II

Ad Hoc Wireless Networks

A sensor network consists of a large number of sensor nodes that are densely deployed either inside or close to the phenomenon. In general, the position of sensor nodes need not be engineered or predetermined. This allows for random deployment in inaccessible terrains or disaster relief operations. Sensor networks can be considered as a special type of ad hoc wireless networks, where sensor nodes are, in general, stationary. A unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are usually fitted with an onboard processor. Instead of sending the raw data to the nodes responsible for the fusion, they use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data. A sensor system normally consists of a set of sensor nodes operated on a limited battery and a base system without any battery constraint. Typically, the base station serves as the gathering point for the collected data (through fusion). The base station also broadcasts different control commands to sensor nodes.

The application areas of sensor networks include health, military, and civilian. In military application, the rapid deployment, self-organization, and fault-tolerance characteristics of sensor nodes make them a promising sensing technique for command, control, communication, computing, intelligence, surveillance, reconnaissance, and targeting systems. In health care, sensor nodes can be used to monitor patients and assist disabled patients. Other applications include managing inventory, monitoring product quality, and monitoring disaster areas.

The design factors involved in sensor networks include fault tolerance, scalability, sensor network topology, data dissemination and gathering, transmission media, and power consumptions, with most work focused on sensor network topology (or topology control), data dissemination and gathering, and power consumption. As in ad hoc wireless networks, the use of protocol stacks to address various technical issues is also a popular approach in sensor networks.

There are many similarities between ad hoc wireless networks and sensor networks, such as energy constraints and dynamic network topology. However, the number of sensor nodes can be several orders of magnitude higher than the nodes in an ad hoc wireless network, and

hence sensor nodes are more densely deployed. Also, sensor nodes are more prone to failures than those of an ad hoc wireless network. Therefore, the topology of a sensor network is changed mainly by switch-on/off of sensor nodes, whereas the topology of an ad hoc wireless network is changed by the movement of mobile hosts. In addition, sensor nodes may not have global identification because of the large amount of overhead and large number of sensors. Finally, nodes in ad hoc wireless networks are strictly peer-to-peer, whereas nodes in sensor networks form a two-level hierarchy with the base station being the gathering point.

This group includes 16 chapters, starting with a chapter of general overview. Four chapters cover four different applications of distributed algorithms: network initialization, self-organization, self-stabilization, and time synchronization. One chapter relates routing (including broadcasting) to topology control. Three chapters are devoted to sensor coverage and deployment. Two chapters deal with surveillance and detection. The issues of QoS and scalability are covered in two separate chapters. One chapter presents solutions for a special security issue. The group ends with some discussion on recent development on a new IEEE standard (IEEE 802.15.4) for sensor networks.

20

Sensor Systems: State of the Art and Future Challenges

Dharma P. Agrawal, Ratnabali Biswas, Neha Jain, Anindo Mukherjee, Sandhya Sekhar, and Aditya Gupta

20.1 Introduction

Recent technological advances have enabled distributed information gathering from a given location or region by deploying a large number of networked tiny microsensors, which are low-power devices equipped with programmable computing, multiple sensing, and communication capability. Microsensor systems enable the reliable monitoring and control of a variety of applications. Such sensor nodes networked by wireless radio have revolutionized remote monitoring applications because of its ease of deployment, ad hoc connectivity, and cost effectiveness.

20.1.1 Characteristics of Wireless Sensor Networks

A wireless sensor network is typically a collection of tiny disposable devices with sensors embedded in them. These devices, referred to as sensor nodes, are used to collect physical parameters such as light intensity, sound, temperature, etc. from the environment where they are deployed. Each node (Figure 20.1) in a sensor network includes a sensing module, a microprocessor to convert the sensor signals into a sensor reading understandable by a user, a wireless interface to exchange sensor readings with other nodes lying within its radio range, a memory to temporarily hold sensor data, and a small battery to run the device. Wireless sensors typically have a low transmission data rate. A small form factor or size of these nodes (of the order of 5 cm^3) limits the size of the battery or the total power available with each sensor node. The low cost of sensor nodes makes it feasible to have a network of hundreds or thousands of these wireless sensors.

A large number of nodes enhance the coverage of the field and the reliability and accuracy of the data retrieved (Figure 20.2). When deployed in large numbers, sensor nodes with limited radio communication range form an ad hoc network. An ad hoc network is basically a peer-to-peer multi-hop wireless network where information packets are transmitted in a store-and-forward method from source to destination, via intermediate nodes. This is in contrast to the well-known single-hop cellular network model that supports the needs of wireless communications by installing base stations (BSs) as access points. Sensor nodes are attractive due to the ease of deployment and autonomous ad hoc connectivity that eliminates the need for any human intervention or infrastructure installation. Sensor networks need to fuse data obtained from several sensors sensing a common phenomenon to provide rich, multidimensional pictures of an environment that a single powerful macrosensor, working alone may not provide.

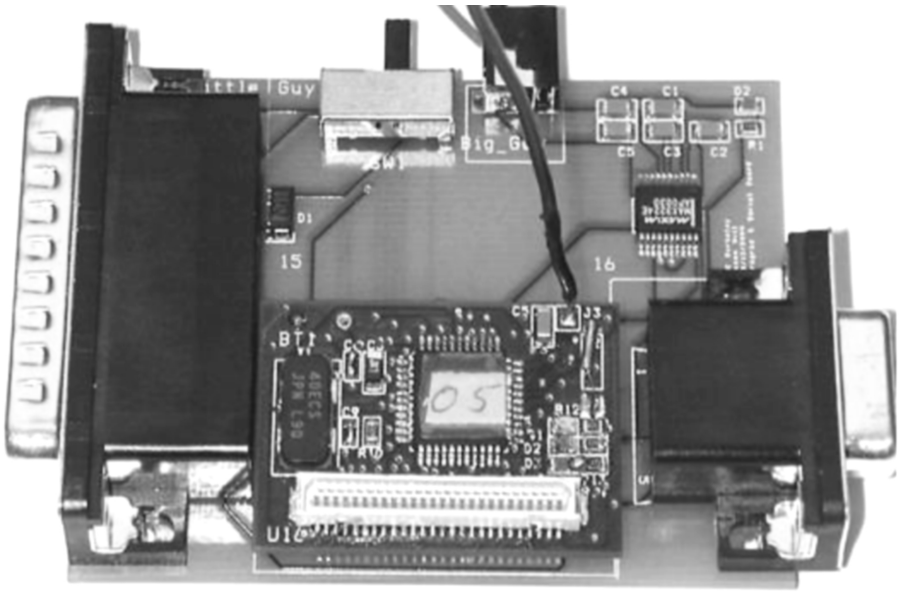
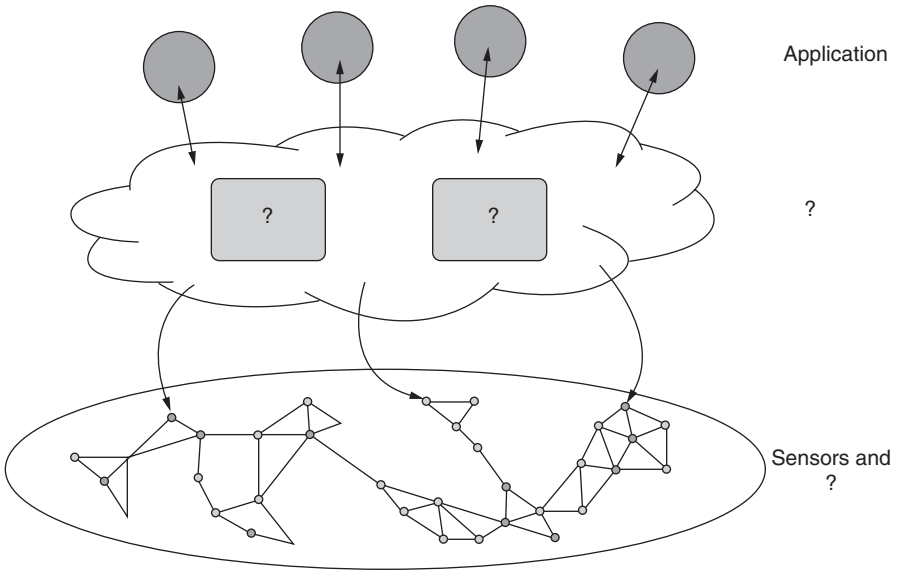


FIGURE 20.1 Sensor mote. (*Source:* www.sce.umkc.edu/~leeyu/Udic/SCE2.ppt)

Multiple sensors can help overcome line-of-sight issues and environmental effects by placing sensors close to an event of interest. This ensures a greater signal-to-noise ratio (SNR) by combining information from sources with different spatial perspectives. This is especially desirable in those applications where sensors may be thrown in an inhospitable terrain with the aid of an unmanned vehicle or a low-flying aircraft. Instead of carefully placing macrosensors in exact positions and connecting them through cables to obtain accurate results, a large number of preprogrammed sensor nodes are randomly dispersed in an environment. Although communication may be lossy due to the inherent unreliable nature of wireless links, a dense network of nodes ensures enough redundancy in data acquisition to guarantee an acceptable



Au: Not readable

FIGURE 20.2 A generic sensor network. (*Source:* eyes.eu.org/eyes-sa.gif)

FIGURE 20.3 Progression of sensors developed by CITRIS investigators. (*Source:* www.citris-uc.org/about_citris/annual_report.html)

quality of the results provided by the network. These sensor nodes, once deployed, are primarily static; and it is usually not feasible to replace individual sensors on failure or depletion of their battery. Each node has a finite lifetime determined by its rate of battery consumption. It is a formidable task to build and maintain a robust, energy-efficient multi-hop sensor network in an ad hoc setting without any global control. It is therefore necessary to consider the different types of sensors that are commercially available.

