UNIT V
WIRELESS SENSOR NETWORKS

5.1 INTRODUCTION

Sensor networks are highly distributed networks of small, lightweight wireless nodes, deployed in large numbers to monitor the environment or system by the measurement of physical parameters such as temperature, pressure, or relative humidity. Building sensors has been made possible by the recent advances in micro-electro mechanical systems MEMS Technology.

Each node of the sensor network consists of three subsystems: the sensor subsystem which senses the environment, the processing subsystem which performs local computations on the sensed data, and the communication subsystem which is responsible for message exchange with neighboring sensor nodes. While individual sensors have limited sensing region, processing power, and energy, networking a large number of sensors gives rise to a robust, reliable, and accurate sensor network covering a wider region. The network is fault- tolerant because many nodes are sensing the same events. Further, the nodes cooperate and collaborate on their data, which leads to accurate sensing of events in the environment. The two most important operations in a sensor network are data dissemination, that is, the propagation of data/queries throughout the network, and data gathering, that is, the collection of observed data from the individual sensor nodes to a sink.

Sensor networks consist of different types of sensors such as seismic, thermal, visual, and infrared, and they monitor a variety of ambient conditions such as temperature, humidity, pressure, and characteristics of objects and their motion. Sensor nodes can be used in military, health, chemical processing, and disaster relief scenarios. Some of the academic and industry-supported research programs on sensor networks include working on Smart Dust at the University of California, Berkeley (UCB), and wireless integrated network sensor (WINS) at the University of California, Los Angeles (UCLA).

The applications of sensor networks are described in the next section, followed by the differences between ad hoc and sensor networks. The major issues and challenges involved in the design of sensor networks are then listed, and the two major forms of sensor network architecture — layered and clustered — are discussed. Various protocols for the major operations of data dissemination and gathering are then described, followed by specialized MAC protocols developed or modified to suit sensor networks. Techniques adopted by sensor nodes to discover their location and the measures to assess the quality of coverage of a sensor network are described. Finally, some sensor-network specific issues such as energy-efficient hardware design, synchronization, transport layer protocols, security, and real-time communication are discussed.

5.1.1 Applications of Sensor Networks

Sensor nodes are used in a variety of applications which require constant monitoring and detection of specific events. The military applications of sensor nodes include battlefield surveillance and monitoring, guidance systems of intelligent missiles, and detection of attack by weapons of mass destruction, such as chemical, biological, or nuclear. Sensors are also used in environmental applications such as forest fire and flood detection, and habitat exploration of animals. Sensors can be extremely useful in patient diagnosis and monitoring. Patients can wear small sensor devices that monitor their physiological data such as heart rate or blood pressure. The data collected can be sent regularly over the network to automated monitoring systems which are designed to alert the concerned doctor on detection of an anomaly. Such systems provide patients a greater freedom of movement instead of their being confined to a hospital.

Sensor nodes can also be made sophisticated enough to correctly identify allergies and prevent wrong diagnosis.
Sensors will soon find their way into a host of commercial applications at home and in industries. Smart sensor nodes can be built into appliances at home, such as ovens, refrigerators, and vacuum cleaners, which enable them to interact with each other and be remote-controlled. The home can provide a "smart environment" which adapts itself according to the user's tastes. For instance, the lighting, music, and ambiance in the room can be automatically set according to the user's preferences. Similar control is useful in office buildings too, where the airflow and temperature of different parts of the building can be automatically controlled. Warehouses could improve their inventory control system by installing sensors on the products to track their movement. The applications of sensor networks are endless, limited only by the human imagination.

5.1.2 Comparison with Ad Hoc Wireless Newtworks

While both ad hoc wireless networks and sensor networks consist of wireless nodes communicating with each other, there are certain challenges posed by sensor networks. The number of nodes in a sensor network can be several orders of magnitude larger than the number of nodes in an ad hoc network. Sensor nodes are more prone to failure and energy drain, and their battery sources are usually not replaceable or rechargeable. Sensor nodes may not have unique global identifiers, so unique addressing is not always feasible in sensor networks. Sensor networks are data-centric, that is, the queries in sensor networks are addressed to nodes which have data satisfying some conditions. For instance, a query may be addressed to all nodes "in the south-east quadrant," or to all nodes "which have recorded a temperature greater than 30 °C." On the other hand, ad hoc networks are address-centric, with queries addressed to particular nodes specified by their unique address. Hence, sensor networks require a different mechanism for routing and answering queries. Most routing protocols used in ad hoc networks cannot be directly ported to sensor networks because of limitations in memory, power, and processing capabilities in the sensor nodes and the non-scalable nature of the protocols. An important feature of sensor networks is data fusion/aggregation, whereby the sensor nodes aggregate the local information before relaying. The main goals of data fusion are to reduce bandwidth consumption, media access delay, and power consumption for communication.

5.1.3 Issues and Challenges in Designing a Sensor Network.

Sensor networks pose certain design challenges due to the following reasons:

• Sensor nodes are randomly deployed and hence do not fit into any regular topology. Once deployed, they usually do not require any human intervention. Hence, the setup and maintenance of the network should be entirely autonomous.

• Sensor networks are infrastructure-less. Therefore, all routing and maintenance algorithms need to be distributed.

• An important bottleneck in the operation of sensor nodes is the available energy. Sensors usually rely only on their battery for power, which in many cases cannot be recharged or replaced. Hence, the available energy at the nodes should be considered as a major constraint while designing protocols. For instance, it is desirable to give the user an option to trade off network lifetime for fault tolerance or accuracy of results.

• Hardware design for sensor nodes should also consider energy efficiency as a primary requirement. The micro-controller, operating system, and application

software should be designed to conserve power.

• Sensor nodes should be able to synchronize with each other in a completely distributed manner, so that TDMA schedules can be imposed and temporal ordering of detected events can be performed without ambiguity.

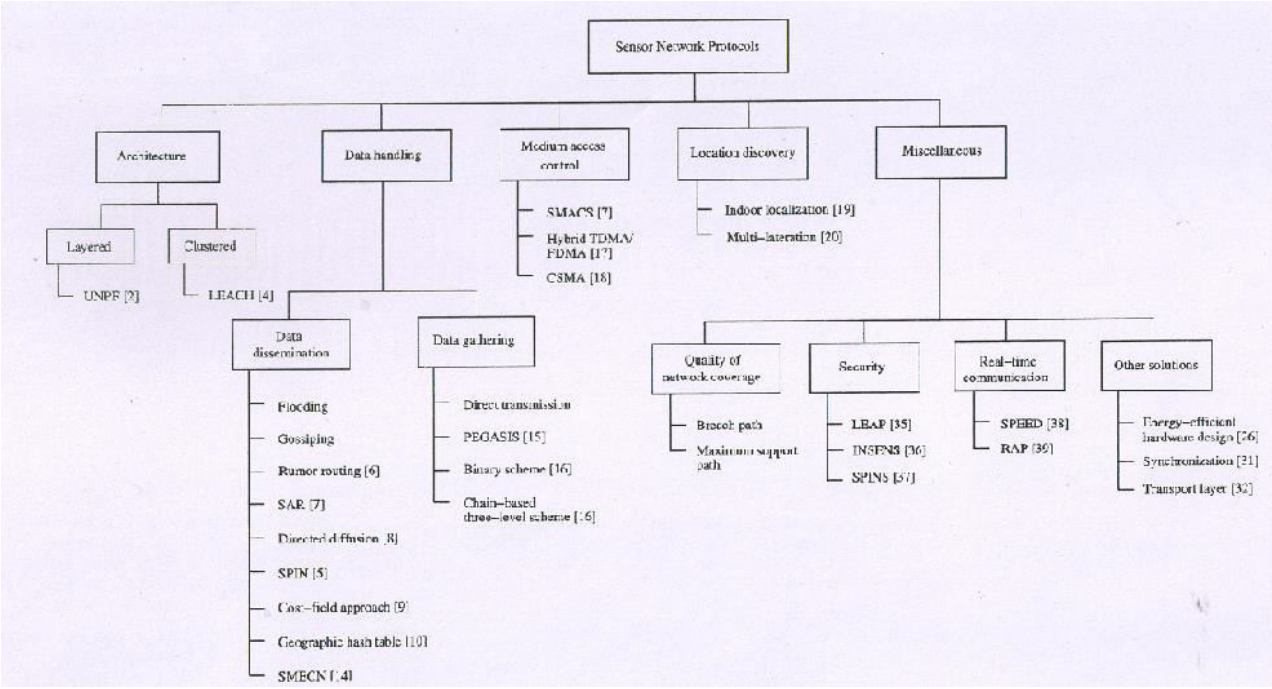
• A sensor network should also be capable of adapting to changing connectivity due to the failure of nodes, or new nodes powering up. The routing protocols should be able to dynamically include or avoid sensor nodes in their paths.

• Real-time communication over sensor networks must be supported through provision of guarantees on maximum delay, minimum bandwidth, or other QoS parameters.

• Provisions must be made for secure communication over sensor networks, especially for military applications which carry sensitive data.

The protocols which have been designed to address the above issues have been classified in Figure 5.1.

Figure 5.1. Classification of sensor network protocols.



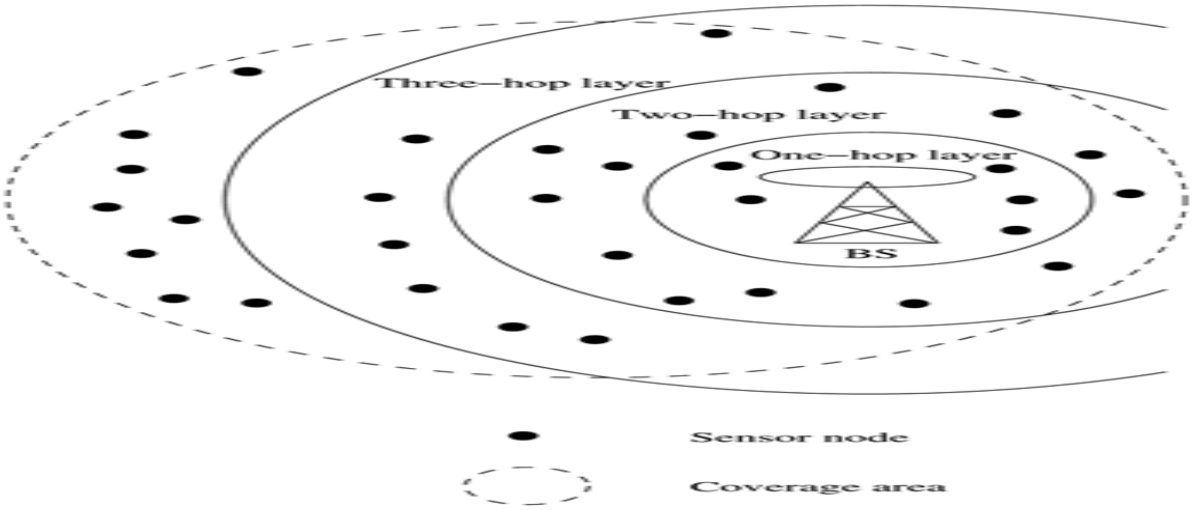
5.2 SENSOR NETWORK ARCHITECTURE

The design of sensor networks is influenced by factors such as scalability, fault tolerance, and power consumption. The two basic kinds of sensor network architecture are layered and clustered.

5.2.1 Layered Architecture

A layered architecture has a single powerful base station (BS), and the layers of sensor nodes around it correspond to the nodes that have the same hop-count to the BS. This is depicted in Figure 5.2.

Figure 5.2. Layered architecture.



Layered architectures have been used with in-building wireless backbones, and in military sensor-based infrastructure, such as the multi-hop infrastructure network architecture. In the in-building scenario, the BS acts an an access point to a wired network, and small nodes form a wireless backbone to provide wireless connectivity. The users of the network have hand-held devices such as PDAs which communicate via the small nodes to the BS. Similarly, in a military operation, the BS is a data-gathering and processing entity with a communication link to a larger network. A set of wireless sensor nodes is accessed by the hand-held devices of the soldiers. The advantage of a layered architecture is that each node is involved only in short-distance, low-power transmissions to nodes of the neighboring layers.

Unified Network Protocol Framework (UNPF)

UNPFis a set of protocols for complete implementation of a layered architecture for sensor networks. UNPF integrates three operations in its protocol structure: network initialization and maintenance, MAC, and routing protocols.

Network Initialization and Maintenance Protocol

The network initialization protocol organizes the sensor nodes into different layers, using the broadcast capability of the BS. The BS can reach all nodes in a one-hop communication over a common control channel. The BS broadcasts its identifier (ID) using a known CDMA code on the common control channel. All nodes which hear this broadcast then record the BS ID. They send a beacon signal with their own IDs at their low default power levels. Those nodes which the BS can hear form layer one since they are at a single-hop distance from the BS. The BS now broadcasts a control packet with all layer one node IDs. All nodes send a beacon signal again. The layer one nodes record the IDs which they hear, and these form layer two, since they are one hop away from layer one nodes. In the next round of beacons, the layer one nodes inform the BS of the layer two nodes, which is then broadcast to the entire network. In this way, the layered structure is built by successive rounds of beacons and BS broadcasts. Periodic beaconing updates neighbor information and alters the layer structure if nodes die out or move out of range.

• **MAC Protocol**

Network initialization is carried out on a common control channel. During the data transmission phase, the distributed TDMA receiver oriented channel (DTROC) assignment MAC protocol [3] is used. Each node is assigned a reception channel by the BS, and channel reuse is such that collisions are avoided. The node schedules transmission slots for all its neighbors and broadcasts the schedule. This enables collision-free transmission and saves energy, as nodes can turn off when they are not involved in a send/receive operation. The two steps of DTROC are channel allocation (the assignment of reception channels to the nodes) and channel scheduling (the sharing of the reception channel among the neighbors). DTROC avoids hidden terminal and exposed terminal problems by suitable channel allocationn algorithms.

