will respond to the REQ with the DATA message. When nodes other than the original advertiser receive the REQ, they cancel their own request timers to avoid redundant copies of the same request.
3. SPIN-BC will broadcast the DATA message only once and will not respond to multiple requests for the same data.

4.4.2.2.4 SPIN-RL

To handle the lossy link in WSNs, the SPIN-RL protocol makes two adjustments on SPIN-BC for reliable transmission. First, nodes employing the SPIN-RL protocol keep track of all the advertisements that are received. If a node does not receive the data within a particular period of time after sending out the request, the node consults the track of all advertisements received and sends another request to a randomly selected advertiser with the same piece of data. Second, nodes in SPIN-RL limit the frequency with which they will resend data to the neighbors. After a node sends the requested data, say (a), to other nodes, the node waits for some period of time before responding to any further requests demanding the same piece of data (a).

4.4.2.3 Evaluating SPIN Protocols [Rwheinzelman99, Jkulik02]

Using metadata names, nodes in SPIN negotiate with each other about the necessary data exchange. These negotiations ensure that nodes only transmit data when necessary and energy is not wasted on useless or redundant transmissions. With the resource manager, each node is aware of the available resources and is able to cut back on the activities to expand the lifetime of the network.

Table 4.1 shows the related parameters in the simulation with a randomly generated 25-node network [Jkulik02]. Each node in the network is initialized with 3 data items, randomly chosen from a set of 25 possible data items. No network loss and queuing delay is considered.

Table 4.1 Simulation Test Bed for SPIN

Nodes	25
Edges	59
Average degree	4.7 neighbors
Diameter	8 hops
Average shortest path	3.2 hops
Antenna reach	10 m
Radio propagation speed	3 × 8 m/s
Processing delay	5–10 ms
Radio speed	1 Mbps
Transmit cost	600 mW
Receive cost	200 mW
Data size	500 bytes
Metadata size	16 bytes

Source: Adapted from Joanna, K. et al., *Wireless Netw., ACM*, 8(2/3), 169, March–May 2002.

Table 4.2 shows the simulation results from SPIN-PP in which the *ideal dissemination* scheme is used as the baseline. Comparing the *flooding* and *gossiping* schemes, SPIN-PP consumes much less energy; it uses energy less by approximately a factor of 3.5 than flooding. This is partially due to the fact that flooding and gossiping schemes introduce much redundant data. As shown in Table 4.2, simulation shows that 77 percent of the transmitted DATA messages are redundant in the flooding scheme and 96 percent of them are redundant in the gossiping scheme. Note that SPIN-PP also introduces limited overhead traffic, such as the ADV and REQ packets, to the network.

The convergence time is defined as the time it takes to ensure that all the nodes in the network receive the intended data. SPIN-PP takes 80 ms longer to converge than flooding, whereas flooding takes only 10 ms longer to converge than the *ideal dissemination* scheme. Although it appears that SPIN-PP performs much worse than the flooding scheme in terms of the convergence time, this increase is actually a constant amount, regardless of the length of the simulation. Thus, for longer simulations, the increase in the convergence time for the SPIN-PP protocol will be negligible [Rwheinzelman99].

Other simulation and analysis results also indicate that SPIN-EC distributes 60 percent more data per unit energy than the flooding scheme. SPIN-PP and SPIN-EC outperform the gossiping scheme and come close to the *ideal dissemination*

Table 4.2 Results for Simulations of the SPIN-PP Protocol

Performance Relative to Ideal	*SPIN*	*Flooding*	*Gossiping*
Increase in energy dissipation	0.45 J	6.3 J	44.1 J
Increase in convergence time	90 ms	10 ms	3025 ms
Slope of energy Dissipation versus node degree correlation line	1.25x	5x	25x
Percent of total data messages that are redundant	0	77 percent	96 percent

Source: Adapted from Heidemann, R.W. et al., Adaptive protocols for information dissemination in wireless sensor networks, *ACM Mobicom '99*, Seattle, WA, August 1999, 174–185.

protocol. In addition, SPIN-BC and SPIN-RL are able to use one-to-many communications exclusively, while still acquiring data faster and using less energy than the flooding scheme. SPIN-RL can efficiently handle packet loss and dissipate twice the amount of data per unit energy as the flooding scheme.