20.1.2 Types of Sensors

Sensor networks present unprecedented opportunities for a broad spectrum of applications such as industrial automation, situation awareness, tactical surveillance for military applications, environmental monitoring, chemical or biological detection, etc.

Figure 20.3 shows a progression of sensors developed by CITRIS (Center for Information Technology in the Interest of Society) investigators (www.citris-uc.org). These wireless sensors can be used to sense magnetic or seismic attributes in military security networks; or sense temperature or pressure in industrial sensing networks; strain, fatigue, or corrosion in civil structuring monitoring networks; or temperature and humidity in agricultural maintenance networks. Over a period of two years, the size of these sensors has decreased from a few cubic inches to a few cubic millimeters, and they can be powered using tiny solar cells or piezoelectric generators running on the minute vibrations of walls inside buildings or vehicles.

Microstrain Inc. (www.microstrain.com) has also developed a variety of wireless sensors for different commercial applications.

The SG-link™ Wireless Strain Gauge system (Figure 20.4) is a complete wireless strain gauge node, designed for integration with high-speed wireless sensor networks. It can be used for high-speed strain, load, torque, and pressure monitoring and finds application in sports performance and sports medicine analysis.

The TC-Link™ Wireless Thermocouple System (Figure 20.5) is a complete, cold junction compensated, multichannel thermocouple node designed to operate as part of an integrated, scalable, ad hoc wireless sensor network system. It finds application in civil structures sensing (concrete maturation), industrial

FIGURE 20.4 SG-link™ wireless strain gauge system. (*Source:* www.microstrain.com/SG-link.htm)

FIGURE 20.5 TC-Link$^{T^{TM}}$ wireless thermocouple system. (*Source:* www.microstrain.com/TCLink.htm)

sensing networks (machine thermal management), food and transportation system (refrigeration, freezer monitoring), and advanced manufacturing (plastics processing, composite cure monitoring).

The miniature EmbedSense™ wireless sensor (Figure 20.6) is a tiny wireless sensor and data acquisition system that is small enough to be embedded in a product, enabling the creation of smart structures, smart materials, and smart machines. A major advantage is that batteries are completely eliminated, thereby ensuring that the embedded sensors and EmbedSense node can be queried for the entire life of the structure. EmbedSense uses an inductive link to receive power from an external coil and to return digital strain, temperature, and unique ID information. Applications range from monitoring the healing of the spine to testing strains and temperatures on jet turbine engines.

We need to consider how a query can be processed in a sensor network.

20.2 Routing in Wireless Sensor Networks

There are a few inherent limitations of wireless media, such as low bandwidth, error-prone transmissions, the need for collision-free channel access, etc. These wireless nodes also have only a limited amount of energy available to them, because they derive energy from a personal battery and not from a constant power supply. Furthermore, because these sensor nodes are deployed in places where it is difficult to replace the nodes or their batteries, it is desirable to increase the longevity of the network. Also, preferably all the nodes should die together so that one can replace all the nodes simultaneously in the whole area. Finding individual dead nodes and then replacing those nodes selectively would require preplanned deployment and eliminate some advantages of these networks. Thus, the protocols designed for these networks must strategically distribute the dissipation of energy, which also increases the average life of the overall system.

20.2.1 Query Classification in Sensor Networks

Before discussing routing protocols for sensor networks, let us first categorize the different kinds of queries that can be posed to a sensor network. Based on the temporal property of data (i.e., whether the user is

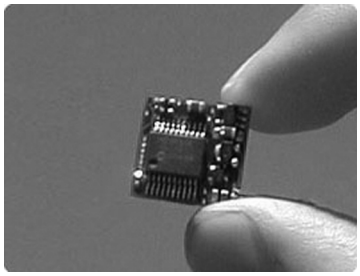


FIGURE 20.6 EmbedSense™ wireless sensor. (*Source:* www.microstrain.com/embed_sense.htm)

interested in data collected in the past, the current snapshot view of the target regions, or sensor data to be generated in future for a given interval of time), queries can be classified as follows:

Historical queries. This type of query is mainly used for analysis of historical data stored at a remote base station or any designated node in the network in the absence of a base station. For example, "What was the temperature two hours prior in the northwest quadrant?" The source nodes need not be queried to obtain historical data as it is usually stored outside the network or at a node equidistant from all anticipated sinks for that data.

One-time query. One-time or snapshot queries provide the instantaneous view of the network. For example, "What is the temperature in the northwest quadrant now?" The query triggers a single query response; hence, data traffic generated by one-time queries is the least. These are usually time critical as the user wants to be notified immediately about the current situation of the network. A warning message that informs the user of some unusual activity in the network is an example of one-time query response that is time critical.

Persistent. Persistent, or long-running, queries are mainly used to monitor a network over a time interval with respect to some parameters. For example, "the temperature in the northwest quadrant for the next 2 hours." A persistent query generates maximum query responses in the network, depending on its duration. The purpose of the persistent query is to perform periodic background monitoring. Energy efficiency is often traded with delay in response time of persistent queries to maximize utilization of network resources, as they are usually noncritical.

20.2.2 Characteristics of Routing Protocols for Sensor Networks

Traditional routing protocols defined for wireless ad hoc networks (Broch et al., 1998; Royer and Toh, 1999) are not well suited for wireless sensor networks due to the following reasons (Manjeshwar and Agrawal, 2001, 2002:

Sensor networks are data centric. Traditional networks usually request data from a specific node but sensor networks request data based on certain attributes such as, "Which area has temperature greater than 100°F?"

In traditional wired and wireless networks, each node is given a unique ID, which is used for routing. This cannot be effectively used in sensor networks because being data centric they do not require routing to and from specific nodes. Also, the large number of nodes in the network implies large IDs (Nelson and Estrin, 2000), which might be substantially larger than the actual data being transmitted.

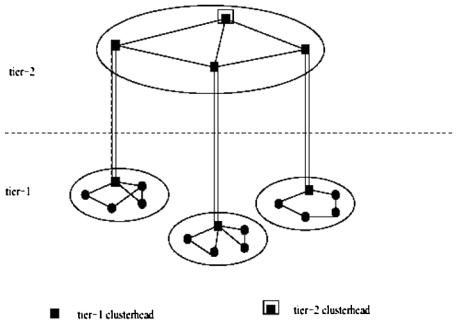
Adjacent nodes may have similar data. So instead of sending data separately from each node to the requesting node, it is desirable to aggregate similar data before sending it.

The requirements of the network change with the application and, hence, it is application specific (Estrin et al. 1999). For example, some applications need the sensor nodes to be fixed and not mobile, while others may need data based only on one selected attribute (i.e., here the attribute is fixed).

Thus, sensor networks need protocols that are application specific, data centric, and capable of aggregating data and minimizing energy consumption.

20.3 Network Architecture

The two main architecture alternatives used for data communication in a sensor network are the hierarchical and the flat network architectures. The hierarchical network architecture is energy efficient for collecting and aggregating data within a large target region, where each node in the region is a source node. Hence, hierarchical network protocols are used when data is will be collected from the entire sensor network. A flat network architecture is more suitable for transferring data between a source destination pair separated by a large number of hops.



Au: Not readable

FIGURE 20.7 Two-tier sensor network architecture. (*Source:* A. Manjeshwar, Energy Efficient Routing Protocols with Comprehensive Information Retrieval for Wireless Sensor Networks, M.S. thesis, 2001.)

20.3.1 Hierarchical Network Architecture

One way of minimizing the data transmissions over long distances is to cluster the network so that signaling and control overheads can be reduced, while critical functions such as media access, routing, and connection setup could be improved. While all nodes typically function as switches or routers, one node in each cluster is designated as the cluster head (CH) and traffic between nodes of different clusters must always be routed through their respective CHs or gateway nodes that are responsible for maintaining connectivity among neighboring CHs. The number of tiers within the network can vary according to the number of nodes, resulting in hierarchical network architecture as shown in Figure 20.7.

Figure 20.7 shows two tiers of cluster heads where the double lines represent that CHs of tier-1 are cluster members of the cluster at the next higher level (i.e., tier-2). A proactive clustering algorithm for sensor networks called LEACH is one of the initial data-gathering protocols introduced by MIT researchers Heinzelman et al. (2000). Each cluster has a CH that periodically collects data from its cluster members, aggregates it, and sends it to an upper-level CH. Only the CH needs to perform additional data computations such as aggregation, etc., and the rest of the nodes sleep unless they have to communicate with the CH. To evenly distribute this energy consumption, all the nodes in a neighborhood take turns to become the CH for a time interval called the cluster period.

20.3.2 Flat Network Architecture

In a flat network architecture as shown in Figure 20.8, all nodes are equal and connections are set up between nodes that are in close proximity to establish radio communications, constrained only by connectivity conditions and security limitations. Route discovery can be carried out in sensor networks using flooding that does not require topology maintenance as it is a reactive way of disseminating information. In flooding, each node receiving data packets broadcasts until all nodes or the node at which the packet was originated gets back the packet. But in sensor networks, flooding is minimized or avoided as nodes could receive multiple or duplicate copies of the same data packet due to nodes having common neighbors or sensing similar data. Intanagonwiwat et al. (2000) have introduced a data dissemination paradigm called directed diffusion for sensor networks, based on a flat topology. The query is disseminated (flooded) throughout the network with the querying node acting as a source and gradients are set up toward the requesting node to find the data satisfying the query. As one can observe from Figure 20.8, the query is propagated toward the requesting node along multiple paths shown by the dashed lines. The arcs show how the query is directed toward the event of interest, similar to a ripple effect. Events (data) start flowing toward the requesting node along multiple paths. To prevent further flooding, a small number of paths can be reinforced (shown by dark lines in the figure) among a large number of paths initially explored to form the multi-hop routing infrastructure so as to prevent further flooding. One advantage of flat networks is the ease of creating multiple paths between communicating nodes, thereby alleviating congestion and providing robustness in the presence of failures.