• **Routing Protocol**

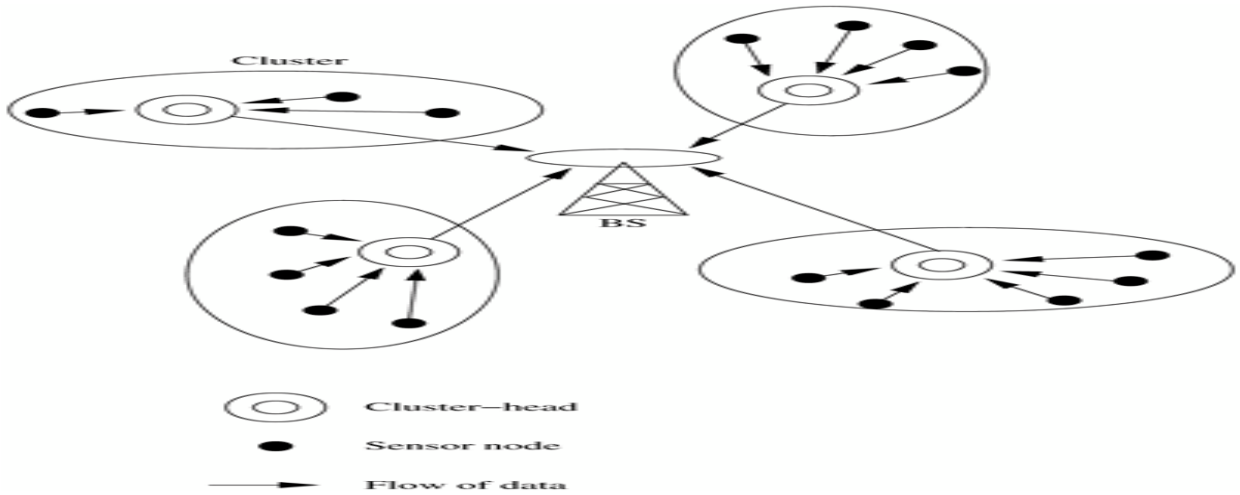
Downlink from the BS is by direct broadcast on the control channel. The layered architecture enables multi-hop data forwarding from the sensor nodes to the BS. The node to which a packet is to be forwarded is selected considering the remaining energy of the nodes. This achieves a higher network lifetime. Existing ad hoc routing protocols can be simplified for the layered architecture, since only nodes of the next layer need to be maintained in the routing table.

A modification to the UNPF protocol set termed the UNPF-R has been proposed. It makes the sensor nodes adaptively vary their transmission range so that network performance can be optimized. While a very small transmission range could cause network partitioning, a very large transmission range will reduce the spatial reuse of frequencies. The optimal range is determined through an algorithm similar to simulated annealing. This is a centralized control algorithm in which the BSevaluates an objective function periodically. For a transmission range *R,* the objective function is $f(R) = \frac{\epsilon \times d}{n/N}$, where *N* is the total number of sensors in the system; *n* is the number of nodes in layer one; ϵ is the energy consumption per packet; and *d* is the average packet delay. The BS selects a new transmission range *R′* as follows. If no packet is received by the BS from any sensor node for some interval of time, the transmission range is increased by Δr, a predefined increment. Otherwise, the transmission range is either decreased by Δr with probability 0.5 × (*n/N*), or increased by Δr with probability [1 - 0.5 × (*n/N*)]. The objective function is reevaluated with the new transmission range. If $f(R') < f(R)$, then the transmission range *R′* is adopted. Otherwise, *R* is modified to *R′* with probability $e^{\frac{(f(R)-f(R'))\times(n/N)}{T}}$, where *T* is the temperature parameter, as in simulated annealing. The advantage of theUNPF-R is that it minimizes the energy × delay metric, and maximizes the number of nodes which can connect to the BS. The minimization of the energy × delay metric ensures that transmission should occur with minimum delay and with minimum energy consumption. The two conflicting objectives are together optimized by minimizing their product.

5.2.2 Clustered Architecture

A clustered architecture organizes the sensor nodes into clusters, each governed by a cluster-head. The nodes in each cluster are involved in message exchanges with their respective cluster-heads, and these heads send messages to a BS, which is usually an access point connected to a wired network. Figure 5.3 epresents a clustered architecture where any message can reach the BS in at most two hops. Clustering can be extended to greater depths hierarchically.

Figure 5.3. Clustered architecture.



Clustered architecture is especially useful for sensor networks because of its inherent suitability for data fusion. The data gathered by all members of the cluster can be fused at the cluster-head, and only the resulting information needs to be communicated to the BS. Sensor networks should be self-organizing, hence the cluster formation and election of cluster-heads must be an autonomous, distributed process. This is achieved through network layer protocols such as the low-energy adaptive clustering hierarchy (LEACH) .

Low-Energy Adaptive Clustering Hierarchy (LEACH)

LEACH is a clustering-bas ed protocol that minimizes energy dissipation in sensor networks. LEACHrandomly selects nodes as cluster-heads and performs periodic reelection, so that the high-energy dissipation experienced by the cluster-heads in communicating with the BS is spread across all nodes of the network. Each iteration of selection of cluster-heads is called a round. The operation of LEACH is split into two phases: set-up and steady.

During the set-up phase, each sensor node chooses a random number between 0 and 1. If this is lower than the threshold for node n, $T(n)$, the sensor node becomes a cluster-head. The threshold $T(n)$ is calculated as

$$T(n) = \begin{cases} \frac{P}{1-P[r \times mod(1/P)]} & \text{if } n \in G \\ 0 & \text{otherwise,} \end{cases}$$

where P is the desired percentage of nodes which are cluster-heads, r is the current round, and G is the set of nodes that has not been cluster-heads in the past $1/P$ rounds. This ensures that all sensor nodes eventually spend equal energy. After selection, the cluster-heads advertise their selection to all nodes. All nodes choose their nearest cluster-head when they receive advertisements based on the received signal strength. The cluster-heads then assign a TDMA schedule for their cluster members.

The steady phase is of longer duration in order to minimize the overhead of cluster formation. During the steady phase, data transmission takes place based on the TDMA schedule, and the

cluster-heads perform data aggregation/fus ion through local computation. The BS receives only aggregated data from cluster- heads, leading to energy conservation. After a certain period of time in the steady phase, cluster-heads are selected again through the set-up phase.

5.3 DATA DISSEMINATION

Data dissemination is the process by which queries or data are routed in the sensor network. The data collected by sensor nodes has to be communicated to the BS or to any other node interested in the data. The node that generates data is called a *source* and the information to be reported is called an *event*. A node which is interested in an event and seeks information about it is called a *sink*. Traffic models have been developed for sensor networks such as the data collection and data dissemination (diffusion) models. In the data collection model, the source sends the data it collects to a collection entity such as the BS. This could be periodic or on demand. The data is processed in the central collection entity.

Data diffusion, on the other hand, consists of a two-step process of interest propagation and data propagation. An *interest* is a descriptor for a particular kind of data or event that a node is interested in, such as temperature, intrusion, or presence of bio-agents. For every event that a sink is interested in, it broadcasts its interest to its neighbors and periodically refreshes its interest. The interest is propagated across the network, and every node maintains an interest cache of all events to be reported. This is similar to a multicast tree formation, rooted at the sink. When an event is detected, it is reported to the interested nodes after referring to the interest cache. Intermediate nodes maintain a data cache and can aggregate the data or modify the rate of reporting data. The paths used for data propagation are modified by preferring the shortest paths and deselecting the weaker or longer paths. The basic idea of diffusion is made efficient and intelligent by different algorithms for interest and data routing.

5.3.1 Flooding

In flooding, each node which receives a packet broadcasts it if the maximum hop-count of the packet is not reached and the node itself is not the destination of the packet. This technique does not require complex topology maintenance or route discovery algorithms. But flooding has the following disadvantages :

• Implosion: This is the situation when duplicate messages are sent to the same node. This occurs when a node receives copies of the same message from many of its neighbors.

• Overlap: The same event may be sensed by more than one node due to overlapping regions of coverage. This results in their neighbors receiving duplicate reports of the same event.

• Resource blindness: The flooding protocol does not consider the available energy at the nodes and results in many redundant transmissions. Hence, it reduces the network lifetime.

5.3.2 Gossiping

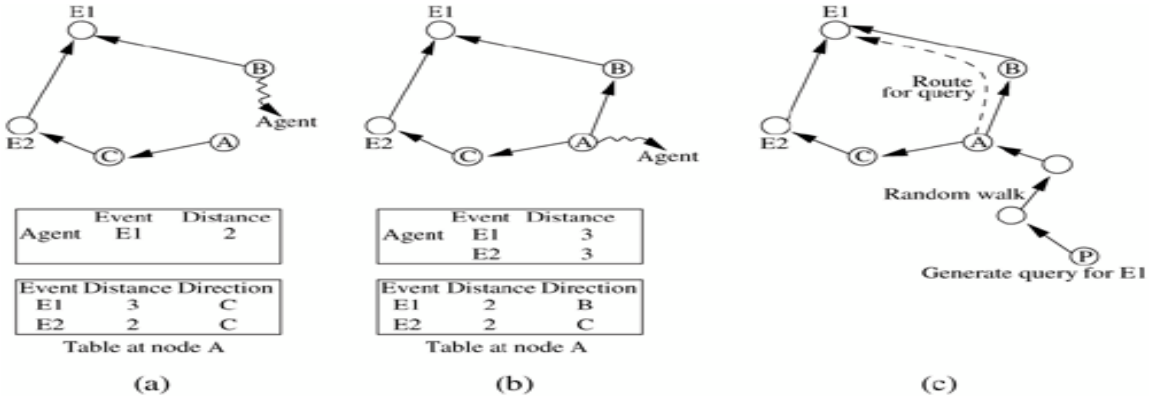
Gossiping is a modified version of flooding, where the nodes do not broadcast a packet, but send it to a randomly selected neighbor. This avoids the problem of implosion, but it takes a long time for a message to propagate throughout the network. Though gossiping has considerably lower overhead than flooding, it does not guarantee that all nodes of the network will receive the message. It relies on the random neighbor selection to eventually propagate the message throughout the network.

5.3.3 Rumor Routing

Rumor routing is an agent-based path creation algorithm [6]. Agents, or "ants," are long-lived entities created at random by nodes. These are basically packets which are circulated in the

network to establish shortest paths to events that they encounter. They can also perform path optimizations at nodes that they visit. When an agent finds a node whose path to an event is longer than its own, it updates the node's routing table. Figure 5.4 illustrates the working of the rumor routing algorithm. InFigure 5.4 (a), the agent has initially recorded a path of distance 2 to event *E*1. Node *A*'s table shows that it is at a distance 3 from event *E*1 and a distance 2 from *E*2. When the agent visits node *A,* it updates its own path state information to include the path to event *E*2. The updating is with one hop greater distance than what it found in *A,* to account for the hop between any neighbor of *A* that the agent will visit next, and *A*. It also optimizes the path to *E*1 recorded at node *A* to the shorter path through node *B*. The updated status of the agent and node table is shown in Figure 5 .4 (b).

Figure 5 .4. Rumor routing.

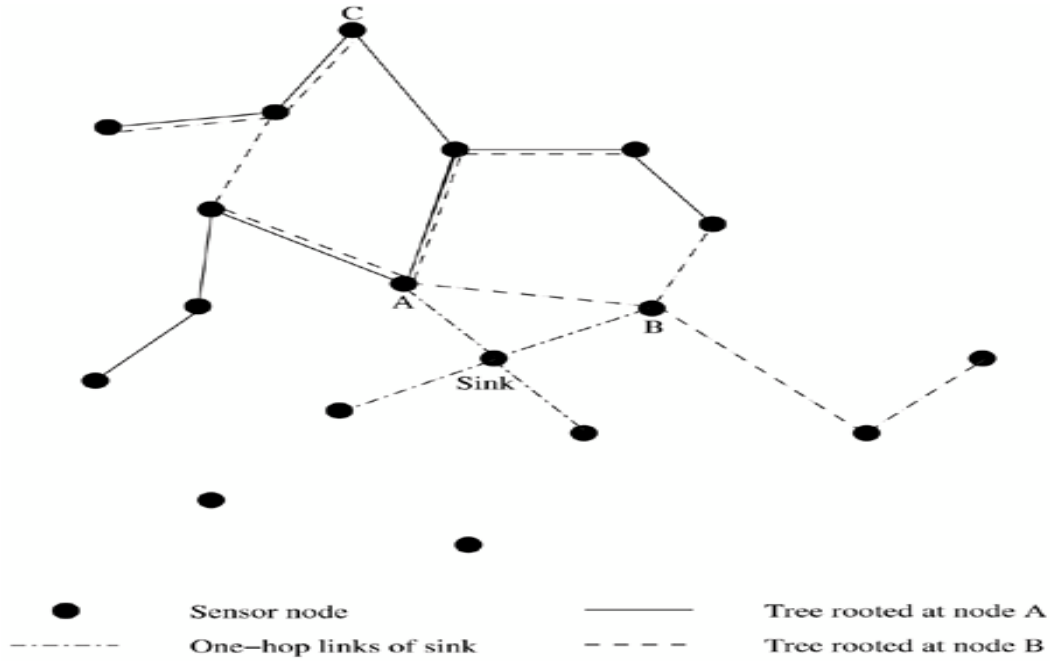


When a query is generated at a sink, it is sent on a random walk with the hope that it will find a path (pre established by an agent) leading to the required event. This is based on the high probability of two straight lines intersecting on a planar graph, assuming the network topology is like a planar graph, and the paths established can be approximated by straight lines owing to high density of the nodes. If a query does not find an event path, the sink times out and uses flooding as a last resort to propagate the query. For instance, as in Figure 5.4 (c), suppose a query for event *E*1 is generated by node *P*. Through a random walk, it reaches *A,* where it finds the previously established path to *E*1. Hence, the query is directed to *E*1 through node *B,* as indicated by *A*'s table.