4.4.3 Directed Diffusion [CIntanagonwiwat00]

Directed Diffusion differs from SPIN in terms of the way data transmission is initiated. The basic idea of *Directed Diffusion* is to diffuse data through sensor nodes by using a naming scheme for the data. With the naming scheme, the sink can issue a query to the sensor nodes regarding the data the sink is interested in. Then the corresponding sensor nodes reply with the necessary information to the sink. To achieve this, *Direct Diffusion* assigns attribute-value pairs to the data and queries on an on-demand basis. To issue a query indicating the type of data the sink is looking for, an *interest* is defined using the attribute-value pairs, such as name of objects, geographical area, duration, interval, etc. The sink disseminates the *interest* through its neighbors. The *interest* is cached in the sensor nodes. Whenever a node receives data, the node can compare the received data with the values of the *interest*. If there is a match, the node will establish paths to the sink from which the node receives the interest. These paths are known as events. Then, the sink can choose paths to resend the *interest* and expect the sensor node to reply with the data back to the sink.

Directed diffusion consists of several elements:

- *Data* is named using attribute-value pairs.
- *Interest* is a sensing task for named data.

and caching offers better network performance in terms of energy efficiency and delay. However, Directed Diffusion cannot be applied to all sorts of sensor network applications, because it is based on a query-driven data-delivery model. The applications that require continuous data delivery to the sink may not work efficiently with a query-driven on-demand model, such as Directed Diffusion [Njamal04].

4.5 Hierarchical Routing Protocols in WSNs

Scalability is one of the major design concerns of sensor networks, particularly for many applications with a large number of sensor nodes deployed to cover a pretty large physical area. A single-tier or flat network operation in such large-scale sensor networks can cause

- Large convergence time for many algorithms and protocols
- Overload with increase in sensor density
- Large memory space required for storing the network information
- Increased latency, complexity, and instability in communication
- Inadequate tracking of events

Because the tiny sensors with limited resources are typically not capable of performing long-haul communication, the concept of clustering or hierarchical network routing has been pursued in many routing approaches to allow the system to cover a large area of interest without degrading the service. A cluster is generally a collection of nodes with similar missions, within similar vicinity, or having similar functionalities/resources. A hierarchical routing protocol can be viewed as a set of flat routing protocols, each operating at different levels of granularity. For example, in a two-tier hierarchical routing protocol, the intercluster component is essentially a flat routing protocol that computes the routes between clusters. Likewise, the intra-cluster component is a flat routing protocol, which generates routes between nodes in each cluster. Hierarchical routing protocols provide global routes to the network clusters, rather than individual nodes, which can simplify many aforementioned scalability issues in the network and are often better suited to very large networks compared to flat routing protocols. In addition, data aggregation and fusion can be performed within the cluster to decrease the number of messages transmitted to the sink, which can enhance the network performance in terms of energy efficiency.

Examples of hierarchical network routing protocols include LEACH [Heinzelman02], TEEN [AManjeshwar01], Adaptive Periodic Threshold-Sensitive Energy-Efficient Sensor Network protocol (APTEEN) [Marati02], Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [Slindsay02], Hierarchical-PEGASIS [ASavvides01], Minimum Energy Consumption Network (MECN) [Vrodoplu99], Small Minimum Energy Consumption Network (SMECN) [Lli01], Self-Organizing Protocol (SOP) [Lsubramanian00], Sensor Aggregate Routing

[Qfang03], Virtual Grid Architecture Routing [JNal-karaki04], Hierarchical Power-Aware Routing [Qli01], and Two-Tier Data Dimension (TTDD) [FYe02].