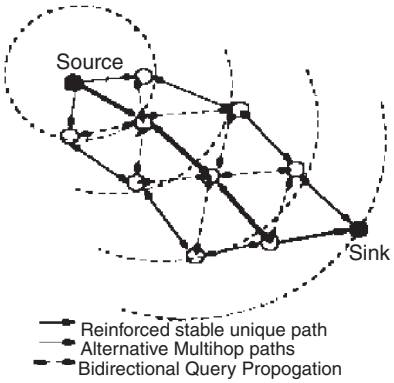


FIGURE 20.8 A flat sensor network that uses directed diffusion for routing. (*Source:* N. Jain, Energy Efficient Information Retrieval in Wireless Sensor Networks, Ph.D. thesis, 2004.)

20.4 Classification of Sensor Networks

Sensor networks can be classified into two types based on their mode of operation or functionality and the type of target applications (Manjeshwar and Agrawal, 2001, 2002):

Proactive networks. In this scheme, the nodes periodically switch on their sensors and transmitters, sense the environment, and transmit the data of interest. Thus, they provide a snapshot of the relevant parameters at regular intervals and are well suited for applications that require periodic data monitoring.

Reactive networks. In this scheme, the nodes react immediately to sudden and drastic changes in the value of a sensed attribute and are well suited for time-critical applications.

Once we have a network, we have to come up with protocols that efficiently route data from the nodes to the users, preferably using a suitable MAC (Medium Access Control) sub-layer protocol to avoid collisions.

Having classified sensor networks, we now look at some of the protocols for sensor networks (Manjeshwar and Agrawal, 2001, 2002).

20.4.1 Proactive Network Protocol

20.4.1.1 Functioning

At each cluster change time, once the cluster heads are decided, the cluster head broadcasts the following parameters (see Figure 20.9):

Report time (T_R): the time period between successive reports sent by a node.

Attributes (A): a set of physical parameters about which the user is interested in obtaining data.

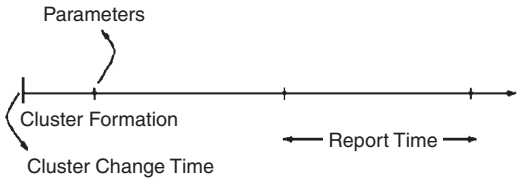


FIGURE 20.9 Timeline for proactive protocol. (*Source:* A. Manjeshwar and D.P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, in *Proc. 15th Int. Parallel and Distributed Processing Symp. (IPDPS'01) Workshop*, 2001.)

At every report time, the cluster members sense the parameters specified in the attributes and send the data to the cluster head. The cluster head aggregates this data and sends it to the base station or a higher-level cluster head, as the case may be. This ensures that the user has a complete picture of the entire area covered by the network.

20.4.1.2 Important Features

Because the nodes switch off their sensors and transmitters at all times except the report times, the energy of the network is conserved.

At every cluster change time, T_R and A are transmitted afresh and thus can be changed. By changing A and T_R, the user can decide what parameters to sense and how often to sense them. Also, different clusters can sense different attributes for different T_R.

This scheme, however, has an important drawback. Because of the periodicity with which the data is sensed, it is possible that time-critical data may reach the user only after the report time, making this scheme ineffective for time-critical data sensing applications.

20.4.1.3 LEACH

LEACH (Low-Energy Adaptive Clustering Hierarchy) is a family of protocols developed by Heinzelman et al. (www-mtl.mit.edu/research/icsystems/uamps/leach). LEACH is a good approximation of a proactive network protocol, with some minor differences.

Once the clusters are formed, the cluster heads broadcast a TDMA schedule giving the order in which the cluster members can transmit their data. The total time required to complete this schedule is called the frame time T_F. Every node in the cluster has its own slot in the frame, during which it transmits data to the cluster head. When the The reporttime T_R discussed earlier is equivalent to the frametime T_F in LEACH. However T_F is not broadcast by the cluster head but is derived from the TDMA schedule and hence is not under user control. Also, the attributes are predetermined and not changed after initial installation.

20.4.1.4 Example Applications

This network can be used to monitor machinery for fault detection and diagnosis. It can also be used to collect data about temperature (or pressure, moisture, etc.) change patterns over a particular area.

20.4.2 Reactive Network Protocol: TEEN

A new network protocol called TEEN (***T**hreshold sensitive **E**nergy **E**fficient sensor **N**etwork*) has been developed that targets reactive networks and is the first protocol developed for reactive networks (Manjeshwar and Agrawal, 2001).

20.4.2.1 Functioning

In this scheme, at every cluster change time, in addition to the attributes, the cluster head broadcasts the following to its members (see Figure 20.10):

Hard threshold (H_T): a threshold value for the sensed attribute. It is the absolute value of the attribute, beyond which the node sensing this value must switch on its transmitter and report to its cluster head.

Soft threshold (S_T): a small change in the value of the sensed attribute that triggers the node to switch on its transmitter and transmit.

The nodes sense their environment continuously. The first time a parameter from the attribute set reaches its hard threshold value, the node switches on its transmitter and sends the sensed data. The sensed

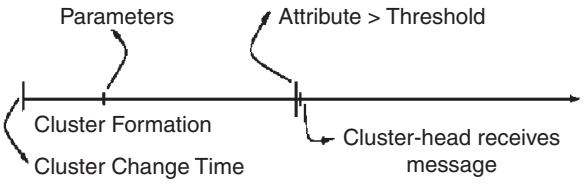


FIGURE 20.10 Timeline for TEEN. (*Source:* A. Manjeshwar and D.P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, in *Proc. 15th Int. Parallel and Distributed Processing Symp. (IPDPS'01) Workshops*, 2001.)

value is also stored in an internal variable in the node, called the *sensed value (SV)*. The nodes will next transmit data in the current cluster period but only when *both* the following conditions are true:

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

Whenever a node transmits data, *SV* is set equal to the current value of the sensed attribute. Thus, the hard threshold tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions by eliminating all the transmissions that might have otherwise occurred when there is little or no change in the sensed attribute once the hard threshold is reached.

20.4.2.2 Important Features

Time-critical data reaches the user almost instantaneously and hence this scheme is well suited for time-critical data sensing applications.
Message transmission consumes much more energy than data sensing. So, although the nodes sense continuously but because they transmit less frequently, the energy consumption in this scheme can be much less than that in the proactive network.
The soft threshold can be varied, depending on the criticality of the sensed attribute and the target application.
A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption. Thus, the user can control the trade-off between energy efficiency and accuracy.
At every cluster change time, the parameters are broadcast afresh and thus the user can change them as required.

The main drawback of this scheme is that if the thresholds are not reached, the nodes will never communicate. Thus, the user will not get any data from the network and will not even know if all the nodes die. Hence, this scheme is not well suited for applications where the user needs to get data on a regular basis. Another possible problem with this scheme is that a practical implementation would have to ensure that there are no collisions in the cluster. TDMA scheduling of the nodes can be used to avoid this problem. This will, however, introduce a delay in the reporting of time-critical data. CDMA is another possible solution to this problem.

20.4.2.3 Example Applications

This protocol is best suited for time-critical applications such as intrusion detection, explosion detection, etc.

20.4.3 Hybrid Networks

There are applications in which the user might need a network that reacts immediately to time-critical situations and also gives an overall picture of the network at periodic intervals to answer analysis queries. Neither of the above networks can do both jobs satisfactorily and they have their own limitations.

Manjeshwar and Agrawal (2001) have combined he best features of the proactive and reactive networks, while minimizing their limitations, to create a new type of network called a *hybrid network*. In this network, the nodes not only send data periodically, but also respond to sudden changes in attribute values. A new routing protocol (Adaptive Periodic Threshold-sensitive Energy Efficient Sensor Network Protocol; APTEEN) has been proposed for such a network and uses the same model as the above protocols but with the following changes. APTEEN works as follows.

20.4.3.1 Functioning

In each cluster period, once the cluster heads are decided, the cluster head broadcasts the following parameters (see Figure 20.11):

Attributes (A): a set of physical parameters about which the user is interested in obtaining data.

Thresholds: this parameter consists of a hard threshold (H_T) and a soft threshold (S_T). H_T is a particular value of an attribute beyond which a node can be triggered to transmit data. S_T is a small change in the value of an attribute that can trigger a node to transmit.

Schedule: a TDMA schedule similar to the one used in Heinzelman et al. (2000), assigning a slot to each node.

Count time (T_C): the maximum time period between two successive reports sent by a node. It can be a multiple of the TDMA schedule length and it introduces the proactive component in the protocol.

The nodes sense their environment continuously. However, only those nodes that sense a data value at or beyond the hard threshold will transmit. Furthermore, once a node senses a value beyond H_T, it next transmits data only when the value of that attribute changes by an amount equal to or greater than the soft threshold S_T. The exception to this rule is that if a node does not send data for a time period equal to the count time T_C, it is forced to sense and transmit the data, irrespective of the sensed value of the attribute. Because nodes near each other may fall into the same cluster and sense similar data, they may try sending their data simultaneously, leading to collisions between their messages. Hence, a TDMA schedule is used and each node in the cluster is assigned a transmission slot, as shown in Figure 20.11.

20.4.3.2 Important Features

It combines both proactive and reactive policies. By sending periodic data, it gives the user a complete picture of the network, like a proactive scheme. It behaves like a reactive network also by sensing data continuously and responding to drastic changes immediately.