5.3.4 Sequential Assignment Routing

A set of algorithms which performs organization and mobility management in sensor networks is proposed. The sequential assignment routing (SAR) algorithm creates multiple trees, where the root of each tree is a one-hop neighbor of the sink. Each tree grows outward from the sink and avoids nodes with low throughput or high delay. At the end of the procedure, most nodes belong to multiple trees. An instance of tree formation is illustrated in Figure 5.5. The trees rooted at *A* and *B,* two of the one-hop neighbors of the sink, are shown. Node *C* belongs to both trees, and has path lengths of 3 and 5, respectively, to the sink, using the two trees. Each sensor node records two parameters about each path through it: the available energy resources on the path and an additive QoS metric such as delay.

Figure 5.5. Sequential assignment routing.



This allows a node to choose one path from among many to relay its message to the sink. The SAR algorithm chooses a path with high estimated energy resources, and provisions can be made to accommodate packets of different priorities. A weighted QoS metric is used to handle prioritized packets, which is computed as a product of priority level and delay. The routing ensures that the same weighted QoS metric is maintained. Thus, higher priority packets take lower delay paths, and lower priority packets have to use the paths of greater delay. For example, if node *C* generates a packet of priority 3, it follows the longer path along tree *B,* and a packet of priority 5 (higher priority) will follow the shorter path along tree *A,* so that the priority × delay QoS metric is maintained. SAR minimizes the average weighted QoS metric over the lifetime of the network. The sink periodically triggers a metric update to reflect the changes in available energy resource after some transmissions.

5.3.5 Directed Diffusion

The directed diffusion protocol is useful in scenarios where the sensor nodes themselves generate requests/queries for data sensed by other nodes, instead of all queries arising only from a BS. Hence, the sink for the query could be a BS or a sensor node. The directed diffusion routing protocol improves on data diffusion using interest gradients. Each sensor node names its data with one or more attributes, and other nodes express their interest depending on these attributes. Attribute-value pairs can be used to describe an interest in intrusion data as follows, where an interest is nothing but a set of descriptors for the data in which the querying node is interested.

```
type = vehicle          /* detect vehicle location */
interval = 1 s          /* report every 1 second */
rect = [0, 0, 600, 800] /* query addressed to sensors within this
                           rectangle*/
timestamp = 02:30:00    /* when the interest was originated*/
expiresAt = 03:00:00    /* till when the sink retains interest in
                           this data*/
```

The sink has to periodically refresh its interest if it still requires the data to be reported to it. Data is propagated along the reverse path of the interest propagation. Each path is associated with a gradient that is formed at the time of interest propagation. While positive gradients encourage the data flow along the path, negative gradients inhibit the distribution of data along a particular path. The strength of the interest is different toward different neighbors, resulting in source-to-sink paths with different gradients. The gradient corresponding to an interest is derived from the interval/data-rate field specified in the interest. For example, if there are two paths formed with gradients 0.8 and 0.4, the source may send twice as much data along the higher gradient path compared to the lower gradient one. For the interest mentioned earlier, a sensor may send data of the following kind:

```
type = vehicle         /* type of intrusion seen */
instance = car         /* particular instance of the type */
location = [200,250]   /* location of node */
confidence = 0.80      /* confidence of match */
timestamp = 02:45:20   /* time of detection */
```

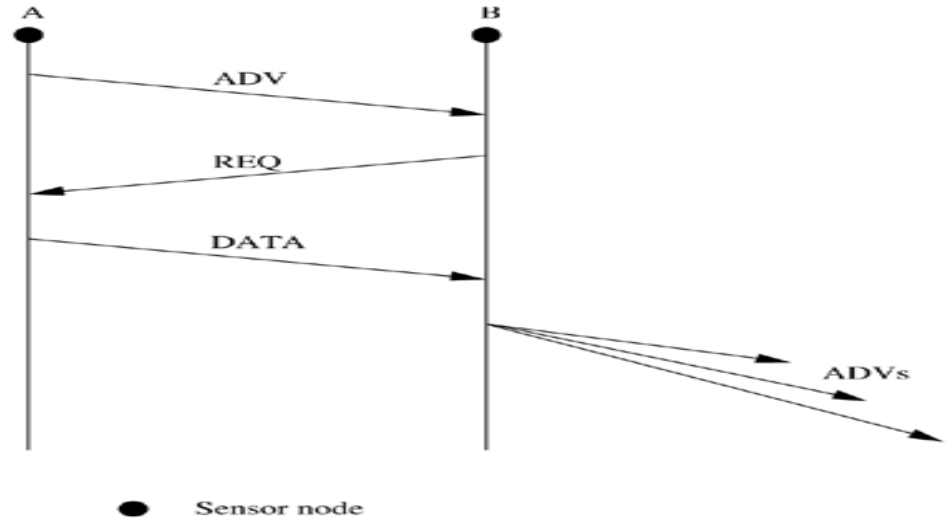
The diffusion model allows nodes to cache or locally transform (aggregate) data. This increases the scalability of communication and reduces the number of message transmissions required. The concept of reinforcement is used to update a node's interest along a particular path For example, suppose the sink wants more frequent updates from the sensors which have detected an event. It reinforces the path by sending an interest with a higher data-rate requirement, in effect increasing the gradient of that path. On the other hand, if the sink needs only fewer updates, it applies negative reinforcement by sending an interest of lower required data-rate.

The directed diffusion model uses data naming by attributes and local data transformations to reflect the data-centric nature of sensor network operations. The local operations of data aggregation are application-specific. Gradients model the network-wide results of local interactions by regulating the flow of data along different paths, depending on the expressed interest.

5.3.6 Sensor Protocols for Information via Negotiation

A family of protocols called sensor protocols for information via negotiation (SPIN) is proposed. SPIN uses negotiation and resource adaptation to address the deficienc ies of flooding. Negotiation reduces overlap and implosion, and a threshold-based resource-aware operation is used to prolong network lifetime. Meta-data, or data describing data, is transmitted instead of raw data. This requires fewer bytes and can be in an application-spec ific format. SPIN has three types of messages: ADV, REQ, and DATA. A sensor node broadcasts an ADV containing meta-data describing the actual data. If a neighbor is interested in the data, it sends a REQ for the data. Then the sensor node sends the actual DATA to the neighbor. The neighbor again sends ADVs to its neighbors and this process continues to disseminate the data throughout the network. This simple version of SPIN is shown in Figure 5.6.

Figure 5 .6. SPIN protocol.



SPIN is based on data-centric routing, where the nodes advertise the available data through an ADV and wait for requests from interested nodes. SPIN-2 expands on SPIN, using an energy or resource threshold to reduce participation. A node may participate in the ADV-REQ-DATA handshake only if it has sufficient resources above a threshold.

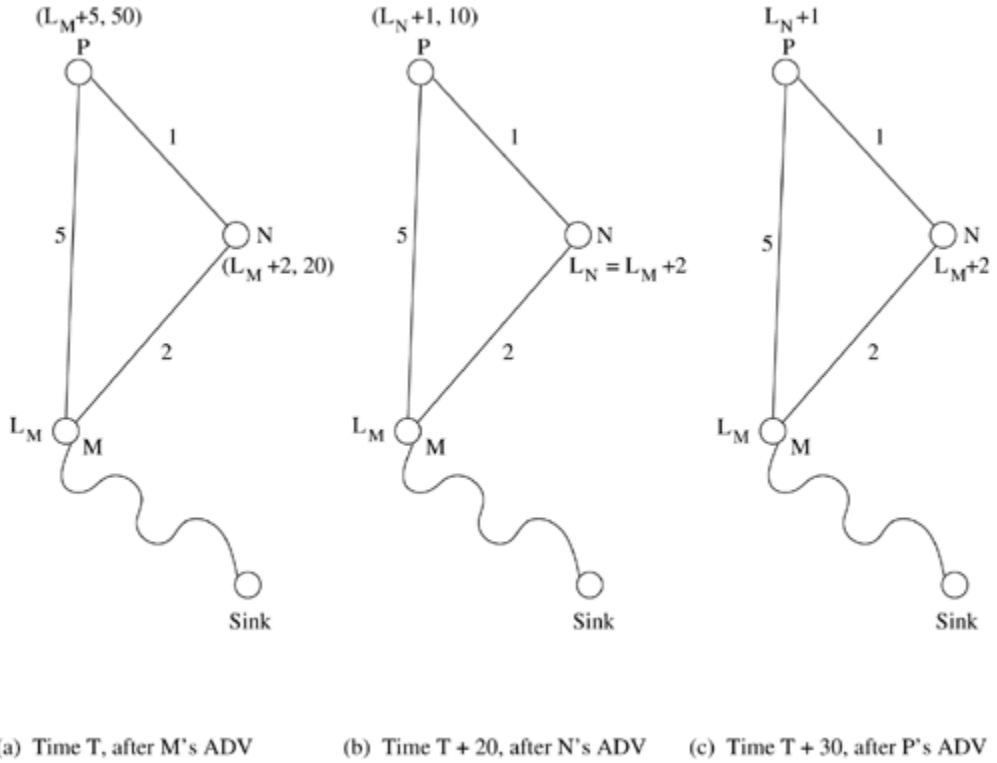
5.3.7 Cost-Field Approach

The cost-field approach considers the problem of setting up paths to a sink. It is a two-phase process, the first phase being to set up the cost field, based on metrics such as delay, at all sensor nodes, and the second being data dissemination using the costs. At each node, the cost is defined as the minimum cost from that node to the sink, which occurs along the optimal path. Explicit path information does not need to be maintained.

Phase 1 sets up a cost field starting from the sink node. A sink broadcasts an ADV packet with its own cost as 0. When a node N hears an ADV message from node M, it sets its own path cost to $min(L_N, L_M + C_{NM})$, where L_N is the total path cost from node N to sink, L_M represents the cost of node M to sink, and C_{NM} is the cost from node N to M. If L_N was updated, the new cost is broadcast through another ADV. This is a flooding-based implementation of the Dijkstra's algorithm. In order to reduce the high communication costs associated with flooding, a back-off-based approach is used. The main reason for overhead is that a node broadcasts its updated cost immediately, whether it was the optimal cost or not. Instead, the back-off modification makes a node defer its ADV instead of immediately broadcasting it. The time to defer is heuristically determined as $\gamma \times C_{MN}$, where γ is a parameter of the algorithm. The working of the cost-field approach with back-off is illustrated in Figure 5.7 . The numbers on the links indicate link costs. The value of γ is assumed to be 10. Initially, nodes N and P did not have a path to the sink and hence had their costs set to ∞. In Figure 5 .7 (a), node M broadcasts an ADV, which is received by nodes N and P. They tentatively fix their costs to $L_M + 2$ and $L_M + 5$, respectively, and set their back-off timers to 20 and 50, respectively. Figure 5.7 (b) shows the costs after 20 time units, when node N's back-off timer expires. Node N finalizes its cost to $L_M + 2$ and broadcasts an ADV, which is heard by node P. Since $L_N + 1 < L_M + 5$, node P updates its cost and sets a new back-off timer to 10. The unnecessary ADV of node P's earlier non-optimal cost is avoided

by setting the back-off timer. Finally, at 30 time units, node P finalizes its cost to $L_N + 1$ and broadcasts an ADV, as shown in Figure 5.7 (c).

Figure 5.7. Cost-field approach.



(a) Time T, after M's ADV (b) Time T + 20, after N's ADV (c) Time T + 30, after P's ADV

Phase 2 is the data dissemination process. Once the cost field is established, a source sends its message to sink S with cost C_S. The message also contains a cost-so-far field, initially set to 0. Each intermediate node forwards the packet if the cost recorded in the packet plus its own cost equals the original source-to- sink cost. This ensures that the original optimal path is used whenever a packet is routed. While forwarding, the intermediate nodes also update the cost-so-far field.

5.3.8 Geographic Hash Table

Geographic hash table (GHT) is a system based on data-centric storage inspired by Internet-scale distributed hash table (DHT) systems such as Chord and Tapestry. GHT hashes keys into geographic coordinates and stores a (key, value) pair at the sensor node nearest to the hash value. The calculated hash value is mapped onto a unique node consistently, so that queries for the data can be routed to the correct node. Stored data is replicated to ensure redundancy in case of node failures, and a consistency protocol is used to maintain the replicated data. The data is distributed among nodes such that it is scalable and the storage load is balanced. The routing protocol used is greedy perimeter stateless routing (GPSR), which again uses geographical information to route the data and queries.GHT is more effective in large sensor networks, where a large number of events are detected but not all are queried. In this case, the data observed is stored in a distributed manner across all nodes, instead of being routed to a central external storage. Queries are routed to the nearest node which contains a copy of the relevant data. This makes the storage and traffic distribution uniform.

5.3.9 Small Minimum Energy Communication Network

Small minimum energy communicatio n network (SMECN) is a protocol proposed in to construct a sub-network from a given communication network. If the entire sensor network is represented by a graph G, the subgraph G' is constructed such that the energy usage of the network is minimized. The number of edges in G' is less than that of G, but all nodes of G are retained in G'. The connectivity between any two nodes is not disrupted by the subgraph. G' is constructed such that the energy required to transmit data from a node to all its neighbors is lower in G' than in G. SMECN also follows the minimum energy (ME) property in its subgraph construction, that is, there exists an ME path in subgraph G' between any two nodes that are connected in G. The power required to transmit data between two nodes u and v is modeled as

$$p(u,v) = t \times d(u,v)^n$$

where t is a constant, n is the path loss exponent indicating the loss of power with distance from the transmitter, and $d(u, v)$ is the distance between u and v. Let the power needed to receive the data be c. Since the transmission power increases exponentially with distance, it would be more economical to transmit data by smaller hops. Suppose the path between u (*i.e.,* $u0$) and v (*i.e.,* u_k) is represented by $r1 = (u0, u1, \ldots u_k)$, such that each (u_i, u_{i+1}) is an edge in the subgraph G', then the total power consumed for the transmission is

$$C(r) = \sum_{i=0}^{k-1} (p(u_i, u_{i+1}) + c)$$

The path r is the ME path if $C(r) \leq C(r')$ for all paths r' between u and v in the graph G. The subgraph G' is said to have the ME property if there exists a path r in G' which is an ME path in G, for all node pairs (u, v). SMECN uses only the ME paths from G' for data transmission, so that the overall energy consumed is minimized.