LEACH forms clusters of the sensor nodes based on the received signal strength and uses the local CHs as the gateway to the BS. TEEN is a hierarchical protocol designed to be responsive to sudden and drastic changes in the sensed attributes, such as temperature, pressure, and rainfall. APTEEN aims at both capturing periodic data collections and reacting to time-critical events. PEGASIS forms chains of sensor nodes so that each node transmits and receives from a neighbor and only one node is selected from that chain to transmit to the BS rather than forming multiple clusters. Hierarchical-PEGASIS, an extension of PEGASIS, aims at decreasing the delay incurred for packets during transmission to the BS. MECN finds a subnetwork of the WSN with less number of nodes and also finds the minimum global energy required for data transfer. SMECN, an extension of MECN, considers the obstacles in data transmission while relaxing the assumption in MECN that every node in the network can transmit to each other node. Sensor Aggregate Routing comprises of the sensor nodes with a grouping predicate for a collaborative, cooperative processing task. The parameters of the predicate depend on the task and the resource requirements. Virtual Grid Architecture Routing uses data processing and in-network processing to maximize the network lifetime. The network is divided into zones based on the global positioning system (GPS) information. Data aggregation is performed at two levels: local aggregation and global aggregation. Each zone has a local aggregator and master aggregator. In Hierarchical Power-Aware Routing, the network is divided into groups based on geographical proximity and each group is allowed to decide how to route the data such that the energy consumed for routing will be minimum. In TTDD, each source node builds a grid structure for disseminating the data to the mobile sinks. The nodes that sense an event process the signal, and one of the nodes in the group that sensed the event becomes the source of the sensed data. The source node then builds a grid structure to route the data to the other nodes in the network.

4.5.1 Low-Energy Adaptive Clustering Hierarchy Protocol [Heinzelman02]

The LEACH protocol is a self-organizing, adaptive clustering protocol that uses randomization to distribute the energy consumption evenly among the sensor nodes in the network. The LEACH protocol aims at increasing the system lifetime and reducing the latency for transferring the data. The LEACH protocol uses the following techniques to achieve its goals [Heinzelman02]:

1. Localized control for data transfers
2. Low-energy medium access control
3. Self-configuring, randomized, and adaptive cluster formation
4. Application-specific data processing, like data compression and data aggregation

The LEACH routing protocol divides the sensor nodes in the network into groups called clusters. The clusters have special types of nodes, called CH nodes. These nodes are used for transmission of the data to the BS. These nodes are also responsible for the medium access among the nodes in the cluster.

The CH nodes in the cluster consume more energy as compared to the non-CH nodes. To make the energy consumption uniform among the nodes in the network, the LEACH protocol uses randomized rotation for the selection of CHs among the other nodes in the network. LEACH uses the CH nodes for the transmission of data from the non-CH nodes to the BS. The data sensed by the nodes is sent to the CH nodes initially, and then the CH nodes transmit the data to the BS.

4.5.1.1 Protocol Design

As described in Chapters 1 and 2, in sensor networks deployed for environment monitoring or surveillance applications, the overlap in the sensing range and application-specific requirements makes the sensed data in a specific region redundant and strongly correlated. The basic idea of LEACH is to form sensor nodes into clusters and locally process the correlated data such that the useless or redundant transmissions in the network are reduced.

In LEACH, the sensor nodes in the network organize themselves into groups, also called clusters. Then LEACH randomly selects a few nodes as CHs and rotates this role to evenly distribute the energy load among all the sensors in the network. All non-CH nodes will collect sensed data and send the data to the CH. The CH aggregates and compresses the data arriving from nodes that belong to the respective cluster before it sends the aggregated packet to the sink (or BS).

Remember

The operation of LEACH is divided into *rounds*. Each round consists of two phases: the *setup* phase and the *steady state* phase. Each round starts with a *setup* phase when the clusters are organized and CHs are selected. What is followed is the *steady state* phase when the CHs collect and process the data from the nodes within their clu… the aggregated data is transferred to the sink.

4.5.1.2 Setup Phase: Cluster Formation and Cluster-Head Selection

In LEACH, sensor nodes are organized into clust… where nodes make …

node sets the necessary power level and sends the aggregated data to the sink. After a certain time, which is determined a priori, the network goes into the next round to start the *setup* and *steady state* phases again. The duration of the steady state phase is longer than the duration of the setup phase to minimize the overhead.

4.5.1.4 LEACH-Centralized

The previous approach provides certain advantages in forming a cluster using the preceding algorithm. But the previous algorithm has many disadvantages, such as the LEACH protocol offers no guarantee about the number of clusters in a particular area and the placement of the clusters. LEACH-C uses a centralized algorithm for clusters' formation and produces better results as compared to those of the LEACH protocol. In the LEACH-C protocol, each node scans the current location using GPS during the *setup* phase and transmits its current location as well as energy level to the sink. Based on the energy level and location information of all the nodes in the network, the sink can select the CHs and form the clusters optimally in terms of minimizing the amount of energy consumed for data transmission. Then the sink broadcasts the information of the cluster formation to the network. If a node is not assigned as a CH in this round, the node can go to sleep based on its TDMA transmission schedule. However, the CH node has to receive, aggregate, and forward data to the sink.