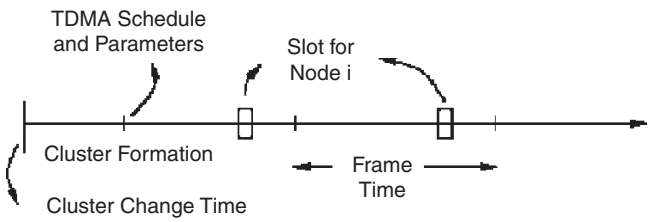
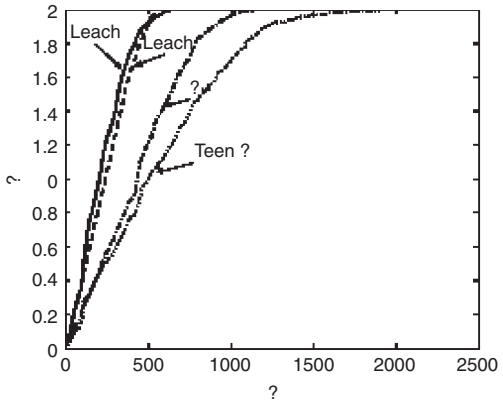


FIGURE 20.11 Timeline for APTEEN. (*Source:* A. Manjeshwar and D.P. Agrawal, APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in *Proc. Int. Parallel and Distributed Processing Symp. (IPDPS'02) Workshops*, 2002.)



Au: Not readable

FIGURE 20.12 Comparison of average energy dissipation. (*Source:* A. Manjeshwar and D.P. Agrawal, APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in *Proc. Int. Parallel and Distributed Processing Symp. (IPDPS'02) Workshops*, 2002.)

It offers a lot of flexibility by allowing the user to set the time interval (T_C) and the threshold values (H_T and S_T) for the attributes.

Energy consumption can be controlled by changing the count time as well as the threshold values.

The hybrid network can emulate a proactive network or a reactive network, based on the application, by suitably setting the count time and the threshold values.

The main drawback of this scheme is the additional complexity required to implement the threshold functions and the count time. However, this is a reasonable trade-off and provides additional flexibility and versatility.

20.4.4 A Comparison of the Protocols

To analyze and compare the protocols TEEN and APTEEN with LEACH and LEACH-C, consider the following metrics (Manjeshwar and Agrawal, 2001, 2002):

Average energy dissipated: shows the average dissipation of energy per node over time in the network as it performs various functions such as transmitting, receiving, sensing, aggregation of data, etc.

Total number of nodes alive: indicates the overall lifetime of the network. More importantly, it gives an idea of the area coverage of the network over time.

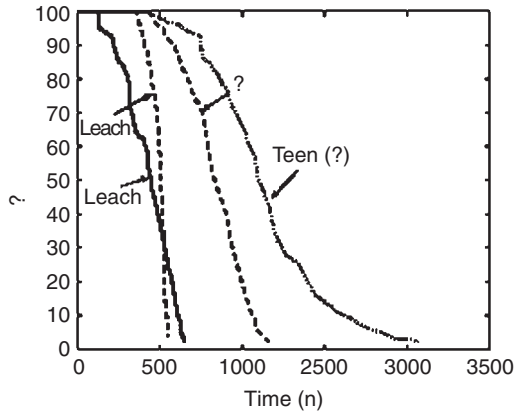
Total number of data signals received at BS: explains how TEEN and APTEEN save energy by not transmitting data continuously, which is not required (neither time critical nor satisfying any query). Such data can be buffered and later transmitted at periodic intervals. This also helps in answering historical queries.

Average delay: gives the average response time in answering a query. It is calculated separately for each type of query.

The performance of the different protocols is given in Figures 20.12 and 20.13.

20.5 Multiple Path Routing

The above protocols considered a hierarchical network; now consider multiple path routing in a flat sensor network. Multiple path routing aims to exploit the connectivity of the underlying physical networks by providing multiple paths between source destination pairs. The originating node therefore has a choice of more than one potential path to a particular destination at any given time.



Au: Not readable

FIGURE 20.13 Comparison of the number of alive nodes, (*Source:* A. Manjeshwar and D.P. Agrawal, APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in *Proc. Int. Parallel and Distributed Processing Symp. (IPDPS'02) Workshops*, 2002.)

20.5.1 The Need for Multiple Path Routing

Classical multiple path routing has been explored for two reasons. The first is *load balancing*, where traffic between the source and destination is split across multiple (partially or fully) disjoint paths to avoid congestion on any one path. The second use of multipath routing is to increase the probability of *reliable data delivery* due to the use of independent paths. To ensure reliable data delivery, duplicate copies of the data can be sent along alternate routes.

Multiple path routing is popularly used to avoid disparity in energy consumption in the network. This suggests that a multiple path scheme would be preferable when there are simultaneous active sources in the network with high traffic intensity. It is typical to have a number of overlapping source sink pairs unevenly distributed in the sensor field. It is a challenging problem to distribute traffic load evenly among majority of the sensor nodes in the network with such random traffic conditions. Multiple path routing is cost effective in heavy load scenarios, while a single path routing scheme with a lower complexity may be more desirable when the numbers of packets exchanged between the random source sink pairs are few.

Load balancing is especially useful in energy-constrained networks because the relative energy level of the nodes does affect the network lifetime more than their absolute energy levels. With classic shortest path routing schemes, a few nodes that lie on many of these shortest paths are depleted of their energy at a much faster rate than the other nodes. As a result of these few dead nodes, the nodes in its neighborhood may become inaccessible, which in turn causes a ripple effect, leading to network partitioning. Chang and Tassiulas (2000) have proved, assuming each node to have a limited lifetime, that the overall lifetime of the network can be improved if the routing protocol minimizes the disparity in the residual energy of every node, rather than minimizing the total energy consumed in routing. In the multiple path routing protocol proposed by Jain et al. (2003a,b), the traffic is spread over the nodes lying on different possible paths between the source and the sink, in proportion to their residual energy. The rationale behind traffic spreading is that for a given total energy consumption in the network, every node should have spent the same amount of energy for data transmission. Their objective is to assign more load to underutilized paths and less load to overcommitted paths so that uniform resource utilization of all available paths can be ensured. They construct multiple paths of variable energy cost, and then design a traffic scheduling algorithm that determines the order of path utilization to enable uniform network resource utilization. They also grade the multiple paths obtained according to their quality-of -ervice (QoS), where the QoS metric is delay in response time (Jain et al., 2003a,b). Thus, they use a reservation-based scheme to provide a good service to time-critical applications, along

with dynamic reallocation of network resources to noncritical applications to avoid underutilization of resources.

There is a need to adapt multiple path routing to overcome the design constraints of a sensor network. Important design considerations that drive the design of sensor networks are the energy efficiency and scalability (Hill et al., 2000) of the routing protocol. Discovery of all possible paths between a source and a sink might be computationally exhaustive for sensor networks because they are power constrained. In addition, updating the source about the availability of these paths at any given time might involve considerable communication overhead. The routing algorithm designed for a sensor network must depend only on local information (Ganesan et al., 2002a) or the information piggy-backed with data packets, as global exchange of information is not a scalable solution because of the sheer number of nodes.

20.5.2 Service Differentiation

Service differentiation is a basic way to provide QoS by giving one user priority over another. The data traffic is classified based on the QoS demands of the application. QoS parameters for typical Internet applications include bounds for bandwidth, packet delay, packet loss rate, and jitter. Certain additional parameters that deal with problems unique to wireless and mobile networks are power restrictions, mobility of nodes, and unreliable link quality. For a sensor network, node mobility is not high but severe power restrictions may force the packet to be routed through longer paths that have a higher residual energy than the shortest route connecting the source to the destination. INSIGNIA, a service differentiation methodology developed by Lee and Campbell (1998), employs a field in the packet to indicate the availability of resources and perform admission control. Such a testing is based on the measured channel capacity to the destination or utilization and the requested bandwidth. When a node accepts a request, resources are committed and subsequent packets are scheduled accordingly. If adequate resources are not available, then a flow adaptation is triggered to adjust the available resources on the new requested path.

Service requirements could be diverse in a network infrastructure. Some queries are useful only when they are delivered within a given timeframe. Information provided may have different levels of importance; therefore, the sensor network should be willing to spend more resources in disseminating packets carrying more important information. Service differentiation is popularly used in the Internet (Vutukury, 2001) to split the traffic into different classes based on the QoS desired by each class. In the multiple path routing protocol proposed by Jain et al. (2003b), a priority and preemption mechanism is used to control end-to-end delay for time-critical queries. Some specific examples of applications that could benefit from a sensor network supporting service differentiation are described here.

Battlefield surveillance. Soldiers conduct periodic monitoring for situational awareness of the battlefield. If the network senses some unusual or suspicious activity in the field that requires immediate attention of the military personnel, an alarm is triggered. These warning signals must reach the end user or the soldier immediately to expedite quick decision making.

Disaster relief operations. In case of natural calamities such as floods, wild fires, tornadoes or earthquakes, or other catastrophes such as terrorist attacks, coordinated operation of sensor nodes could be very useful in conducting efficient rescue operations. Precise information about the location of victims or environmental parameters of risky areas could provide facts that help in the planning of rescue operations. Here, a flexibility to prioritize information retrieval could be beneficial in avoiding communication delays for time-critical responses.

Infrastructure security. A network of sensors could be deployed in a building or a campus that needs to be secured from any intrusion detection. The sensors could be programmed to discriminate among the attacks if they occur simultaneously or associate higher priority for packets confirming intrusion, as compared to other normal data packets containing information related to the usual background monitoring of the building for parameters like light intensity, temperature, or the number of people passing by.