5.4 DATA GATHERING

The objective of the data-gathering problem is to transmit the sensed data from each sensor node to aBS. One round is defined as the BS collecting data from all the sensor nodes once. The goal of algorithms which implement data gathering is to maximize the number of rounds of communication before the nodes die and the network becomes inoperable. This means minimum energy should be consumed and the transmission should occur with minimum delay, which are conflicting requirements. Hence, the *energy* × *delay* metric is used to compare algorithms, since this metric measures speedy and energy-efficient data gathering. A few algorithms that implement data gathering are discussed below.

5.4.1 Direct Transmission

All sensor nodes transmit their data directly to the BS. This is extremely expensive in terms of energy consumed, since the BS may be very far away from some nodes. Also, nodes must take turns while transmitting to the BS to avoid collision, so the media access delay is also large. Hence, this scheme performs poorly with respect to the energy × delay metric.

5.4.2 Power-Efficient Gathering for Sensor Information Systems

Power-efficient gathering for sensor information systems (PEGASIS) is a data-gathering

protocol based on the assumption that all sensor nodes know the location of every other node, that is, the topology information is available to all nodes. Also, any node has the required transmission range to reach the BS in one hop, when it is selected as a leader. The goals of PEGASIS are as follows:

• Minimize the distance over which each node transmits

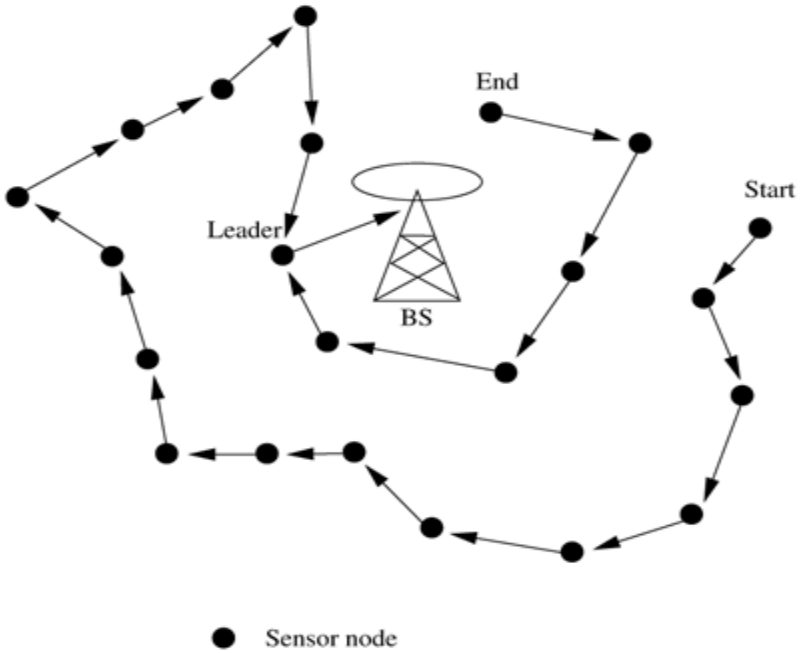
• Minimize the broadcasting overhead

• Minimize the number of messages that need to be sent to the BS

• Distribute the energy consumption equally across all nodes

A greedy algorithm is used to construct a chain of sensor nodes, starting from the node farthest from the BS. At each step, the nearest neighbor which has not been visited is added to the chain. The chain is constructed *a priori*, before data transmission begins, and is reconstructed when nodes die out. At every node, data fusion or aggregation is carried out, so that only one message is passed on from one node to the next. A node which is designated as the leader finally transmits one message to the BS. Leadership is transferred in sequential order, and a token is passed so that the nodes know in which direction to pass messages in order to reach the leader. A possible chain formation is illustrated in Figure 5.8. The delay involved in messages reaching the BS is $O(N)$,where N is the total number of nodes in the network.

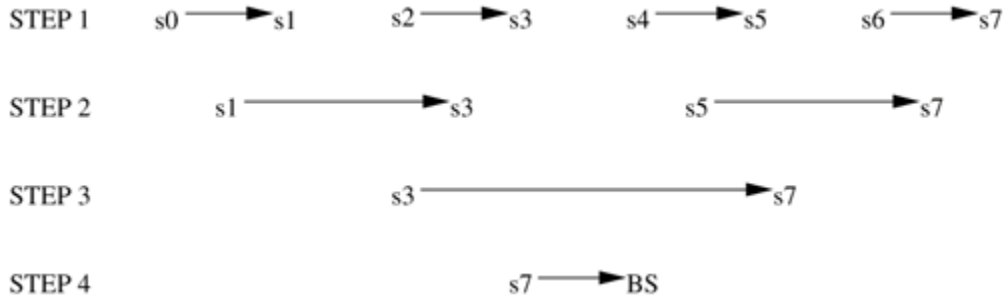
Figure 5.8. Data gathering with PEGASIS.



5.4.3 Binary Scheme

This is also a chain-based scheme like PEGASIS, which classifies nodes into different levels. All nodes which receive messages at one level rise to the next . The number of nodes is halved from one level to the next. For instance, consider a network with eight nodes labeled $s0$ to $s7$. As Figure 5.9 shows, the aggregated data reaches the BS in four steps, which is $O(log_2 N)$, where N is the number of nodes in the network. This scheme is possible when nodes communicate using CDMA, so that transmissions of each level can take place simultaneously.

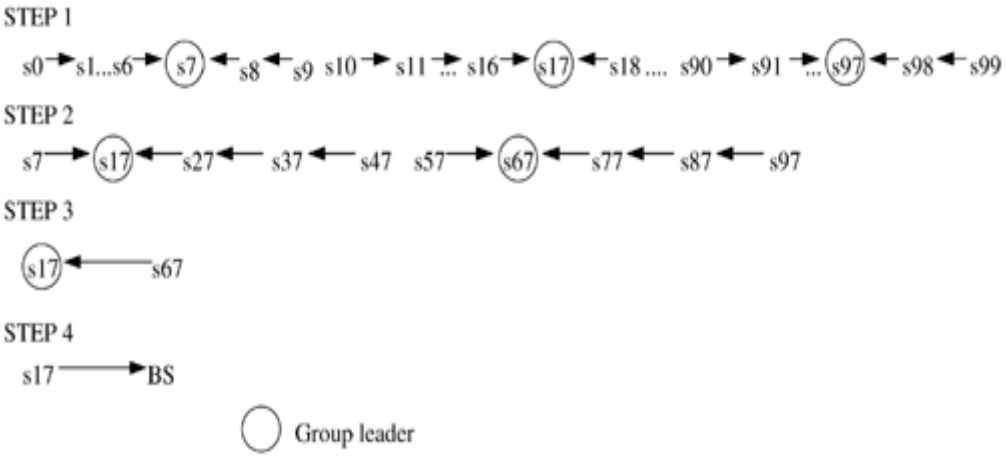
Figure 5.9. Binary scheme.



5.4.4 Chain-Based Three-Level Scheme

For non-CDMA sensor nodes, a binary scheme is not applicable. The chain- based three-level scheme addresses this situation, where again a chain is constructed as in PEGASIS. The chain is divided into a number of groups to space out simultaneous transmissions in order to minimize interference. Within a group, nodes transmit one at a time. One node out of each group aggregates data from all group members and rises to the next level. The index of this leader node is decided *a priori*. In the second level, all nodes are divided into two groups, and the third level consists of a message exchange between one node from each group of the second level. Finally, the leader transmits a single message to the BS. The working of this scheme is illustrated in Figure 5.10. The network has 100 nodes, and the group size is ten for the first level and five for the second level. Three levels have been found to give the optimal energy × delay through simulations.

Figure 5.10. Chain-based three-level scheme.



5.5 MAC PROTOCOLS FOR SENSOR NETWORKS

MAC protocols in sensor networks must create a network infrastructure to establish communication links among the thousands of randomly scattered sensors. It must also ensure fair and efficient sharing of communication resources among the nodes, so that the overall lifetime of the network can be maximized. The challenges posed by sensor network MAC protocols make them distinct from other wireless based networks. Unlike infrastructure-based

cellular networks, there is no single controlling authority in sensor networks, so global synchronization becomes difficult. Power efficiency is of utmost concern in sensor networks. They also encounter frequent topology changes due to mobility and failure. These factors emphasize the need for MAC protocols specific to sensor networks.

There are three basic kinds of MAC protocols used in sensor networks: fixed- allocation, demand-based, and contention-based. Fixed- allocation MAC protocols share the common medium through a predetermined assignment. They are appropriate for sensor networks that continuously monitor and generate deterministic data traffic, since all nodes which have been allotted the channel can make use of their slot in each round. Fixed-allocation protocols provide a bounded delay for each node. However, in the case of bursty traffic, where the channel requirements of each node may vary over time, a fixed allocation may lead to inefficient usage of the channel. Demand- based MAC protocols are used in such cases, where the channel is allocated according to the demand of the node. Though they require the additional overhead of a reservation process, variable rate traffic can be efficiently transmitted using demand-based MAC protocols. Finally, the contention-based MAC protocols involve random-access-based contention for the channel when packets need to be transmitted. They are again suitable for bursty traffic, but there is a possibility of collisions and no delay guarantees can be provided. Hence, they are not suitable for delay-sensitive or real-time traffic. Some of the popular sensor network MAC protocols have been briefly described in the next section.

5.5.1 Self-Organizing MAC for Sensor Networks and Eavesdrop and Register

Self-organizing MAC for sensor (SMACS) networks and eavesdrop and register (EAR) are two protocols which handle network initialization and mobility support, respectively. SMACS is a distributed protocol for network initialization and link-layer organization. In this protocol, neighbor discovery and channel assignment take place simultaneously in a completely distributed manner. A communication link between two nodes consists of a pair of time slots, at a fixed frequency, which is randomly chosen at the time of establishing the link. Such an assignment is possible in sensor networks without interference from neighboring nodes because the available bandwidth is much larger than the data rate required for a message transmission between two nodes. This scheme requires synchronization only between communicating neighbors, in order to precisely define the slots to be used for their communication. Power is conserved by turning off the transceiver during idle slots, and using a random wake-up schedule during the network start-up phase.

The EAR protocol enables seamless connection of nodes under mobile and stationary conditions. This protocol makes use of certain mobile nodes, besides the existing stationary sensor nodes, to offer service to maintain connections. Mobile nodes eavesdrop on the control signals and maintain neighbor information. The mobile nodes assume full control over connections and can drop connections when they move away. Mobility is hence made transparent to SMACS, since it is independently handled by EAR.

5.5.2 Hybrid TDMA/FDMA

This is a centrally controlled scheme which assumes that nodes communicate directly to a nearby BS. A pure TDMA scheme minimizes the time for which a node has to be kept on, but the associated time synchronization costs are very high. A pure FDMA scheme allots the minimum required bandwidth for each connection. The hybrid TDMA/FDMA scheme, proposed in , uses an optimum number of channels, which gives minimum overall power

consumption. This is found to depend on the ratio of power consumption of transmitter to receiver. If the transmitter consumes more power, a TDMA scheme is favored, since it can be switched off in idle slots to save power. On the other hand, the scheme favors FDMA when the receiver consumes greater power. This is because, in FDMA, the receiver need not expend power for time synchronization by receiving during the guard band between slots, which becomes essential in a TDMA scheme.

5.5.3 CSMA-Based MAC Protocols

Traditional CSMA-based schemes are more suitable for point-to-point stochastically distributed traffic flows. On the other hand, sensor networks have variable but periodic and correlated traffic. A CSMA-based MAC protocol for sensor networks has been described in . The sensing periods of CSMA are constant for energy efficiency, while the back-off is random to avoid repeated collisions. Binary exponential back-off is used to maintain fairness in the network. An adaptive transmission rate control (ARC) is also used, which balances originating and route-through traffic in nodes. This ensures that nodes closer to the BS are not favored over farther nodes. ARC uses linear increase and multiplicative decrease of originating traffic in a node. The penalty for dropping route-through traffic is higher, since energy has already been invested in making the packets reach until that node. ARC performs phase changes, that is, it staggers the transmission times of different streams so that periodic streams are less likely to collide repeatedly. Hence, CSMA based MAC protocols are contention-based and are designed mainly to increase energy efficiency and maintain fairness.

5.6 LOCATION DISCOVERY

The location information of sensors has to be considered during aggregation of sensed data. This implies each node should know its location and couple its location information with the data in the messages it sends. A low-power, inexpensive, and reasonably accurate mechanism is needed for location discovery. A global positioning system (GPS) is not always feasible because it cannot reach nodes in dense foliage or indoors. It also consumes high power and makes sensor nodes bulkier. Two basic mechanisms of location discovery are now described.

5.6.1 Indoor Localization

Indoor localization techniques use a fixed infrastructure to estimate the location of sensor nodes. Fixed beacon nodes are strategically placed in the field of observation, typically indoors, such as within a building. The randomly distributed sensors receive beacon signals from the beacon nodes and measure the signal strength, angle of arrival, and time difference between the arrival of different beacon signals. Using the measurements from multiple beacons, the nodes estimate their location. Some approaches use simple triangulation methods, while others require *a priori* database creation of signal measurements. The nodes estimate distances by looking up the database instead of performing computations. However, storage of the database may not be possible in each node, so only the BS may carry the database.