4.5.1.5 Evaluating LEACH Protocol

In hierarchical routing protocols, each cluster designates a single CH node to relay intercluster traffic. To prevent the CH node from becoming the traffic/energy "hot spot," potentially resulting in network congestion and single point of failure, LEACH adopts a distributed scheme to rotate CH roles to evenly distribute the load among all the nodes in the network. In addition, LEACH employs dynamic clustering, in-network data processing, power-controlled transmission, and collision avoidance schemes to increase the network lifetime. Studies in [Heinzelman02] show that LEACH can achieve over a factor of 7 reductions in energy dissipation compared to direct communication and a factor of 4–8 compared to the minimum transmission energy routing protocol. In addition, the LEACH-C protocol can further improve the network performance by forming better clusters using the global knowledge of the location and energy levels of each node in the network.

However, restricting nodes accessing through CHs can lead to suboptimal routes and data transmission, as potential neighbors in different clusters are prohibited from communicating directly. The idea of dynamic clustering incurs extra overhead for the cluster formation/maintenance, which may diminish the gain in energy consumption. Moreover, LEACH assumes that each node can transmit directly to the CH and the sink, which may be not applicable for networks deployed in large regions. Hence, LEACH has been extended to account for heterogeneous sensor nodes, better scalability, and energy efficiency in the literature.

4.5.2 Threshold-Sensitive Energy-Efficient Sensor Network Protocol [AManjeshwar01]

SPIN, LEACH, and Directed Diffusion protocols have been developed for applications requiring periodic environment monitoring or querying a snapshot of the relevant parameters at certain intervals. On the other hand, two hierarchical routing protocols called TEEN and APTEEN are proposed in [AManjeshwar01, Marati02] for time-critical applications, where responsiveness to changes in the sensed attributes is important. TEEN pursues a hierarchical approach along with the use of a data-centric mechanism to provide the end user with the ability to control the trade-off between energy efficiency, accuracy, and response time dynamically.

4.5.2.1 Sensor Network Model in TEEN

In TEEN, the sink or BS can transmit data to all the nodes in the network at any point of time. However, the sensor node cannot always reach the sink directly due to the constraints of power and transmission range. Unlike LEACH with only one-tier hierarchy, the network architecture in TEEN is based on multilevel hierarchical grouping, as shown in Figure 4.8, where closer nodes form clusters and this process takes place for the multiple levels (or tiers). The CH in each cluster collects data from its cluster members, aggregates the data, and sends the data to an upper-level CH or the BS. Figure 4.8 shows an example of multi-tier clustering. Nodes 1.1.1, 1.1.2, 1.1.3, 1.1.4, and 1.1.5 form a low-level cluster with node 1.1 as the CH. Similarly, nodes 1.2 and 1 serve as the CHs for the respective low-level clusters. The CHs 1.1, 1.2, and 1 from the low-level clusters, in turn, form a cluster with node 1 as the CH. Hence, node 1 also becomes the CH of the second-level cluster. This hierarchy pattern is repeated through the network to form multilevel hierarchies. The uppermost-level cluster nodes will be able to send data directly to the BS, which acts as the root of the uppermost hierarchy and supervises the entire network.

With this network architecture, TEEN allows the nodes to communicate with their immediate CH. Hence, a node does not have to reach the BS directly (as required in LEACH). The data from low-level clusters may travel through multiple CHs before reaching the BS. The CHs at each level will perform necessary data processing, such as aggregation and compression, to conserve energy for the transmission. To evenly distribute the energy consumption, the nodes take turns to serve as CHs, which is similar to LEACH.

4.5.2.2 Operation of TEEN Protocol

Figure 4.9 shows the time line of the TEEN operation. After the clusters are formed, the CH broadcasts two thresholds to the nodes: *hard threshold* and *soft threshold*.