Environmental or biomedical applications. Monitoring presence of certain gases or chemicals in remote areas such as mines, caves or under water, or in chambers where research experiments are carried out, or radiation levels in a nuclear plant. Alarm signals might be required infrequently to report the presence of attributes in a volume or degree at more than an expected threshold. These warning messages, if received on time, can help in accomplishing the research goals or desired monitoring operations.

20.5.3 Service Differentiation Strategies for Sensor Networks

Bhatnagar et al. (2001) discussed the implications of adapting these service differentiation paradigms from wired networks to sensor networks. They suggest the use of adaptive approaches; the sensor nodes learn the network state using eavesdropping or by explicit state dissemination packets. The nodes use this information to aid their forwarding decisions; for example, low-priority packets could take a longer route to make way for higher-priority packets through shorter routes. The second implication of their analysis is that the applications should be capable of adapting their behavior at runtime based on the current allocation, which must be given as a feedback from the network to the application.

In our work (Manjeshwar and Agrawal, 2003a,b), we aim to achieve twofold goals of the QoS-aware routing described by Chen et al. (1998). The two goals are (1) selecting network paths that have sufficient resources to meet the QoS requirements of all admitted connections, and (2) achieving global efficiency in resource utilization. In ad hoc networks, static provisioning is not enough because node mobility necessitates dynamic allocation of resources. In sensor networks, although user mobility is practically absent, dynamic changes in the network topology may be present because of node loss due to battery outage. Hence, multi-hop ad hoc routing protocols must be able to adapt to the variation in the route length and its signal quality while still providing the desired QoS. It is difficult to design provisioning algorithms that achieve simultaneously good service quality as well as high resource utilization. Because the network does not know in advance where packets will go, it will have to provision enough resources to all possible destinations to provide high service assurance. This results in severe underutilization of resources.

20.6 Continuous Queries in Sensor Networks

Multiple path routing is used for uniform load distribution in a sensor network with heavy traffic. Now consider energy optimization techniques for sensor networks serving continuous queries. Queries in a sensor network are spatio-temporal; that is, the queries are addressed to a region or space for data, varying with time. In a monitoring application, the knowledge of the coordinates of the event in the space, and its time of occurrence is as important as the data itself. Queries in a sensor network are usually location based; therefore, each sensor should be aware of its own location (HighTower and Borreillo, 2001). When self-location by GPS is not feasible or too expensive, other means of self-location, such as relative positioning algorithms (Doherty et al., 2001), can be used. A timestamp associated with the data packet reveals the temporal property of data.

Depending on the nature of the application, the types of queries injected in a sensor network can vary. Queries posed to a sensor network are usually classified as one-time queries or periodic queries. "One-time" queries are injected at random times to obtain a snapshot view of the data attributes, but "periodic" queries retrieve data from the source nodes after regular time intervals. An example of a periodic query is, "Report the observed temperature for the next week at the rate of one sample per minute." We now concentrate on periodic queries that are long running; that is, they retrieve data from the source nodes for a substantially long duration, possibly the entire lifetime of the network. We now classify queries into three different categories based on the nature of data processing demanded by the application.

Simple queries. These are stand alone queries that expect an answer to a simple question from all or a set of nodes in the network. For example, "Report the value of temperature."

Aggregate queries. These queries require collaboration among sensor nodes in a neighborhood to aggregate sensor data. Queries are addressed to a target region consisting of many nodes in a geographically bounded area instead of individual nodes. For example, "Report the average temp of all nodes in region X."

Approximate queries. These are queries that require data summarization and rely on synopsis data structures to perform holistic data aggregation (Ganeson et al., 2002a) in the form of histograms, isobars, contour maps, tables, or plots. For example, "Report the contours of the pollutants in the region X." Such sophisticated data representation results in a tremendous reduction in data volume at the cost of additional computation at nodes. Although offline data processing by the user is eliminated, due to a lack of raw sensor data, the user may not be able to analyze it later in ways other than the query results.

Au: Is this the correct ref. citation?

Complex queries. If represented in SQL, these queries would consist of several joins nested or condition-based sub-queries. Their computation hierarchy is better represented by a query tree. For example, "Among regions X and Y, report the average pressure of the region that has higher temperature." Sub-queries in a complex query could be simple, aggregate, or approximate queries. In our proposed work, we design a general in-network query processing architecture for evaluating complex queries.

20.6.1 The Design of a Continuous Query Engine

We now present the four basic design characteristics of a continuous query processing architecture:

Data buffering. It is required to perform blocking operations (Babcock et al., 2002) that need to process an entire set of input tuples before an output can be delivered. Examples of blocking operators are GroupBy, OrderBy, or aggregation functions such as maximum, count, etc. A time-based sliding window proposed by Datar et al. (2002) is used to move across the stream of tuples and restrict the data buffered at a node at any instant of time for data processing. Time synchronization to process data also requires data buffering (Motwani et al. (2003). If the data streams arriving from different sources must be combined based on the times each tuple is generated, then synchronization between sensor nodes along the communication path is required. To ensure temporal validity of the results produced while evaluating operators, tuples arriving from the faster stream should be buffered until the tuples in the same time window belonging to the other input stream reach the QP node evaluating the operator.

Au: Correct citation?

Au: Correct citation?

Data summarization. If data is arriving too fast to be processed by the node, it will not be able to hold the incoming tuples in its limited memory. Therefore, tuples might have to be dropped or a sample of tuples could be selected from the data stream to represent the entire data set (Carney et al., 2002).

Au: Correct citation?

Sharing. It is important to share processing whenever feasible among multiple queries for maximizing the reuse of resources. This is used to achieve scalability with increasing workload.

Adaptability. In spatio-temporal querying, the number of sources and their data arrival rates may vary during a query's lifetime, thereby rendering static decisions ineffective. Hence, the continuous query evaluation (Madden et al., 2002) should keep adapting to changes in data properties.

20.6.2 Applications of an Adaptive Continuous Query Processing System

Distributed query processing has numerous applications in remote monitoring tasks over wireless sensor networks. Described here some of the real-world applications that will benefit from the proposed query processing architecture. These examples illustrate the range of applications that can be supported by the proposed query processing architecture. We classify potential commercial applications of wireless sensor networks into two broad classes.

Infrastructure-based monitoring. Sensor nodes are *attached to an existing physical structure* to remotely monitor complex machinery and processes, or the health of civil infrastructures such as highways, bridges, or buildings.

Field-based monitoring. A space in the environment is monitored using a dense network of *randomly scattered* nodes that are not particularly installed on any underlying infrastructure. Environmental monitoring of ecosystems, toxic spills, and fire monitoring in forests are examples of field monitoring.

We now describe infrastructure-based and field-based monitoring in detail.

20.6.2.1 Infrastructure-Based Monitoring

Commercial interest in designing solutions for infrastructure-based monitoring applications using sensor networks is growing at a fast pace. EmberNet (Ember Corporation; www.ember.com) has developed an embedded networking software for temperature sensing and heat trace control using wireless sensors. This drastically reduces the installation cost when the number of temperature monitoring points runs into thousands.

We propose that the efficiency of the monitoring task can be enhanced by deploying additional powerful nodes that are able to process the temperature readings and draw useful inferences within the network. As results occupy fewer data bytes than raw data, data traffic between the monitoring points and the external monitoring agency can be reduced. Data is routed faster and processing time at the external monitoring agency is saved, which enables quicker decision making based on the results of data monitoring. Another potential example of infrastructure-based monitoring is *detection of leakage* in a water distribution system. A self-learning, distributed algorithm can enable nodes to switch among various roles of data collection, processing, or forwarding among themselves so that the network continues retrieving data despite the failure of a few nodes. Human intervention would be deemed necessary only when a majority of the nodes in an area malfunction or run out of battery power. Some other infrastructure-based applications that might benefit from the proposed query processing and the resulting real-time decision support designed in this work are as follows:

Civil structure/machine monitoring. Continuous monitoring of civil structures such as bridges or towers yields valuable insight into their behavior under varying seismic activity. By examining moisture content and temperature, it is possible to estimate the maturity of concrete or a corrosive subsurface in structural components before serious damage occurs. Similar principles apply to underground pipes and drainage tiles. Another relevant application is monitoring the health of machines. Thousands of sensor nodes can track vibrations coming from various pieces of equipment to determine if the machines are about to fail.

Traffic monitoring on roadways. Sensors can be deployed on roads and highways to monitor the traffic or road conditions to enable quick notification of drivers about congestion or unfavorable road conditions so that they can select an alternate route.

Intrusion detection. Several sensor nodes can be deployed in buildings at all potential entrances and exits of the building to monitor movement of personnel and any unusual or suspicious activity.

Heat control and conditioning. Precise temperature control is very crucial for many industries, such as oil refineries and food industries. This can be achieved by placing a large number of sensors at specific places in the civil structure (such as pipes).

Intel (2003) has developed a heterogeneous network to improve the scalability of wireless sensor networks for various monitoring applications. Intel overlays a 802.11 mesh network on a sensor network analogous to a highway overlaid on a roadway system. Data is collected from local sensor nodes and transmitted across the network through the faster, more reliable 802.11 network. The network lifetime is therefore enhanced by offloading the communication overhead to the high-end 802.11 nodes. This network is capable of self reorganization in case any 802.11 node fails. Similarly, we propose to offload computation-intensive tasks from low-power sensor motes to high-performance nodes.

A few important inferences that can be drawn about infrastructure-based applications are as follows:

The layout (blueprint) of the infrastructure (like a machine or a freeway) where the sensor nodes are to be placed is known.

As nodes may have been manually placed or embedded on the structure at monitoring points while manufacturing it, the location of sensors may be known to the user.
It might be possible to provide a renewable source of power to some of the nodes.
The communication topology should adapt to the physical limitations (shape or surface area where nodes are placed) of underlying infrastructure.