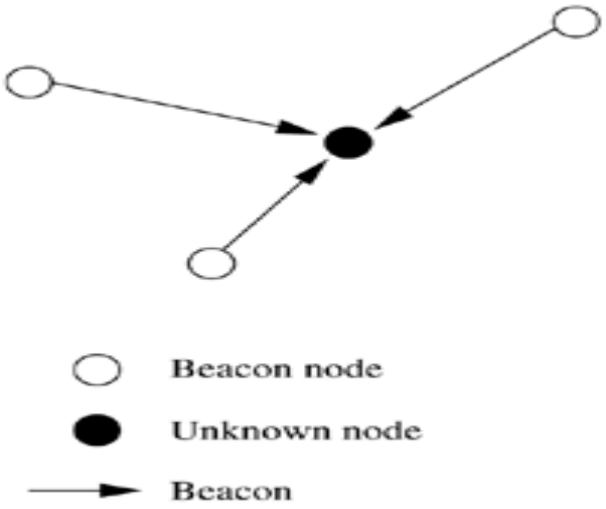
5.6.2 Sensor Network Localization

In situations where there is no fixed infrastructure available and prior measurements are not possible, some of the sensor nodes themselves act as beacons. They have their location information, using GPS, and these send periodic beacons to other nodes. In the case of communication using RF signals, the received signal strength indicator (RSSI) can be used to estimate the distance, but this is very sensitive to obstacles and environmental conditions. Alternatively, the time difference between beacon arrivals from different nodes can be used to

estimate location, if RF or ultrasound signals are used for communication. This offers a lower range of estimation than RSSI, but is of greater accuracy. Localization algorithms require techniques for location estimation depending on the beacon nodes' location. These are called multi-lateration (ML) techniques. Some simple ML techniques are described in what follows.

• Atomic ML: If a node receives three beacons, it can determine its position by a mechanism similar toGPS. This is illustrated in Figure 5.11.

Figure 5.11. Atomic multi- lateration.



• Iterative ML: Some nodes may not be in the direct range of three beacons. Once a node estimates its location, it sends out a beacon, which enables some other nodes to now receive at least three beacons. Iteratively, all nodes in the network can estimate their location. This is shown in Figure 5.12. The drawback of this multi-hop method is that errors are propagated, hence estimation of location may not be accurate.

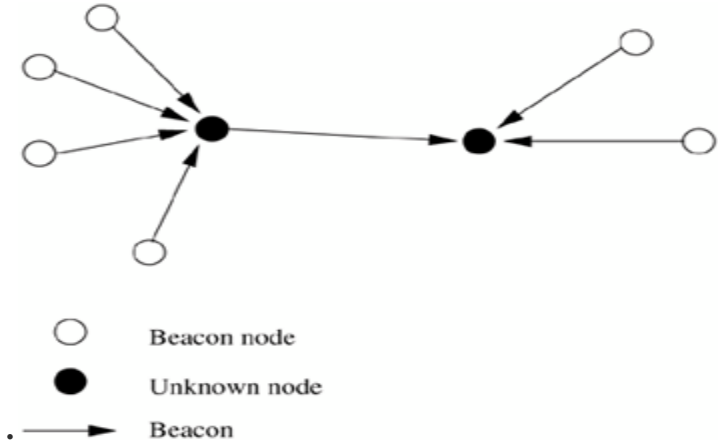
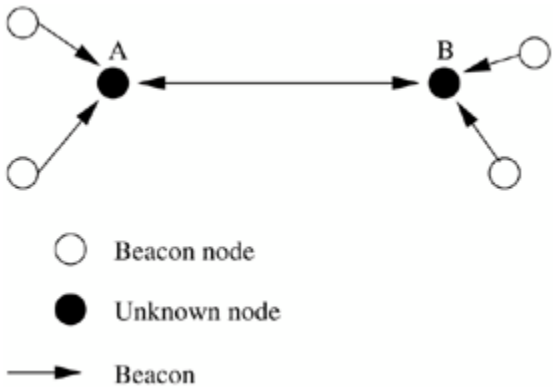


Figure 5.12. Iterative multi-lateration

• Collaborative ML: When two or more nodes cannot receive at least three beacons each, they collaborate with each other. As shown in Figure 5.13, node *A* and node *B* have three neighbors each. Of the six participating nodes, four are beacons, whose positions are known. Hence, by solving a set of simultaneous quadratic equations, the positions of *A* and *B* can be determined.

Figure 5.13. Collaborative multi-lateration.



A directionality-based localization approach has been explored in. This assumes that beacon nodes have broadcast capability to reach all nodes of the network, and a central controller rotates the beacons with a constant angular velocity ω radians/s. A constant angular separation is maintained between the beacon nodes. Nodes in the network measure the angles of arrival of beacon signals to estimate their location. The errors in this technique occur due to non- zero beam-width from the beacons. The beam is not a straight line as theoretically imagined, but it has a finite width. Hence, the measurement of the angle of the beacon signal will be inaccurate. The authors of propose an algorithm which derives the location of sensor nodes based mainly on information about connectivity between nodes. The all- pairs shortest paths algorithm is run on the network graph, which has edges indicating connectivity between nodes. Hence, the shortest distance between each pair of nodes is obtained. A mathematical technique called multi- dimensional scaling (MDS), an $O(n^3)$ algorithm (where n is the number of sensors), is used to assign locations to nodes such that the distance constraints are satisfied. The obtained picture of the network could be a rotated or flipped version of the actual network. If the actual positions of any three nodes in the network are known, then the entire network can be normalized (rotated or flipped) to obtain a very accurate localization of all other nodes.

5.7 QUALITY OF A SENSOR NETWORK

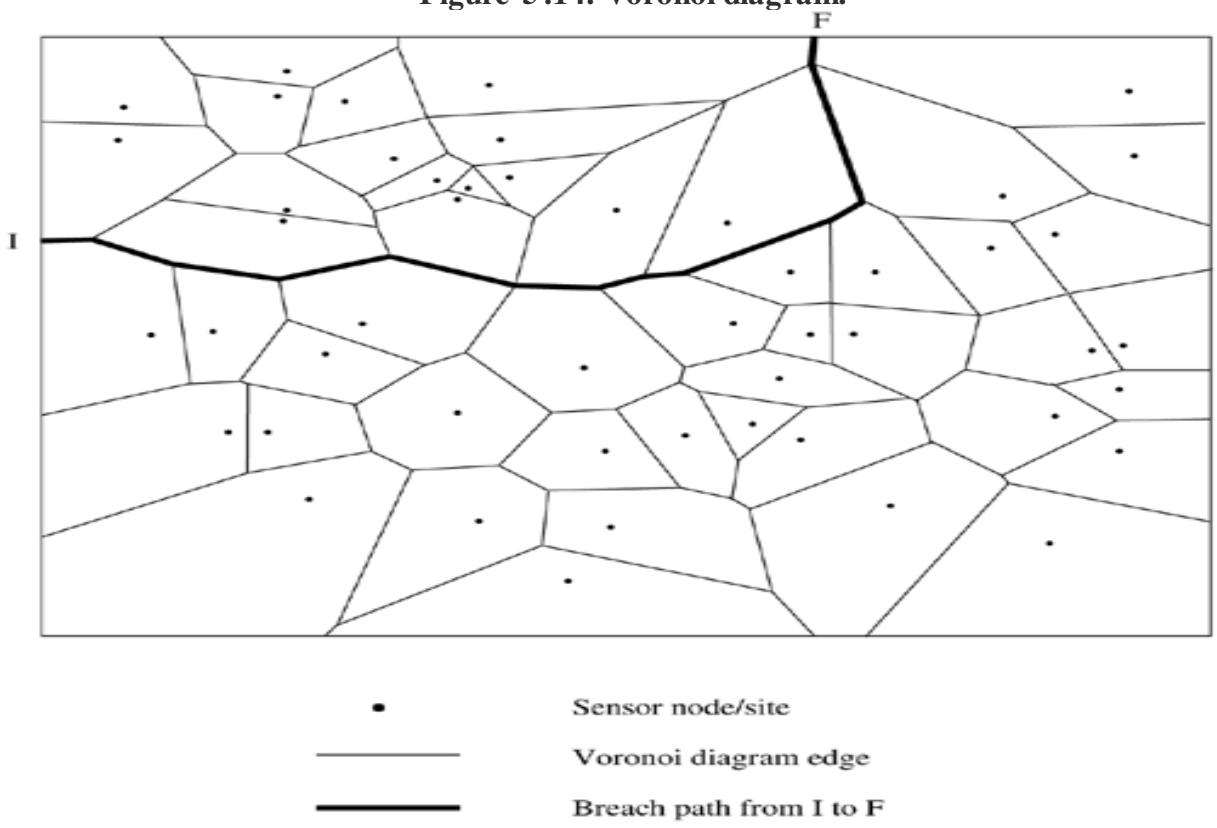
The purpose of a sensor network is to monitor and report events or phenomena taking place in a particular area. Hence, the main parameters which define how well the network observes a given area are "coverage" and "exposure." In this section, we shall formally define the coverage and exposure problems, and briefly describe some mathematical techniques to solve them.

5.7.1 Coverage

Coverage is a measure of how well the network can observe or cover an event. Coverage depends upon the range and sensitivity of the sensing nodes, and the location and density of the sensing nodes in the given region. The *worst- case* coverage defines areas of breach, that is, where coverage is the poorest. This can be used to determine if additional sensors need to be deployed to improve the network. The *best-case* coverage, on the other hand, defines the areas of best coverage. A path along the areas of best coverage is called a maximum support path or maximum exposure path. The coverage problem is formally defined as follows: Given a field A with a set of sensors $S = \{s_1, s_2, ..., s_n\}$, where for each sensor s_i in S, its location coordinates

(x_i , y_i) are known, based on localization techniques. Areas I and F are the initial and final locations of an intruder traversing the field. The problem is to identify P_B, the maximal breach path starting in I and ending in F. P_B is defined as the locus of points p in the region A, where p is in P_B if the distance from p to the closest sensor is maximized. A mathematic al technique to solve the coverage problem is the Voronoi diagram. It can be proved that the path P_B will be composed of line segments that belong to the Voronoi diagram corresponding to the sensor graph. In two dimensions, the Voronoi diagram of a set of sites is a partitioning of the plane into a set of convex polygons such that all points inside a polygon are closest to the site enclosed by the polygon, and the polygons have edges equidistant from the nearby sites. A Voronoi diagram for a sensor network, and a breach path from I to F, are shown in Figure 5.14.

Figure 5.14. Voronoi diagram.



The algorithm to find the breach path P_B is:

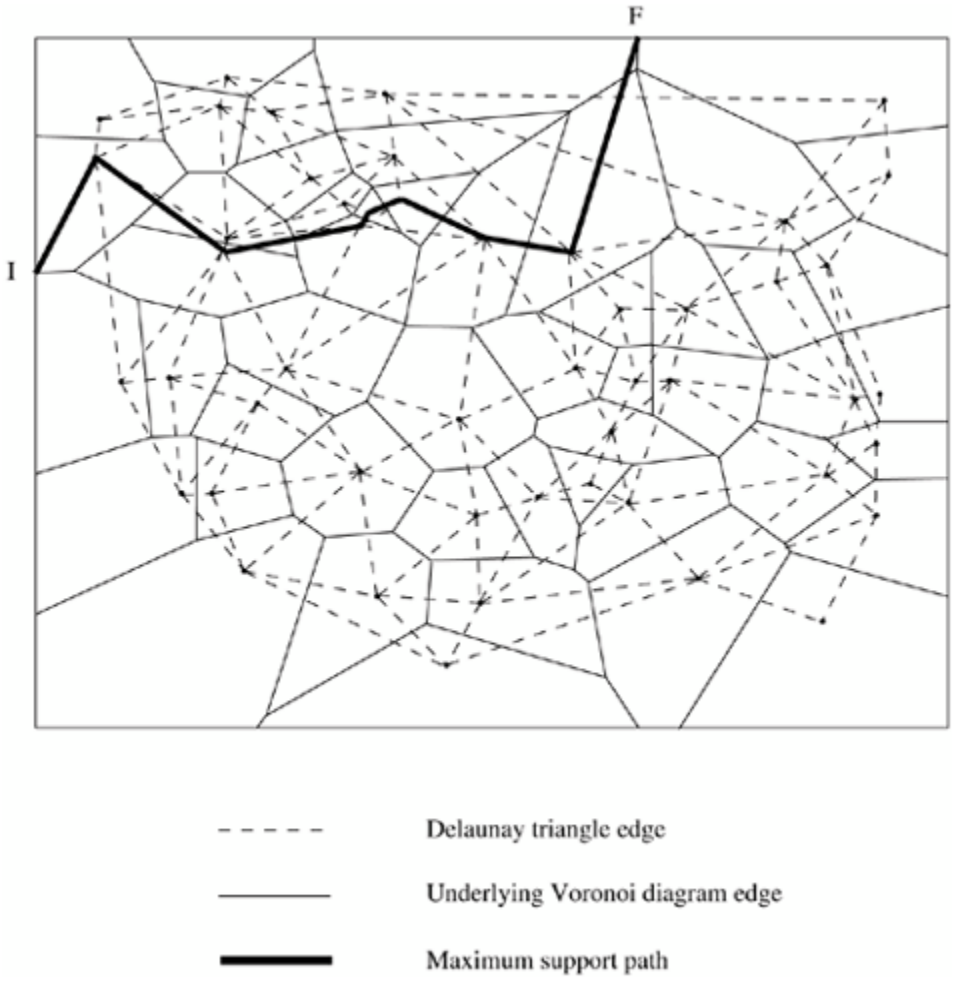
• Generate the Voronoi diagram, with the set of vertices V and the set of edges E. This is done by drawing the perpendicular bisectors of every line segment joining two sites, and using their points of intersection as the vertices of the convex polygons.