Figure 4.8 An e
and Agarwal, D.
sensor networks,
Processing Symp

Cluster
change tim

Figure 4.9 Op
Agarwal, D.P.,
sensor networks
Processing Symp

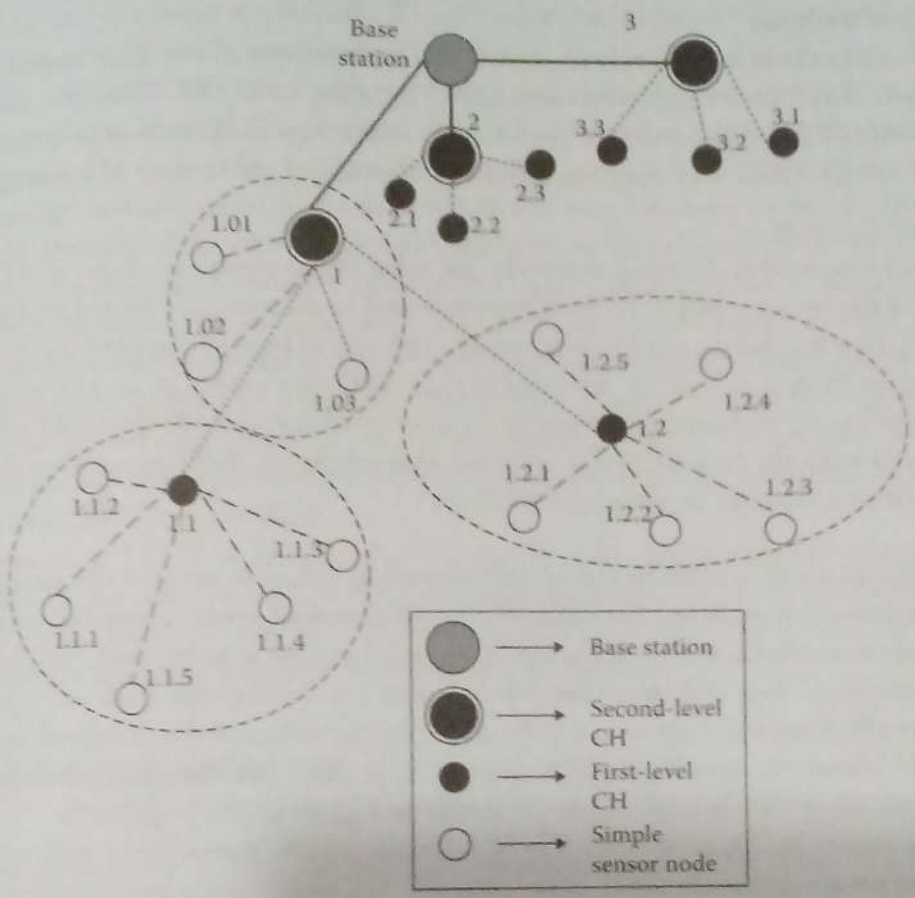

Figure 4.8 An example of network hierarchies in TEEN. (From Manjeshwar, A. and Agarwal, D.P., TEEN: A routing protocol for enhanced efficiency in wireless sensor networks, *Proceedings of 15th IEEE International Parallel and Distributed Processing Symposium*, San Francisco, CA, April 2001, 2009–2015.)

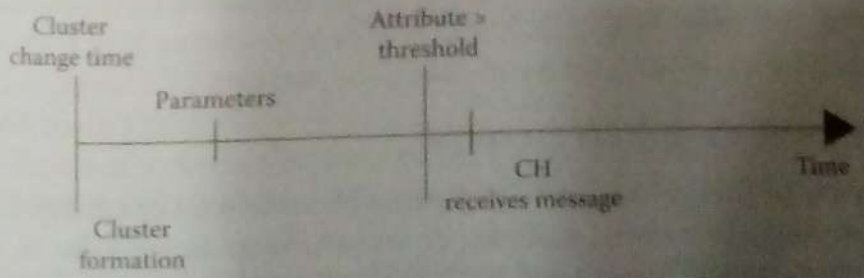

Figure 4.9 Operation of the TEEN protocol. (From Manjeshwar, A. and Agarwal, D.P., TEEN: A routing protocol for enhanced efficiency in wireless sensor networks, *Proceedings of 15th IEEE International Parallel and Distributed Processing Symposium*, San Francisco, CA, April 2001, 2009–2015.)

Hard threshold
Hard threshold is the threshold *value* of the attribute beyond which the sensing node must switch on its transmitter and report the value to its CH. Therefore, the hard threshold allows the nodes to transmit only when the sensed attribute is in the range of interest, which may result in significant reduction in the number of transmissions.