20.6.2.2 Field-Based Monitoring

To perform field-based monitoring, the sensor nodes are randomly dispersed in large numbers to form a dense network to ensure sensor coverage of the environment to be monitored. The purpose is to observe physical phenomena spread over a large region, such as pollution levels, the presence of chemicals, and temperature levels. An example query for field monitoring would be: "Report the direction of movement of a cloud of smoke originating at location (x, y)." The purpose is to monitor the general level of physical parameters being observed — unlike infrastructure-based monitoring, which is usually applied for high-precision monitoring. The use of a heterogeneous network for field monitoring is not very obvious because node placement in the field usually cannot be controlled. But, at the expense of increasing the cost of deployment, the density of high-performance nodes can be increased to ensure that most of the low-power nodes in the network can access at least one high-performance node. For example, in a greenhouse, the same plant is grown in varying soil or atmospheric conditions, and the growth or health of the plant is monitored to determine the factors that promote its growth. The different soil beds can be considered the different target regions, and their sensor data is compared or combined with each other within the field to derive useful inferences to be sent to a remote server. Some other applications that might benefit from such an automated system of data monitoring are as follows.

Scientific experiments. Sensors can be randomly deployed in closed chambers in laboratories, or natural spaces such as caves or mines, to study the presence of certain gases, elements, or chemicals. Similarly, levels of radioactive materials can be observed to monitor toxic spills.
Examination of contaminant level and flow. The sensor nodes with chemical sensing capabilities can be used to monitor the levels and flow patterns of contaminants in the environment.
Habitat or ecosystem monitoring. Studying the behavior of birds, plants, and animals in their natural habitats using sensor networks has been employed on a small scale (Mainwaring et al., 2002).
Wild fire monitoring. This is particularly useful in controlling fires in forests by studying the variation in temperature over the areas affected, and the surrounding habitat.

20.7 Mobile Sensor Systems

In-situ sensors have been the traditional monitoring method for sensor networks. Integrating data from all these sensors is time-consuming and tedious. When data is analyzed, it is a view of what was and not what to expect, which is a major drawback. Hence, there is a need to have on-site configuration, and rapid collection and integration of data using mobile outfits. Sensors can be mounted onto mobile devices (e.g., a PDA or a robot) to enable mobility. These provide flexibility in data collection for dynamic and spatially extensive environments. Portable Palm pilots and PDAs are now replacing the traditional data collectors that have on-board storage. These are lightweight and long-standing devices that can work without GPS capabilities. At remote sites, mobile data collectors can provide precise field data and metadata information to the users.

20.7.1 Characteristics of Mobile Sensor Systems

There are some features that are inherent to a *mobile sensor network*. The mobile nodes must be amenable in a distributed setting. The network should be scalable with respect to communication and computation complexity. It is also important for the mobile nodes to be adaptive to hostile surroundings (see Figure 20.14). The nodes must also be reliable and the scheme involved should be asynchronous.

FIGURE 20.14 Sensors can be mounted on hand-held devices. (*Source:* www.ia.hiof.no/prosjekter/hoit/html/nr2_02/grafikk/palm-pilot.jpg)

Consider a sensor network with mobile robots — a network architecture for low-power and large-scale sensor networks. This network has two types of nodes: (1) sensors and (2) mobile robots. Sensors are low-power and low-cost nodes that have limited processing and communication capability. They are deployed in large quantity, perhaps randomly through aerial drops covering the entire network. The mobile robots are powerful hardware nodes, both in their communication or processing capability and in their ability to traverse the network. Mobile robots perform information retrieval and post processing (see Figure 20.15).

These nodes can sense various physical phenomena such as light, temperature, humidity, chemical vapors, and sound. They are deployed in large quantities, perhaps randomly through an aerial drop, covering a large area and constituting a single network. Any sensor node detecting interesting data reports to a predetermined location that receives information. This location is termed a *sink*.

20.7.2 Need for Mobile Sensor Networks

In general, sensors are low-power and low-cost nodes and the energy consumed in communicating their data is much more than that for complex computations. For these reasons, the network dies when a majority of the sensors run out of energy. Robots have high energy or can be renewed periodically because they are mobile and often report to the sink. Excellent communication between robots and sensor network is desirable for precise movement and actions. The collaboration between mobile robots and sensor networks is a key factor in achieving efficient transmission of data, network aggregation, quick detection of events, and timely action by robots. Coordination between multiple robots for resource transportation has been explored for quite some time. Transporting various types of resources for different applications such as

FIGURE 20.15 Mobile sensing environment. (*Source:* www.isiindustrysoftware.com/main.html)

defense, manufacturing process, etc. has been suggested (Vaughan et al., 2000). In these schemes, the time taken to detect an event depends entirely on the trail followed by the robots. Although the path progressively gets better with the use of an ant-type algorithm (Hayes et al., 2003; Vaughan et al., 2000), the whole process must be started anew when the position of the event changes. Suitable algorithms for the detection of events at any point in the network have not been formulated. Another drawback in these systems is that there are no sensors that guide them toward the event. Two visually guided robots can simultaneously carry resources from place to place (Schenker et al., 2001). Constant information exchange is mandatory here and the use of a single or multiple robots (more than two) has been overlooked.

Enhancements in the field of robotics are paving the way for industrial robots to be applied to a wider range of tasks. Newer robots are being developed with better control, safety measures, guidance, and robust sensing. Various mechanisms for producing dexterous motion, safety, and issues related to coordinating multiple robots have also been taking giant strides in neoteric times. Thus, it seems very roseate to look in terms of merging the two fields of sensor networks and robotics.

Robots are rapidly moving from the pages of science fiction novels to everyday life. The enhancements in the field of robotics are paving the way for industrial robots to be applied to a wider range of tasks. Advances in materials and technology have made modern robots much smaller, much lighter, and more precise, which means that there can be more applications of these robots than previously envisioned. However, harnessing their full efficiency also depends on how accurately they understand their environment. To measure physical parameters of the surroundings, different sensor devices can be applied so that useful information can be procured. Also, to get a global view, each robot needs to retrieve and aggregate information from sensors, while sensors themselves can exchange information using wireless devices.

20.7.3 Applications of Mobile Sensor Networks

Gupta et al. (2004) have analyzed the possibility of coalescing sensor networks with mobile robots. Multiple mobile robots and their coexistencewere examined, while the relative merits and demerits were also evaluated. The cooperation between the mobile robots and sensor networks is a key factor in achieving speedy in-network aggregation and transmission of data. The scheme (Gupta et al. 2004) is very efficient for monitoring queries. The authors claim that the frontiers of sensor networks will be significantly advanced by enabling mobile robots to herd them. Low-power sensor nodes are used to detect an event and guide mobile robots to such locations. These robots have been modeled as resource-carrying ones and are used to transport resources within the network. The symbiosis of two independently powerful spheres leads to the overall efficiency of the network. The adroitness of the mobile robots forms the backbone of our proposed scheme and their synchronization forms the communication channel of the network (see Figure 20.16).

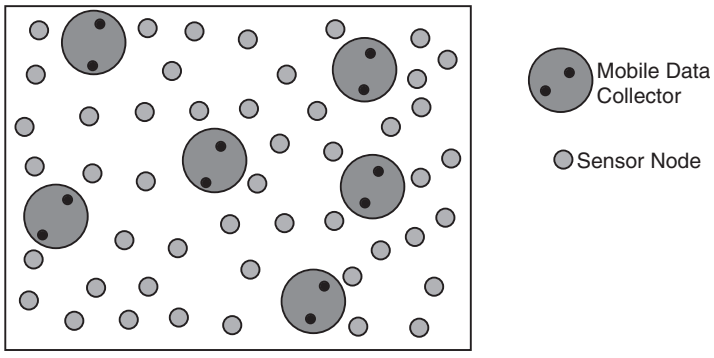


FIGURE 20.16 Heterogeneous sensor network. (*Source:* A.K. Gupta, S. Sekhar, and D.P. Agrawal, Efficient event detection by collaborative sensors and mobile robots, in *Ohio Graduate Symp. On Computer and Information Science and Engineering*, Dayton, OH, June 2004.)

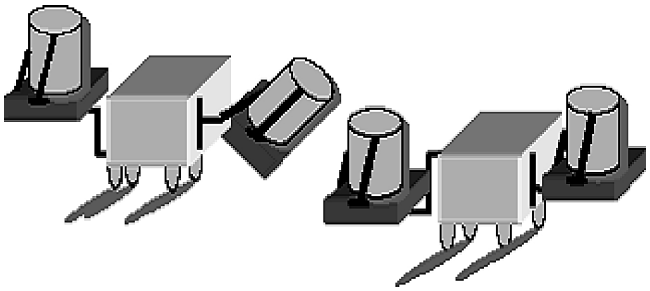


FIGURE 20.17 A resource carrying robot. (*Source:* A.K. Gupta, S. Sekhar, and D.P. Agrawal, Efficient event detection by collaborative sensors and mobile robots, in *Ohio Graduate Symp. On Computer and Information Science and Engineering*, Dayton, OH, June 2004.)

In terrains where human ingress is difficult, we use mobile robots to imitate the human's chore. These shoebox-sized robots are vibrant with energetic, immune to poison, impervious to pain, vacuum resistant, hunger tolerant, invulnerable to sleep, etc. Typical resource-carrying robots are depicted in Figure 20.17.

Figure 20.18 shows a robot transferring its resource to another. These robots have the capability to carefully transport their contents and transfer their resources to another. Once depleted of their resource, they can get themselves refilled from the sink, which is a local reservoir of resources. The resource in demand could be water or sand (to extinguish fire), oxygen supply, medicine, bullets, clothes, or chemicals to neutralize hazardous wastes, etc.