• Create a weighted graph with vertices from V and edges from E, such that the weight of each edge in the graph is the minimum distance from all sensors in S. The edge weights represent the distance from the nearest sensor. Smaller edge weights imply better coverage along the edge.

• Determine the maximum cost path from I to F, using breadth-first search. The maximum cost implies least coverage. Hence, the required breach path is along this maximum-cost path

determined from the Voronoi diagram. The breach path shows the region of maximum vulnerability in a sensor network, where the coverage provided by the sensors is the weakest. A related problem is that of finding the best-case coverage. The problem is formally stated as finding the path which offers the maximum coverage, that is, the maximum support path P_S in S, from I to F. The solution is obtained by a mathematical technique called Delaunay triangulation, shown in Figure 5.15. This is obtained from the Voronoi diagram by connecting the sites whose polygons share a common edge. The best path P_S will be a set of line segments from the Delaunay triangulation, connecting some of the sensor nodes. The algorithm is again similar to that used to find the maximum breach path, replacing the Voronoi diagram by the Delaunay triangulation, and defining the edge costs proportional to the line segment lengths. The maximum support path is hence formed by a set of line segments connecting some of the sensor nodes.

Figure 5.15. Delaunay triangulation.



5.7.2 Exposure

Exposure is defined as the expected ability of observing a target in the sensor field. It is formally defined as the integral of the sensing function on a path from source node P_s to destination node P_d. The sensing power of a node s at point p is usually modeled as

$$S(s,p) = \frac{\lambda}{[d(s,p)]^k}$$

where λ and k are constants, and $d(s, p)$ is the distance of p from s. Consider a network with sensors $s1$,$s2$, ..., s_n . The total intensity at point p, called the all- sensor field intensity, is given by

$$I_A(F,p) = \sum_{i=1}^{n} S(s_i,p)$$

The closest-sensor field intensity at p is

$$I_C(F,p) = S(s_{min},p)$$

where s_{min} is the closest sensor to p. The exposure during travel of an event along a path $p(t)$ is defined by the exposure function

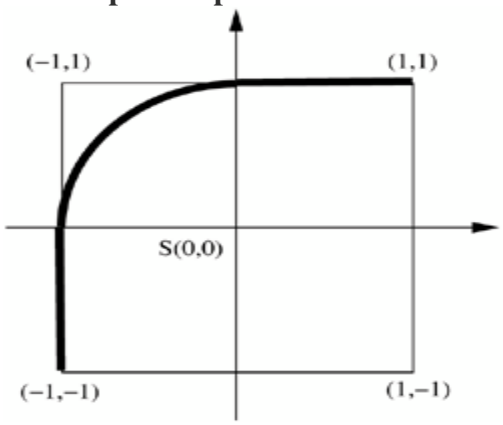
$$E[p(t),t_1,t_2] = \int_{t_1}^{t_2} I_{AorC}(F,p(t)) \left|\frac{dp(t)}{dt}\right| dt$$

where $\frac{dp(t)}{dt}$ is the elemental arc length, and $t1$, $t2$ are the time instances between which the path is traversed. For conversion from Cartesian coordinates
$(x(t), y(t))$,

$$\frac{dp(t)}{dt} = \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2}$$

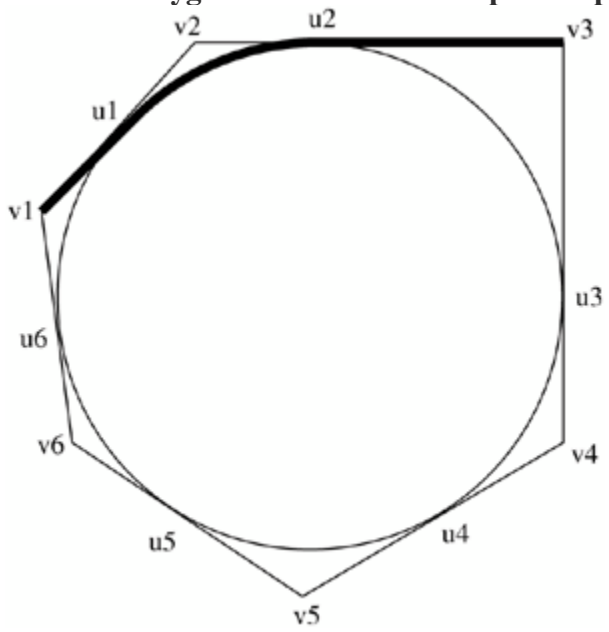
In the simplest case of having one sensor node at (0, 0) in a unit field, the breach path or minimum exposure path (MEP) from (-1, -1) to (1, 1) is shown in Figure5.16

Figure 5.16. Unit field minimum exposure path.



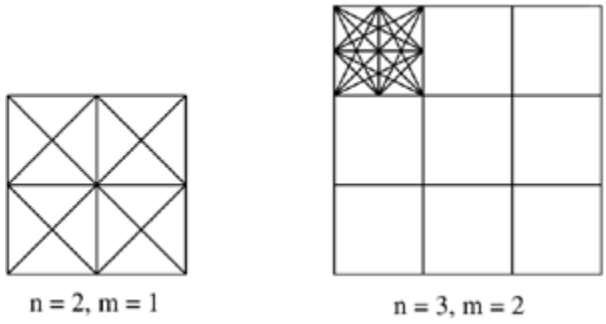
It can also be proved that for a single sensor s in a polygonal field, with vertices $v1$, $v2$, ..., vn, the MEP between two vertices v_i and v_j can be determined as follows. The edge (v_i, v_{i+1}) is tangent to the inscribed circle at u_i. Thenthe MEP consists of the line segment from v_i to u_i, part of the inscribed circle from u_i to u_j, and the line segment from u_j to v_j. This is shown in Figure 5.17.

Figure 5.17. Polygon field minimum exposure path.



The exposure problem is still unsolved for two points in the same corner, or for points within the inscribed circle. For the generic exposure problem of determining the MEP for randomly placed sensor nodes in the network, the network is tessellated with grid points. An example is shown in Figure 5.18. To construct an $n \times n$ grid of order m, each side of a square is divided into m equal parts, creating $(m + 1)$ vertices on the edge. Within each square, all vertices are connected to obtain a grid. Higher order grids have greater accuracy. For each edge in the grid network, the exposure function is used to determine the edge weights, and the MEP is defined as the shortest path, determined by Dijkstra's algorithm.

Figure 5.18. Generic minimum exposure path.



The mathematical concept of exposure is important for evaluating the target detection capability of a sensor network. Sensors are deployed in a given area to detect events occurring in the field

of interest. The nodes collaborate among themselves (perform data fusion) through the exchange of localized information, and reach a decision about the location and movement of a given event or target. In , Clouqueur *et al.* discuss a probabilistic protocol for target detection, where the observations made by individual sensors are collaborated, and the presence or movement of a target is probabilistically determined by data fusion, with allowance for noise in data recording. The network topology which gives a maximum exposure is also determined analytically.

5.8 EVOLVING STANDARDS

Standards for sensor networks are at an incipient stage. The IEEE 802.15.4 low- rate wireless personal area networks (LR-WPANs) standard investigates a low data rate solution with multi-mo nth to multi-year battery life and very low complexity. It is intended to operate in an unlicensed, international frequency band. Potential applications of this standard include sensor networks, home automation, and remote controls. The eighteenth draft of this standard was accepted in May 2003.

This standard aims to define the physical and MAC layer specifications for sensor and other WPAN networks. Low power consumption is an important feature targeted by the standard. This requires reduced transmission rate, power- efficient modulation techniques, and strict power management techniques such as sleep modes. Different network configurations and topologies were compared, and star and mesh networks were found to be favorable. The standard also proposes a generic frame structure whose length can be varied according to the application. Other standards under development include the SensIT project by the Defense Advanced Research Projects Agency (DARPA) which focuses on large distributed military systems, which addresses industrial and vehicular appliances. The IEEE 1451.5 wireless smart transducer interface standard is still under review. It is proposed to include multiple combinations of MAC and physical layers, using the IEEE 802 approach as a model.

5.9 OTHER ISSUES

This section deals with some issues that are recently being explored in sensor networks, such as energy-effic ient hardware and architecture, real-time communication on sensor networks, transport layer protocols, and security issues. Because these are mostly in the research stage, there are many improvements to be made on these fronts.

5.9.1 Energy-Efficient Design

As has been emphasized throughout the chapter, sensor nodes have a very stringent energy constraint. Energy optimization in sensor networks must prolong the life of a single node as well as of the entire network. Power saving in the micro-controller unit has been analyzed in, where the power required by different processors has been compared. The choice of the processor should be application-spec ific, such that performance requirements are met with the least power consumption. Computation can be carried out in a power-aware manner using dynamic power management (DPM). One of the basic DPM techniques is to shut down several components of the sensor node when no events take place. The processor has a time-varying computational load, hence the voltage supplied to it can be scaled to meet only the instantaneous processing requirement. This is called dynamic voltage scaling (DVS). The software used for sensor networks such as the operating system, application software, and network software can also be made energy-aware. The real-time task scheduler should actively support DVS by predicting the computation and communication loads. Sensor applications can use a trade-off between energy and accuracy by performing the most significant operations first, so that premature termination of the computation due to energy constraints does not affect the result by a large margin.

The communications subsystem should also perform energy-aware packet forwarding. The use of intelligent radio hardware enables packets to be forwarded directly from the communication subsystem, without processing it through the micro-controller. Techniques similar to DVS are used for modulation, to transmit data using a simpler modulation scheme, there by consuming less energy, when the required data transmission rate is lower. This is called modulation scaling. Besides incorporating energy-efficient algorithms at the node level, there should be a network-wide cooperation among nodes to conserve energy and increase the overall network lifetime. The computation-communication trade-off determines how much local computation is to be performed at each node and what level of aggregated data should be communicated to neighboring nodes or BSs. Traffic distribution and topology management algorithms exploit the redundancy in the number of sensor nodes to use alternate routes so that energy consumption all over the network is nearly uniform.

5.9.2 Synchronization

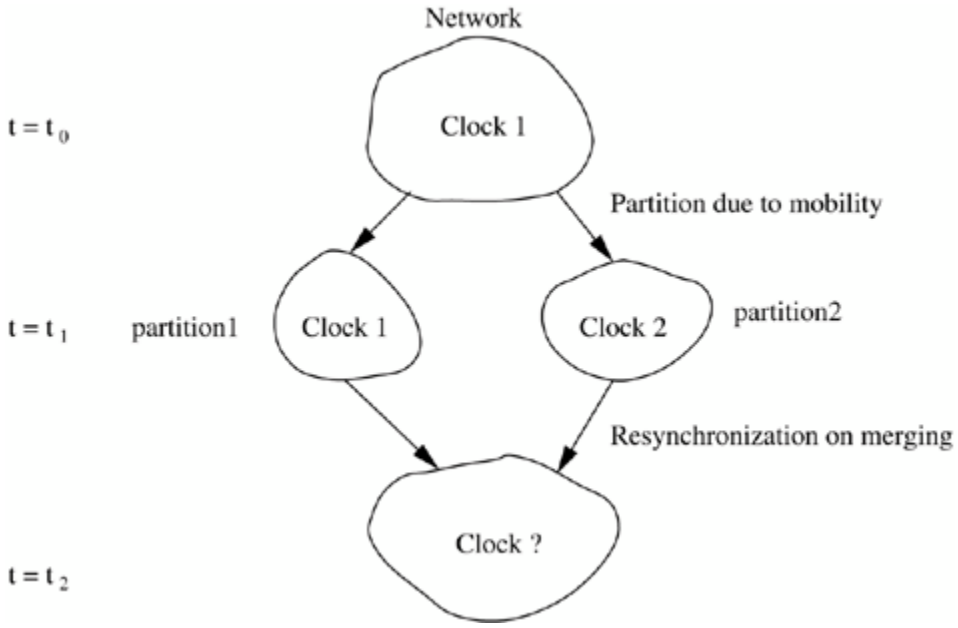
Synchronization among nodes is essential to support TDMA schemes on multi- hop wireless networks. Also, time synchronization is useful for determining the temporal ordering of messages sent from sensors and the proximity of the sensors. Usually, sensor nodes are dropped into the environment from which data has to be collected, and their exact positions are not fixed before deployment. Hence, synchronization is the only way by which the nodes can determine their relative positions. Further, in order to furnish aggregate data to the monitor node, the sensors must evolve a common timescale using their synchronized clocks, to judge the speed of a moving target or phenomenon. Sensors must be able to recognize duplicate reports of the same event by different nodes and discard them, which means that the node must be able to precisely determine the instant of time at which the event occurred. There are two major kinds of synchronization algorithms: one which achieves long-lasting global synchronization, that is, lasts throughout the network for its entire lifetime, and one which achieves a short-lived or pulse synchronization where the nodes are synchronized only for an instant. Synchronization protocols typically involve delay measurements of control packets. The delays experienced during a packet transmission can be split into four major components send time, access time, propagation time, and receive time. The send time is the time spent at the sender to construct the message. The access time is the time taken by the MAC layer to access the medium, which is appreciable in a contention-based MAC protocol. The propagation time reflects the time taken by the bits to be physically transmitted through the medium over the distance separating the sender and receiver. The receive time is the time for processing required in the receiver's network interface to receive the message from the channel and notify the host of its arrival. If the arrival time is time-stamped at a low layer, overheads of context switches and system calls are avoided, and the arrival time-stamp closely reflects the actual arrival time, with the only non-determinism introduced being due to reception of the first bit. Many existing synchronization algorithms for sensor networks rely on the time information obtained through the GPS to provide coarse time synchronization. The accuracy of time synchronization provided by GPS depends on the number of satellites observed by the GPS receiver. In the worst case, with only one observed satellite, GPS offers an accuracy of 1 μs . However, GPS is not a suitable choice for sensor networks because GPS receivers cannot be used inside large buildings and basements, or underwater, or in other satellite- unreachable environments where sensor networks may have to be deployed.