Soft threshold
Soft threshold is the small *change* in the value of the sensed attribute, which triggers the node to switch on its transmitter and to transmit the sensed data to the BS. In other words, once a node senses a value at or beyond the hard threshold, it transmits data only when the value of that attribute changes by an amount equal to or greater than the *soft threshold*. As a consequence, the soft threshold will further reduce the number of transmissions that might otherwise occur when there is little or no change in the sensed attribute.

One can adjust both hard and soft threshold values to control the number of data transmissions. A smaller value of the *soft threshold* gives a more accurate picture of the network, at the expense of increased data transmission and, thus, energy consumption. This indicates that the end user can control the trade-off between energy efficiency and data accuracy by adjusting the values of the threshold. In fact, TEEN allows the user to assign new threshold values and broadcast them to the network when CHs are to change (as shown in Figure 4.9).

As shown in Figure 4.9, the TEEN protocol initially forms the clusters and the parameters are sent to the nodes in the network. The nodes continuously monitor their environment. The first time the value of an attribute reaches its hard threshold value, the node switches on its transmitter and transmits the sensed data to the CH. The sensed data is also stored in an internal variable of the node, called the *sensed value* (SV), which is also updated whenever a node transmits data. The nodes will transmit data in any cluster period, only when both the following conditions are true [AManjeshwar01]:

1. The current value of the sensed attribute has to be greater than the hard threshold
2. The current value of the sensed attribute differs from the SV by an amount equal to or greater than the soft threshold

4.5.2.3 Evaluating TEEN Protocol

The important features of the TEEN protocol include its suitability for time-critical sensing applications. A sudden or drastic change in the value of a sensed attribute in these applications will reach the sink or user almost instantaneously. Also, as message transmission consumes much more energy than data sensing, TEEN can reduce unnecessary transmission, and hence the energy consumption in this scheme can potentially be much less when compared to that in the proactive network. By adjusting the threshold values according to the criticality of the sensed

attribute and the target application, TEEN can quickly adapt to the network's real condition and the user's specific requirements.

The simulation has been performed on a network of 100 nodes with a fixed BS in [AManjeshwar01]. The nodes are placed in a random fashion with an initial energy of 2 J in each node. Cluster formation is done as in the LEACH protocol. The energy consumption of the node is modeled as idle-time power dissipation (equal to the radio electronics energy) and sensing power dissipation (equal to 10 percent of the radio electronics energy). Two performance metrics are used to analyze and evaluate the protocols: *average energy dissipated* and *total number of nodes alive*. The *average energy dissipated* is defined as the average dissipation of energy per node over time in the network (as it performs various functions, such as transmitting, receiving, sensing, and aggregation of data). The *total number of nodes alive* indicates the overall lifetime of the network. Simulation results show that TEEN performs better than LEACH-C and LEACH.

However, TEEN is not suitable for applications where periodic reports are needed, because the user may not get any data at all whether or not the thresholds are reached. Thus, the user may not get any data and will never be able to know whether there are any nodes in the network that are alive.

4.5.2.4 Adaptive Periodic Threshold-Sensitive Energy-Efficient Network Protocol [Marati02]

As an extension to TEEN, the APTEEN protocol, on the other hand, is a hybrid protocol that changes the periodicity or threshold values used in the TEEN protocol according to user needs and the application type. APTEEN aims at proactively capturing periodic data collections and reactively responding to time-critical events. Its network-clustering architecture is the same as in TEEN. When the BS forms the clusters, the CH nodes broadcast the attributes, the threshold values, and the transmission schedule to all the nodes.

- *Attributes* are a set of physical parameters that need to be sensed in the network.
- *Thresholds* include soft and hard thresholds, which are the same as the thresholds in the TEEN protocol and serve the same purposes as in the TEEN protocol.
- *Count time* (*CT*) is the period of time after which the sensed data needs to be sent to the CHs.
- *Schedule* refers to the time division multiple access schedule, which is used for sharing the transmission medium among the sensor nodes in the network.