The choice of resource is limited only by the carrying capacity of the robots. The nature of the resource depends on the application that the network supports. Hence, one can see that there are a number of applications where sensors and robots could work together. In all these cases, the important factor is that the whole process is self-organizing without any external surveillance. Sensors detect events autonomously and the mobile robots take appropriate actions based on the nature of the event. The performance of the system considerably increases when continued assistance to the event location is needed. Coordination between the mobile robots is critical in achieving better network efficiency. These networks can be implemented in various applications, such as defense, locating a user, environmental monitoring, fire fighting, rescue operations, tele-monitoring, damage detection, traffic analysis, etc.

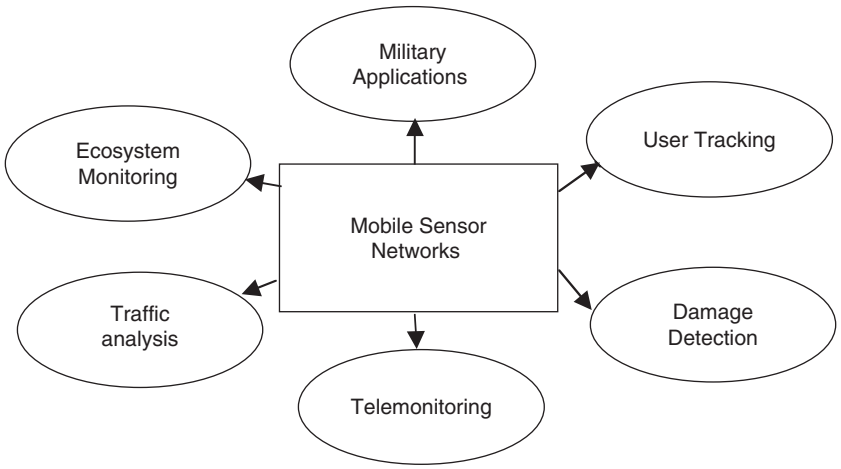


FIGURE 20.18 Applications of mobile sensor networks. (*Source:* A.K. Gupta, S. Sekhar, and D.P. Agrawal, Efficient event detection by collaborative sensors and mobile robots, in *Ohio Graduate Symp. On Computer and Information Science and Engineering*, Dayton, OH, June 2004.)

Caching on two independently powerful spheres, we propose a scheme that conjoins them. Both the entities try to complement each other and the pitfall of each field is covered by the other. This directly leads to an improved net work lifetime. The possible applications of this scheme are manifold. Because the network has a stable lifetime, it could be deployed in many areas that need continuous sensing. Because the field of robotics is ever-improving, one can expect the network to perform better with time. Unlike traditional networks, data collection can be done periodically or intermittently as per the needs of the system. Loss of data is also largely alleviated. Very efficient and automated surveillance and monitoring systems will be of significant value in the near future and that is exactly what our system achieves! The adroitness of these mobile robots forms the backbone of the scheme and their synchronization forms the communication channel of the network.

20.8 Security

Before proceeding further, it should be noted that errors in sensor measurements are inevitable. Thus, to achieve high-fidelity measurements from any sensor, there is a need to map from raw sensor readings to the correct values before the decision process is invoked. However, in addition to erroneous sensor readings, there could be malicious or compromised nodes in a network and, hence, security measures are also important. Research on security mainly focuses on the creation of a trustworthy, efficient, and easy-to-manage ad hoc scenario. Any security infrastructure essentially consists of two components. The first involves intrusion prevention and the second concerns intrusion detection. We focus on both and also look at schemes for security for group communications and secure routing. The following sections look at the various aspects of research into these various domains.

20.8.1 Security for Group Communication

Security for group communication essentially requires managing keys such that common keys are shared and the rekey overhead is minimized. To this end, a multitude of schemes have been suggested.

Group communication has a few inherent issues. First, when a node joins the group for the first time, it should receive a communication key shared by each of the other members of the group. However, providing it with the past keying material would mean that the node would be able to decipher conversation to which it is not entitled. Thus, a new set of keying material must be distributed. A leave also results in a similar problem, however with magnified impact. This problem has been solved by providing a scheme (Mukherjee et al. 2004) in which the problem of rekeying is carried out at each individual step and in a level-based manner.

In Mukherjee et al. (2004), a node is assigned a "level" as soon as it joins the network. This procedure also establishes a level key between the node and its parent. The level key is the same as the one that the parent shares with its other children. The parent node passes on its level key to any node that wishes to join it. We thus have a tree in which each parent-children group shares the same common level key. In our scheme, an entire area need not be rekeyed whenever a "transfer" occurs, thereby drastically reducing the rekeying overhead. The multicast group key update takes place by repeated decryption and encryption of the multicast key at each level. We thus achieve low communication overhead (because the update message would travel only once along each link); low latency (by the assumption that the routing layer has some cross-layer information and can do this encryption-decryption process very fast without the packet going to the application layer); effective handling of user mobility (by introducing mobility parameters, as shown later); and most importantly, a security framework that is entirely distributed in nature.

Mukherjee et al. (2004) defined a parameter called *self-mobility* for a node based on the change in the neighborhood of node in a specified time period. Based on this, they calculated a combined mobility factor. This factor enables one to predict a join or a leave operation's overhead consequences with high accuracy. A communicating entity wishing to leave or join the system would make a decision to join the multicast or group tree so as to have minimal post join or leave rekey overheads. Simulation results show very positive results as well as a drastic reduction in the keying overhead.

20.8.2 Key Exchange

A major problem in tackling security issues in ad hoc networks is the absence of any central authority. Without any central body to distribute the key material, nodes joining a network need to get their key material from the network itself. A lot of assumptions are made about the structure of the network, thus leading to a multitude of solutions.

20.8.2.1 ID-Based Key Exchange

The central idea behind an ID-based key exchange process is that a node joining the network gets its keying material according to its ID. The primary assumption here is that a central authority initializes a group of communicating nodes with the keying material and these nodes provide shares for the keying material to any node that joins the network.

Deng et al. (2004) have proposed an asymmetric keying mechanism where the node's ID is considered its public key. A major issue with most asymmetric key mechanisms is the distribution of the private keys. This problem is solved by the ID itself becoming the public key. As soon a node joins a network, it sends out a share request to all its neighbors. The neighbors respond to the request by calculating a share using a master key share and the requesting node's ID. It then sends this share to the requesting node. The requesting node now combines all the shares to form a private key. The public key is the node's ID, as previously mentioned.

20.8.2.2 Group Key Mechanisms

Next, consider a group key generation mechanism. The idea here is mainly based on the Diffie-Hellman primitive. Ding et al. (2004) have defined new trust relationships.

Ding et al (2004) have suggested a novel method to carry out distributed key exchange in such a manner that each user now introduces a separate level of trust into the system for itself. Most authentication schemes treat the problem of authentication as separate from that of communication. Their contention is that for highly secure networks, a user should be able to choose a level of trust for the party he wishes to communicate with, and the authentication process should be embedded with the key exchange process.

The basic Diffie-Hellman primitive derives itself from the fact that factorization is hard in a finite field. Thus, if two parties wish to exchange a symmetric key, they exchange numbers and raise these numbers to the power of a secret number.

Ding et al. (2004) also introduced the notion of a body of servers. These servers are responsible for authenticating and distributing key materials to an incoming member. Trust assumptions are such that no node that has not been authenticated by this set of servers would be able to communicate with this node. Thus, we establish the notion of distributed trust.

As future work we can come up with schemes to reduce the number of messages required. One way in which this can be done is to have neighbor graphs to direct a node to choose servers that are close to it and not incur too much communication overhead on the network. Also, traffic analysis might be done and key messages can be piggy-backed on other data messages. A third way is to have servers maintaining multicast groups to which messages might be sent, thereby saving the overhead of unicasting to each server separately. Second, a possible extension would be to apply the scheme to real-life scenarios involving multiple security levels for different users. Finally, we would like to make our system fool-proof by coming up with schemes to detect whether or not a server sends the shares of the secret key honestly. This can be done by either introducing intrusion detection mechanisms or by having some kind of a mechanism to see if the obtained shares satisfy a set of criteria.

We can also build robustness into the distributed keying mechanisms by building validity verifier protocols. Thus, when the key shares are generated in a distributed fashion and compromised nodes lie about the keying material, there should be a way in which the incoming nodes can detect the lies and work toward omitting the wrong values and also pinpoint the malicious node, thereby contributing to intrusion detection.

Furthermore, the symmetry in the polynomials can be used to derive symmetric keys. Thus, multivariate symmetric polynomials, partially evaluated by a central authority, can be utilized to generate pair-wise

or group keys between nodes. We have exploited this fact and are in the process of making the process distributed.

Finally, polynomial-based hierarchical keying mechanisms can be considered. The evaluation of the polynomials would be such that the higher the tiers, the more information would be provided to the nodes. Thus, the lowest tier nodes would have much less information and the nodes at the upper tiers would have a lot of information.

20.8.3 Secure Routing Schemes

Secure routing schemes concern protocols in which routing messages are neither tampered with nor dropped.

In the main CBRP routing scheme, routing is carried out in a clustered manner. A central cluster head is responsible for the routing decisions in each cluster. It is our contention that this scheme can fall victim to severe routing attacks if the cluster head is compromised.

Ojha et al. (2204) and Poosarla et al. (2004) have come up with schemes to prevent this from happening. A cluster head is not just one node but a collection of nodes. These nodes together form a COUNCIL. Any routing decision that needs to be taken must be done after an agreement with this COUNCIL. The structure of the COUNCIL is such that compromise of less than $t$members will not compromise the system. Thus, by choosing a suitable threshold value for the COUNCIL, a robust routing algorithm can be devised.