A low-power synchronization scheme called *post facto* synchronization has been proposed by Elson and Estrin in for wireless sensor networks. In this scheme, the clocks of the

nodes are normally unsynchronized. When an event is observed, a synchronization pulse is broadcast by a beacon node, with respect to which all nodes normalize their time-stamps for the observation of the event. This scheme offers short-lived synchronization, creating only an "instant" of synchronization among the nodes which are within transmission range of the beacon node. The propagation delay of the synchronization pulse is assumed to be the same for all nodes. Yoram Ofek proposed a global synchronization protocol based on exchange of control signals between neighboring nodes. A node becomes a leader when elected by a majority of nodes in the network. A distributed election protocol is used which ensures the presence of a unique leader for the network. The leader then periodically sends synchronization messages to its neighbors. These messages are broadcast in turn to all nodes of the network. The time-difference bounds have been theoretically analyzed, and fault- tolerance techniques have been added to account for errors in the synchronization messages.

A long-lasting synchronization protocol is proposed in, which ensures global synchronization of a connected network, or synchronization within connected partitions of a network. Each node in the network maintains its own local clock (real clock) and a virtual clock to keep track of its leader's clock. A unique leader is elected for each partition in the network, and virtual clocks are updated to match the leader's real clock. The leader election process occurs as follows. On power-up, every node makes an attempt to either locate a leader in its partition or claims to be a leader itself. A node decides, with a small probability, to stake a claim for leadership and announces its claim with a random number sent on the claim packet. This *Leader Announcement* packet also contains the transmission power used by the node. A node which receives this claim applies a correction for the propagation delay experienced by the claim packet (calculated based on received power), and updates its virtual clock to the expected value of the leader's real clock at that instant. Time-stamping of claims is performed at the physical layer, to avoid the variable queuing and medium access delays introduced by the MAC layer. The claim is flooded throughout the partition, bounded by a TTL field. In case two nodes within a partition stake a leadership claim, the one whose *Leader Announcement* has a higher random number resynchronizes to the leader whose *Leader Announcement* has the lower random number, and then rebroadcasts the *Leader Announcement* of the node that generated the lower random number. In the highly unlikely case of two leaders generating the same random number, node ID is used for resolution. Periodic beaconing ensures that synchronization is maintained throughout the partition, and nodes which join it later also synchronize their clocks. Resynchronization is the process of synchronizing different network partitions that are independently synchronized to different clocks to a common clock. In dynamic networks such as sensor networks, frequent changes in topology make resynchronization an important issue. Resynchronization takes place in situations such as the merging of two partitions due to mobility, where all clocks in a partition may need to be updated to match the leader of the other partition, as shown in Figure 5.19.

Figure 5.19. Resynchronization.



The typical TDMA super frame structure is shown in Figure 5.20. Pre synch frames define the start and end of a super frame, control frames transmit control information, and data frames are the TDMA time slots allotted to the nodes involved in data transfer. A positive shift in resynchronization is defined as the transmission of a data packet at an absolute time later than the slot in the current frame structure. Negative shift is defined as advancing the start of a super frame to transmit the data packets earlier than the start of transmission in the current frame structure. Resynchronization maintains slot assignment to routes through the node, but shifts the start of the super frame. If the clocks of nodes of partition1 have to be updated, the super frame can be shifted without loss of data or reconfiguration. However, if the clocks of partition 2 have to be shifted, as shown in Figure 5.20 (b), some data frames are lost due to the negative shift. If the policy of positive shift is followed uniformly, the nodes must have the capacity to buffer up to an entire super frame's data packets to start afresh with the new timing, as shown in Figure 5.20 (c). Buffering alleviates the problem of data loss on the link whose end-points are being resynchronized, but neighboring links may suffer collisions when they follow different clocks. Hence, as the resynchronization proceeds radically from the new leader, there is data loss along the head of the resynchronization wave. This remains for a time period proportional to the time taken for the *Leader Announcement* packet to propagate from the leader to the farthest node, which in turn depends on the diameter of the network, until the entire network is resynchronized. Also, different methods for transmitting the synchronization information have been studied. Out-of-band synchronization uses a separate control channel for sending claim and beacon packets. Collisions are reduced to a great extent for the control packets. However, the available bandwidth for data transmission is reduced, and the cost of the mobile nodes increases because of the need for an additional radio interface. In in-band synchronization, control information for synchronization shares the same channel with the data packets, as shown in Figure 5.21 (a). This leads to a greater number of collisions, but avoids an additional channel or bandwidth reservation. Piggy-backing can be used to reduce explicit control packets. Control information is

piggy-backed onto outgoing data packets, as in Figure 5.21 (b). This involves very low overhead with each packet and leads to considerable bandwidth saving. A control packet carrying the synchronization information is originated only if there are no data packets to be sent from the node. The scheme can also be applied with piggy- backing on the link-level acknowledgments. In sensor networks, data usually flows from all sensors to the monitor, which is a fixed node with greater computing and power resources than the sensors. If the monitor is forced to be the leader, the synchronization information moves in the reverse direction, that is, along the link-level acknowledgments sent by the nodes for each hop of the data packets, as shown in Figure 5.21 (c). Using simulation studies, this has been observed to be the most efficient mechanism.

Figure 5.20. Shifting of frames on resynchronization.

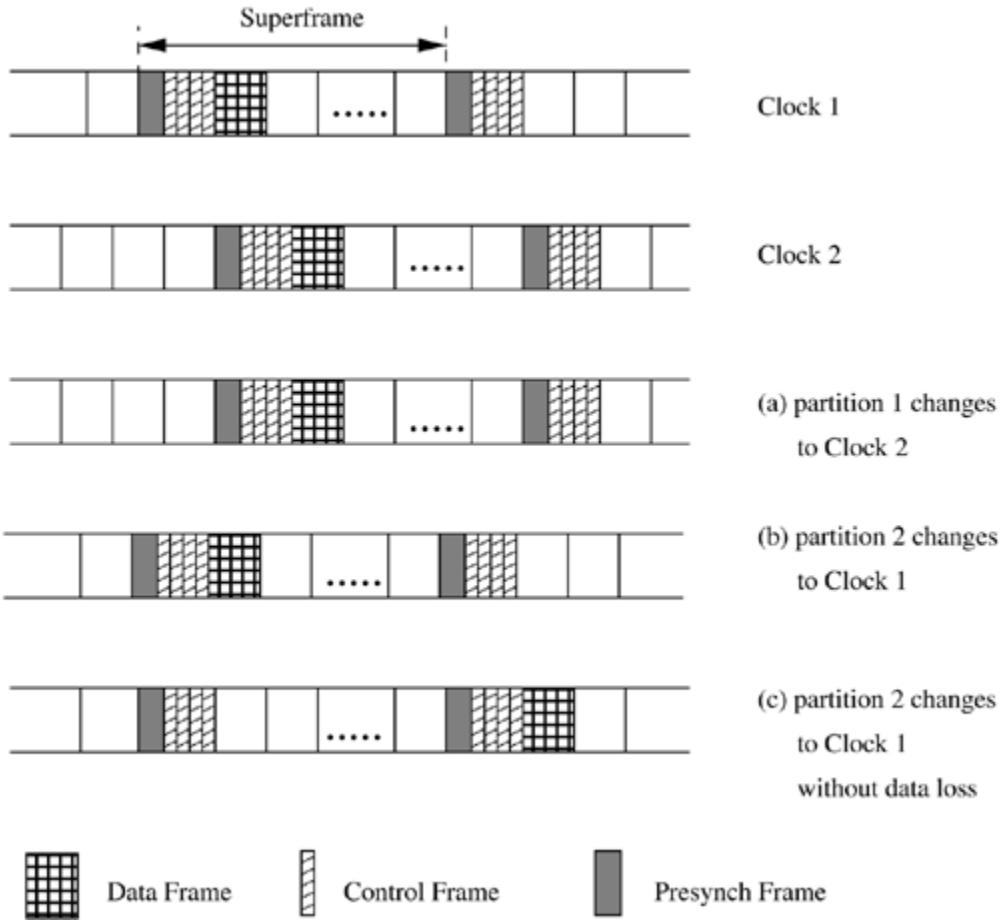
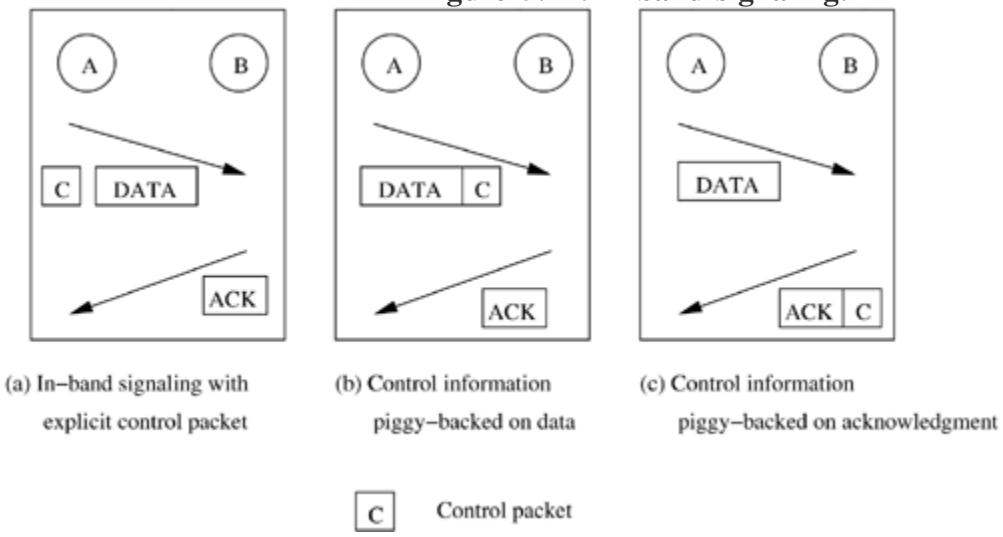


Figure 5.21. In-band signaling.



5.9.3 Transport Layer Issues

The major issue in transport layer protocols for sensor networks is the provision of reliable data delivery. This assumes special significance in the design of general-purpose sensor networks, where groups of nodes may need to be reconfigured or reprogrammed to suit an evolving application. This may require disseminating a code segment to some nodes, where loss of even a single line of code would render the re tasking operation a failure. In, a reliable, robust, scalable, and customizable transport protocol, pump slowly fetch quickly (PSFQ), is proposed. The key concept behind the protocol is that a source node distributes data at a slow rate (pump slowly), and a receiver node which experiences data loss retrieves the missing data from immediate neighbors quickly (fetch quickly). PSFQ assumes that data loss is due to poor link conditions rather than traffic congestion. It proposes a hop-by-hop error recovery scheme, rather than holding only the destination node responsible for error detection. The overhead of requiring intermediate nodes to keep track of forwarded data is justified in sensor networks, because most transmissions are intended for groups of sensors, so intermediate nodes are also intended receivers PSFQ consists of three functions: message relaying (pump), error recovery (fetch), and selective status reporting (report). The pump operation disseminates data to all target nodes, performs flow control, and localizes loss by ensuring caching at intermediate nodes. Hence, the errors on one link are rectified locally without propagating them down the entire path. When a receiver detects gaps in the received sequence numbers, a loss is indicated, and it goes into fetch mode. It requests a retransmission from neighbor nodes. An attempt is made to aggregate losses, that is, many message losses are batched into a single fetch operation, which is especially appropriate for bursty losses. PSFQ supports a report operation to provide feedback on data delivery status to the source. The farthest target node initiates its report on the reverse path of data, and all intermediate nodes append their reports to the same. Hence, PSFQ ensures that data segments are delivered to all intended receivers in a scalable and reliable manner, even in an environment where the radio link quality is poor. It has been observed that the ratio between the fetch and pump rates should be around 5 for maximum effectiveness. A recent protocol, event-to-sink reliable transport (ESRT), studies a new perspective on reliability in sensor networks. It defines event-to-sink reliability in place of the traditional end-to-end reliability provided by the

transport layer, that is, data about the event is to be carried reliably to the sink, with minimum energy expenditure. The sink is required to track reliably only the collective report about an event and not individual reports from each sensor. This enables a relaxation in stringent end-to-end reliability for each flow. The salient features of ESRT are its self-configuring capability, energy awareness, and congestion control. ESRT defines the term *observed reliability* as the number of packets that are routed from event to sink, and *required reliability* as the desired number of such packets for the event to be successfully tracked. If the observed reliability of an event falls below the requirement, ESRT increases the reporting frequency. On the other hand, if the reliability level required has been exceeded, ESRT decreases the reporting frequency in order to conserve energy. The frequency at which sensors must send their reports is conveyed to them through broadcasts from the sink, after appropriate calculations, so that the required reliability is achieved. Congestion control is achieved by monitoring buffer levels at forwarding sensors.

Reliability in the reverse direction, from sink to sensors, is discussed. The different kinds of reliability required in this direction are listed. On one hand, small queries are sent in a single packet, whereas software that needs to be updated in the sensors may be sent across multiple packets. Accordingly, reliability should be ensured for single or multiple packets, depending on the content. Further classification is based on the intended set of receivers. A message may need to be sent to all sensors of the network, or to all within a sub- area (as in a location-based query), or maybe to a subset of sensors which, among themselves, covers a certain area. In the last case, not all sensors in the area need to receive the message, but only a small subset, the union of whose coverage areas adds up to the required area, needs to reliably receive the message. One of the ways to ensure any of these forms of reliability is to use some nodes as recovery servers, which retransmit the message to sensors which did not receive it.