Similar to TEEN, the node in APTEEN senses the environment continuously, and only those nodes that sense a data value at or beyond the thresholds report the data to CHs. If a node does not send data for a time period equal to CT, APTEEN forces the node to sense and transmit the data. APTEEN supports three different query types:

- *Historical:* To analyze past data
- *One time:* To take a snapshot view of the network
- *Persistent:* To monitor an event for a period of time

A TDMA schedule is used, and each node in the cluster is assigned a transmission slot. APTEEN also allows the user to set the CT interval and the threshold values for energy efficiency. Simulations show that APTEEN's performance is somewhere between LEACH and TEEN in terms of energy dissipation and network lifetime. TEEN gives the best performance because it decreases the number of transmissions more significantly than APTEEN does. The drawbacks of TEEN and APTEEN, are the overhead and complexity associated with forming clusters at multiple levels, threshold-based functions, managing counter time and schedule, as well as dealing with attribute-based naming of queries.

4.6 Location-Based Routing Protocols in WSNs

With advances in sensor technologies, many applications densely deploy a large number of sensor nodes carrying a global positioning system (GPS) or a ranging device to facilitate the monitoring, tracing, or surveillance tasks. In the absence of a GPS unit, the location of nodes can be estimated through intelligent localization methods based on techniques such as coarse-grained connectivity, trilateration principle, robust quadrilaterals, and acoustic and multimodal sensing [Bulusu00, Ward97, Moore04, Girod01]. The location information of the sensors can be used to calculate the distance between the source and the destinations so that the energy consumption can be estimated or the transmission power level can be properly adjusted. In addition, recall the routing scheme called Directed Diffusion described earlier in this chapter; the location information can facilitate the sink to issue the query specifying the region in the *interest* message. Accordingly, location-based protocols are proposed to utilize position information to relay the data to the desired regions. Instead of diffusing the data to the whole network, nodes can target the data on a particular region or direction with the help of the geographical information, which potentially reduces the number of transmissions significantly, hence improving the network performance. Examples of location-based routing protocols are Geographic Adaptive Fidelity (GAF) [Yxu01], GEAR [Yyan01], Greedy Other Adaptive Face Routing (GOAFR) [Fkuhn03], and SPAN [Bchen02].

More specifically, GAF is an energy-aware location-based routing algorithm, designed primarily for MANETs, but may be applicable to sensor networks as well. GAF conducts routing based on the location of the node, which is associated with a point in the virtual grid formed for the covered area. GEAR uses energy-aware and geographically informed neighbor selection heuristics to route a packet toward the target region. The protocol suggests the use of geographical information while disseminating queries to appropriate regions, because data queries often include geographic

attributes. GOAFR routes the data by picking up the nearest neighbor to the node to be the next hop in the routing process. SPAN identifies some nodes as coordinators based on their positions to form a backbone network for data transmission.

4.6.1 Geographical and Energy-Aware Routing Protocol [Yyan01]

Unlike unicast communication, the GEAR protocol attempts to deliver data to all the nodes inside a target region, which is a common primitive in data-centric WSN applications. GEAR uses energy-aware and geographically informed neighbor selection heuristics to route data toward the specified region. Each node keeps an estimated cost and a learned cost of reaching the destination region through each neighbor. The estimated cost is a combination of residual energy and distance to the destination region, while the learned cost is a refinement of the estimated cost that accounts for routing around holes in the network. Based on the cost information, GEAR picks the next-hop neighbors intelligently to route the data to the destination region in an energy-efficient way. Once the data reaches the region, GEAR employs a recursive geographic forwarding technique to disseminate the packet within the region.

In fact, GEAR complements Directed Diffusion by restricting the number of interests' dissemination to a certain region rather than sending the interests to the whole network, thus conserving more energy.

4.6.1.1 Phases of GEAR

GEAR employs two phases in the process of forwarding data to all the nodes in the target region:

1. Forwarding the packet toward the target region
2. Disseminating the packet within the region

In the first phase, GEAR routes the data toward the target region. To forward the data toward the target region in an energy-efficient way, GEAR takes advantage of the geographical and energy information of sensor nodes to make routing decisions.

In the second phase, GEAR disseminates the data in the target region by using either recursive geographical forwarding or restricted flooding schemes. When the density in the target region is high, the region is further divided into four subregions. Four copies of the data are created and delivered to the subregions. This splitting and forwarding process continues until all the nodes in the target region are covered. On the other hand, when the density in the target region is low, restricted flooding is a better fit to save energy.