5.9.4 Security

Sensor networks, based on an inherently broadcast wireless medium, are vulnerable to a variety of attacks. Security is of prime importance in sensor networks because nodes assume a large amount of trust among themselves during data aggregation and event detection. From a set of sensor nodes in a given locality, only one final aggregated message may be sent to the BS, so it is necessary to ensure that communication links are secure for data exchange. The basic kinds of attacks on sensor networks at the network layer level have been listed. Cryptographic solutions based on symmetric or public key cryptography are not suitable for sensor networks, due to the high processing requirements of the algorithms. Routing protocols can be affected by spoofing or altering the routing information exchanged between nodes. This can lead to errors in routing, higher latency, or even partitioning of the network. The Sybil attack occurs when a single node presents itself as multiple entities to the network. This can affect the fault tolerance of the network and mislead geographic routing algorithms. Encryption and authentication using a globally shared key can prevent these attacks caused by an outsider trying to corrupt the messages in the network.

A selective forwarding attack is a situation when certain nodes do not forward many of the messages they receive. The sensor networks depend on repeated forwarding by broadcast for messages to propagate throughout the network. Sinkhole attacks are those which make a malicious node seem very favorable to the routing algorithm so that most data is routed through it. This node then performs selective forwarding, or acts as a "sink." Sensor networks are especially vulnerable to sinkhole attacks because most traffic is toward the BS. So, providing a

single "favorable" route is likely to influence a large number of nodes to route their data through the malicious node. The wormhole attack lures traffic through a very long path by giving false information to the nodes about the distance between them. This increases latency by avoiding some other possible shorter paths. Wormhole and sinkhole attacks are difficult to counter because routing information supplied by a node is difficult to verify. However, geographic routing protocols are not affected by these attacks since the routes are considered on demand using the location coordinates, so false distances can be verified. *Hello* flood attacks can be caused by a node which broadcasts a *Hello* packet with very high power, so that a large number of nodes even far away in the network choose it as the parent. All messages now need to be routed multi-hop to this parent, which increases delay. This can be avoided by checking the bi directionality of a link, so that the nodes ensure that they can reach their parent within one hop. The rest of this section deals with some protocols that have been proposed to improve security in sensor networks.

Localized Encryption and Authentication Protocol (LEAP)

Localized encryption and authentication protocol (LEAP) is a key management protocol (a protocol to distribute cryptographic keys) for sensor networks based on symmetric key algorithms, that is, the same key is used by sender and receiver. In a network, requiring every pair of nodes to have a shared key to be used for communication between them is ideal for security, because an attack on any one node does not compromise the security of other nodes. However, in sensor networks, the neighbors of a node may not be known in advance, hence this sharing of keys must take place after the network Ideployed, which will cause a high overhead. Also, sensor networks may employ certain processing optimizations such as a node's deciding not to report an event if it overhears its neighbor reporting the same. Such optimizations will be precluded by the usage of a separate key for each neighboring pair. On the other hand, having a common key for all nodes in the network has lower overhead, but compromise of any node affects the entire system. LEAP uses different keying mechanisms for different packets depending on their security requirements. For instance, routing information, which is usually in broadcast mode, does not require confidentiality, whereas aggregated data sent to the BS must be confidential. Every sensor node maintains four types of keys: an individual key which it shares with the BS; a group key shared with all nodes of the network and the BS; a cluster key shared between a node and its neighbors; and a pair wise shared key with each of its neighbors. The individual key is preloaded into the node before deployment, and is used for transmission of any special information between the BS and the node, such as exclusive instructions to a node, or report from a node to BS about the abnormal behavior of a neighboring node. It is assumed that the time required to attack a node is greater than the network establishment time, during which a node can detect all its immediate neighbors. A common initial key is loaded into each node before deployment. Each node derives a master key which depends on the common key and its unique identifier. Nodes then exchange *Hello* messages, which are authenticated by the receivers (since the common key and identifier are known, the master key of the neighbor can be computed). The nodes then compute a shared key based on their master keys. The common key is erased in all nodes after the establishment, and by assumption, no node has been compromised up to this point. Since no adversary can get the common key, it is impossible to inject false data or decrypt the earlier exchange messages. Also, no node can later forge the master key of any other node. In this way, pair wise shared keys are established between all immediate neighbors. The cluster key is established by a node after the pair wise key

establishment. A node generates a cluster key and sends it encrypted to each neighbor with its pair wise shared key. The group key can be preloaded, but it should be updated once any compromised node is detected. This could be done, in a naive way, by the BS's sending the new group key to each node using its individual key, or on a hop-by-hop basis using cluster keys. Other sophisticated algorithms have been proposed for the same. Further, the authors have proposed methods for establishing shared keys between multi-hop neighbors.

Intrusion Tolerant Routing in Wireless Sensor Networks (INSENS)

Intrusion tolerant routing in wireless sensor networks (INSENS)adopts a routing-based approach to security in sensor networks. It constructs routing tables at each node, bypassing malicious nodes in the network. The protocol cannot totally rule out attack on nodes, but it minimizes the damage caused to the network. The computation, communication, storage, and bandwidth requirements at nodes are reduced, but at the cost of greater computation and communication at the BS. To prevent DoS attacks, individual nodes are not allowed to broadcast to the entire network. Only the BS is allowed to broadcast, and no individual nodes can masquerade as the BS, since it is authenticated using one-way hash functions (*i.e.,* a hash function whose inverse is not easy to obtain). Control information pertaining to routing must be authenticated by the BS in order to prevent injection of false routing data. The BS computes and disseminates routing tables, since it does not face the computation and energy constraints that the nodes do. Even if an intruder takes over a node and does not forward packets, INSENS uses redundant multipath routing, so that the destination can still be reached without passing through the malicious node. INSENS has two phases: route discovery and data forwarding. During the route discovery phase, theBS sends a request message to all nodes in the network by multi-hop forwarding (not using its broadcast). Any node receiving a request message records the identity of the sender and sends the message to all its immediate neighbors if it has not already done so. Subsequent request messages are used to identify the senders as neighbors, but repeated flooding is not performed. The nodes respond with their local topology by sending feedback messages. The integrity of the messages is protected using encryption by a shared key mechanism. A malicious node can inflict damage only by not forwarding packets, but the messages are sent through different neighbors, so it is likely that it reaches a node by at least one path. Hence, the effect of malicious nodes is not totally eliminated, but it is restricted to only a few downstream nodes in the worst case. Malicious nodes may also send spurious messages and cause battery drain for a few upstream nodes. Finally, the BS calculates forwarding tables for all nodes, with two independent paths for each node, and sends them to the nodes. The second phase of data forwarding takes place based on the forwarding tables computed by the BS.

Security Protocols for Sensor Networks (SPINS)

Security protocols for sensor networks (SPINS) consists of a suite of security protocols that are optimized for highly resource-constrained sensor networks. SPINS consists of two main modules: sensor network encryption protocol (SNEP) and a micro-version of timed, efficient, streaming, loss-tolerant authentication protocol (μTESLA). SNEP provides data authentication, protection from replay attacks, and semantic security, all with low communication overhead of eight bytes per message. Semantic security means that an adversary cannot get any idea about the plaintext even by seeing multiple encrypted versions of the same plaintext. Encryption of the plaintext uses a shared counter (shared between sender and receiver). Hence, the same message is encrypted differently at different instances in time. Message integrity and confidentiality are

maintained using a message authentication code (MAC). This is similar to a checksum derived by applying an authentication scheme with a secret shared key to the message. The message can be decrypted only if the same shared key is present. The message also carries the counter value at the instance of transmission (like a time-stamp), to protect against replay attacks. μTESLA ensures an authenticated broadcast, that is, nodes which receive a packet can be assured of its sender's identity. It requires a loose time synchronization between BS and nodes, with an upper bound on maximum synchronization error. The MAC keys are derived from a chain of keys, obtained by applying a one-way function F (a one-way function is one whose inverse is not easily computable). All nodes have an initial key K_0, which is some key in the key-chain. The relationship between keys proceeds as $K_0 = F(K_1)$, $K_1 = F(K_2)$, and, in general, $K_i = F(K_{i+1})$. Given K_0, K_1, ..., K_i, it is not possible to compute K_{i+1}. The key to be used changes periodically, and since nodes are synchronized to a common time within a bounded error, they can detect which key is to be used to encrypt/decrypt a packet at any time instant. The BS periodically discloses the next verification key to all the nodes and this period is known to all nodes. There is also a specified lag of certain intervals between the usage of a key for encryption and its disclosure to all the receivers. When the BS transmits a packet, it uses a MAC key which is still secret (not yet disclosed). The nodes which receive this packet buffer it until the appropriate verification key is disclosed. But, as soon as a packet is received, the MAC is checked to ensure that the key used in the MAC has not yet been disclosed, which implies that only the BS which knows that yet undisclosed key could have sent the packet. The packets are decrypted once the key-disclosure packet is received from the BS. If one of the key-disclosure packets is missed, the data packets are buffered till the next time interval, and then authenticated. For instance, suppose the disclosure packet of K_j does not reach a node; it waits till it receives K_{j+1}, then computes $K_j = F(K_{j+1})$ and decrypts the packets received in the previous time interval.

5.9.5 Real-Time Communication

Support for real-time communication is often essential in sensor networks which are used for surveillance or safety-critical systems. The communication delay between sensing an intrusion and taking appropriate action greatly affects the quality of tracking provided by a surveillance system. Similarly, in a nuclear power plant, the detection of an abnormality in temperature or pressure must be conveyed in real-time to the control system in order to take immediate action. Hence, delay guarantees on routing would be extremely useful for such systems. Two protocols which support real-time communication in sensor networks - SPEED and RAP — are discussed in this section.

SPEED

A stateless protocol, SPEED, which supports real-time communication in sensor networks, has been proposed. SPEED is a localized algorithm which provides real-time unicast, real-time area-multicast (multicast to all nodes in a particular region), and real-time any cast support for packet transmission. SPEED has minimal overheads, as it does not require routing tables. It is compatible with best-effort MAC layer, not requiring any special MAC support. It also distributes traffic and load equally across the network using non- deterministic forwarding. The SPEED protocol requires periodic beacon transmissions between neighbors. It also uses two specific on-demand beacons for delay estimation and congestion (back-pressure) detection. These are used to adapt to changes in the network. The load at a node is approximated using

single-hop delay. The measurement is made using data packets which pass by a node, instead of separate probe packets. This minimizes the overhead. Nodes also respond using the delay estimation beacon to inform neighbors of the estimated delay. Routing of packets is performed by stateless non-deterministic geographic forwarding (SNGF). Using geographic information, packets are forwarded only to the nodes which are closer to the destination. Among the eligible closer nodes, the ones which have least estimated delay have a higher probability of being chosen a an intermediate node. If there are no nodes that satisfy the delay constraint, the packet is dropped. SPEED uses a neighbor feedback loop (NFL) to maintain the estimated delay fairly constant, so that frequent updates of delay estimates are not required. When a packet has to be dropped, that is, there is no path which can meet the delay constraint, the sending rate to the downstream nodes (nodes which are closer to the receiver) is reduced to avoid congestion, thereby maintaining the delay. The NFL issues a back-pressure beacon indicating the average delay. The increased delay is noted, and SNGF accordingly reduces the probability of selecting the congested downstream nodes for routing, until eventually the congestion eases out and delay is reduced. This can continue recursively, propagating the back-pressure from downstream to upstream nodes (nodes which are closer to the sender), to relieve congestion in a hotspot. Many geographic routing algorithms may encounter a situation when there is no node close to the destination to forward a packet. This is called a "void." SPEED uses a void-avoidance technique by issuing a back-pressure beacon with estimated delay as infinite. This will trigger a search for alternative paths. Hence, if there exists any path to a destination satisfying the delay constraint, it will be detected by SPEED. The protocol provides support for real-time communication over sensor networks by providing guarantees on the maximum delay.

RAP

RAP provides APIs for applications to address their queries. An application layer program in the BS can specify the kind of event information required, the area to which the query is addressed, and the deadline within which information is required. The underlying layers of RAP ensure that the query is sent to all nodes in the specified area, and the results are sent back to the BS. The protocol stack of RAP consists of location addressed protocol (LAP) in the transport layer, velocity monotonic scheduling (VMS) as the geographic routing protocol, and a contention-based MAC scheme that supports prioritization. LAP is a connectionless transport layer protocol which uses location to address nodes instead of a unique addressing scheme such as IP address. It supports three kinds of communication: unicast, area multicast, and area anycast. VMS is based on the concept of packet-requested velocity, which reflects both the timing and the distance constraints. Hence, requested velocity is a measure of the urgency of the packet. If a packet can travel at its requested velocity, that is, can cover the required distance within a specified time, then it can meet its deadline. VMS gives higher priority to packets which have requested higher velocities. The velocity of a packet is calculated as the ratio of the geographic distance between sender and receiver, to the deadline. Dynamic VMS recalculates the velocity at each intermediate node, so that a packet which has been slower than its requested velocity until then can be given higher priority. This is mapped onto a MAC layer priority, which is handled by the contention-based MAC layer. The protocol hence provides convenient services for application layer programs that require real-time support.

5.10 SUMMARY

Sensor networks realize an all-pervasive distributed network to create an intelligent environment. The possible applications of sensor networks are wide- ranging, from intelligent buildings and sensor-controlled chemical plants, to habitat-monitoring and covert military operations. The direction of research in sensor networks is toward overcoming the challenges of scalability, reliability, robustness, and power-efficiency, so that a variety of applications can be implemented in highly constrained scenarios Hardware design of sensor nodes needs to be further miniaturized, and power- efficient hardware and operating systems need to be developed. On the MAC layer, provisions need to be made for mobility of sensor nodes. New ideas for routing, network establishment, and maintenance are still in the development stage. Similarly, a transport layer protocol with limited power consumption and computational costs, and which is capable of interfacing with TCP or UDP, is still on the drawing board. Handling the sensed data, and development of application-spec ific query languages, would greatly help in fine-tuning the performance of sensor networks for a variety of applications.