20.8.4 Intrusion Detection

Next consider intrusion detection. Here, the primary goal is to indicate if malicious activity goes around in an ad hoc network. In considering this problem, there are two issues to solve. The first concerns the issue of selecting parameters to monitor and the second concerns getting statistics for the network.

As far as the first issue is concerned, we first identified the various forms of attacks that plague an ad hoc network. These involve blackhole, wormhole, false routes, extra data packets, gracious detour, etc. Based on these attacks, we identified parameters that might indicate anomalies in the network. Deng et al. (2003a,b,c) have mainly focused on indicating the blackhole attack.

The second issue involves the actual gathering of statistics. In a practical scenario, an ad hoc node would not have enough computational power to carry out the various calculations for anomaly detection. Thus, a dimension reduction scheme is needed. Two dimension reduction schemes can be considered: (1) random projection and (2) the use of support vector machines.

20.9 Middleware Infrastructure for Sensor Networks

Middleware sits between the operating system and the application. On traditional desktop computers and portable computing devices, operating systems are well established, both in terms of functionality and systems. For sensor nodes, however, the identification and implementation of appropriate operating system primitives is still a research issue. Hence, at this early stage, it is not clear on which basis future middleware for sensor networks can typically be built (Römer et al., 2002).

The main purpose of middleware for sensor networks is to support the development, maintenance, deployment, and execution of sensing-based applications. This includes mechanisms for formulating complex, high-level sensing tasks; communicating this task to the sensor network; coordination of sensor nodes to split the task and distribute it to the individual sensor nodes; data fusion for merging the sensor readings of the individual sensor nodes into a high-level result; and reporting the result back to the task issuer.

There are already some projects to develop middleware for sensor networks. Cougar (www.cs.cornell.edu/database/cougar) adopts a database approach wherein sensor readings are treated like virtual relational database tables. The Smart Messages Project (www.rutgers.edu/sm) is based on agent-like messages containing code and data that migrate through the sensor network. NEST (www.cs.virginia.edu/nest) provides

microcells that are similar to operating system tasks with support for migration, replication, and grouping. SCADDS (Scalable Coordination Architectures for Deeply Distributed Systems; www.isi.edu/dov7/scadds) is based on *directed diffusion*, which supports robust and energy-efficient delivery and in-network aggregation of sensor events. However, most of these projects are in an early stage, focusing on developing algorithms and components that might later serve as a foundation for middleware for sensor networks.

20.10 Future Challenges of Sensor Networks

Sensor networks represent a paradigm shift in computing. Traditional computing involves computers directly interacting with human operators. However, in the near future, hundreds of computers will be embedded deep in the world around us. When we are in control of hundreds or thousands of computers each, it will be impossible for us to interact directly with each one. On the contrary, the computers themselves will interact more directly with the physical world. They will sense their environments directly, compute necessary responses, and execute them directly (CSTB Publications, 2001).

For example, the networks in current cars are highly engineered systems in which each microprocessor and the overall network are carefully designed as a whole. However, as the complexity of the network and the functionality of the networked elements grow, the ability to approach the networks as single, fully engineered, closed systems becomes strained. For example, owners might want to integrate their own devices into the car viz. integrating the address book in a PDA with the navigation system in the car. The major automobile companies plan to change the car from a self-contained network (or pair of networks) into a node in a much larger network. One approach to this is General Motors' immensely successful OnStar offering. OnStar connects the car to the manufacturer, allowing the latter to monitor emergency situations and give on-demand help to the occupants of the car.

Intel (www.intel.com/research/exploratory/digital_home.htm) is contributing to the development of digital home technologies for aging in place by linking together computers and consumer electronic (CE) devices throughout the home in a wireless network. Once the digital home infrastructure is in place, any computer or CE device could be used to deliver health and wellness applications. Older adults will be able to access these applications through whatever interfaces are most familiar to them, from phones to PCs to televisions; they will not have to learn new technology. The goal is to have a variety of interfaces distributed throughout the home, all within easy reach of the person needing assistance.

Sensors 2000 has developed an easy-to-implement wireless biotelemetry system — called the Wireless On-Patient Interface for Health Monitoring (WOPI) — that can nonintrusively measure the health parameters of humans and animals in space (see Figure 20.19). The device's sensors, which are connected to miniature transceiver modules, are implanted, ingested, or attached to the body with Band-Aids. Sensors communicate with a belt-worn device that retransmits or records the data and also sends basic commands to each sensor. The device also displays a quick status of all physiological and biological parameters (see Figure 20.20). The technology also has potential applications in athletics and emergency response activities.

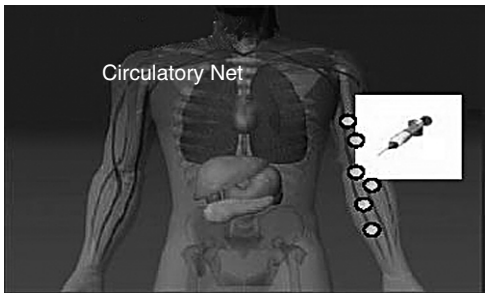


FIGURE 20.19 Biosensors. (*Source:* www-lia.deis.unibo.it/Courses/RetiLS/seminari/WSN.pdf)

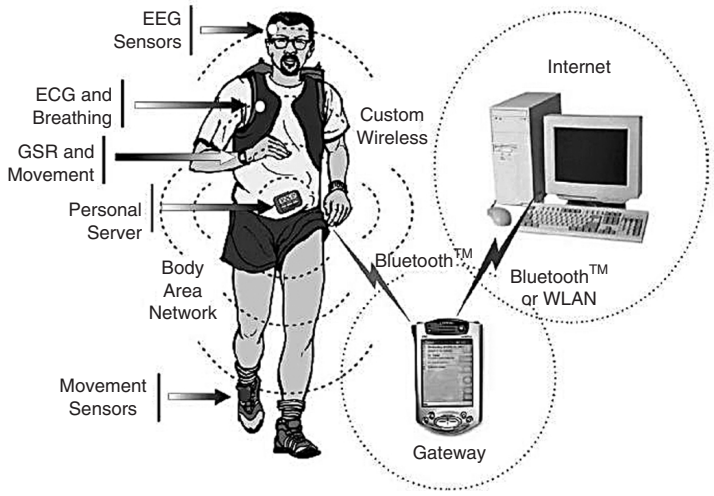


FIGURE 20.20 Wireless body area network of intelligent sensors in the telemedical environment. (*Source*: E. Jovanov, A. O'Donnell Lords, D. Raskovic, P. Cox, R. Adhami, and F. Andrasik, Stress monitoring using a distributed wireless intelligent sensor system, *IEEE Eng. in Medicine and Biology Magazine*, May/June 2003.)

References

1. D.P. Agrawal and Q. Zeng, *Introduction to Wireless and Mobile Systems,* Brooks/Cole Publishing, 436 pp., 2003.
2. R. Avnur and J.M. Hellerstein Eddies: continuously adaptive query processing, in *Proc. 2000 ACM SIGMOD Int. Conference on Management of Data,* May 2000, pp. 261–272.
3. B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, Models and issues in data stream systems, in *Proceedings of the 2002 ACM Symposium on Principles of Database Systems,* June 2002, pp. 1–16.
4. S. Bhatnagar, B. Deb, and B. Nath, Service Differentiation in Sensor Networks, in *Proc. Fourth Int. Symp. Wireless Personal Multimedia Communications,* 2001.
5. B.J. Bonfils and P. Bonnet, Adaptive and Decentralized Operator Placement for In-Network Query Processing Information Processing in Sensor Networks, *Second Int. Workshop, IPSN 2003,* Palo Alto, CA, April 22–23, 2003.
6. P. Bonnet, J.E. Gehrke, and P. Seshadri, Towards Sensor Database Systems, in *Proc. Second Int. Conf. on Mobile Data Management,* Hong Kong, January 2001.
7. E. Brewer, R. Katz, and E. Amir, A network architecture for heterogeneous mobile computing, *IEEE Personal Communications Magazine,* October 1998.
8. J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in *Proc. 4th Annual ACM/IEEE Int. Conf. Mobile Computing (MOBICOM),* ACM, October 1998.
9. Campbell Scientific, Inc., Measurement and Control Systems. Web page. www.campbellsci.com.
10. D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S.B. Zdonik, Monitoring streams — a new class of data management applications, in *Proc. 28th VLDB,* 2002.
11. A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, Habitat monitoring: application driver for wireless communications technology, *2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean,* Costa Rica, April 2001.
12. S. Chandrasekaran, O. Cooper, A. Deshpande, M.J. Franklin, J.M. Hellerstein, W. Hong, S. Krishnamurthy, S.R. Madden, V. Raman, F. Reiss, and M.A. Shah, TelegraphCQ: continuous dataflow

processing for an uncertain world, *1st Biennial Conf. Innovative Data Systems Research (CIDR 2003),* January 2003.
13. J. Chang and L. Tassiulas, Energy conserving routing in wireless ad-hoc networks, *Proc. IEEE INFOCOM,* pp. 22–31, 2000a.
14. J. Chang and L. Tassiulas, Maximum lifetime routing in wireless sensor networks, in *Proc. Advanced Telecommunications and Information Distribution Research Program,* 2000b.
15. J. Chen, P. Druschel, and D. Subramanian, An efficient multipath forwarding method, *Proc. IEEE INFOCOM,* 1998.pp. 1418–1425.
16. S. Chen and K. Nahrstedt, Distributed quality-of-service routing in ad-hoc networks, *IEEE Journal on Special Areas in Communications,* 17(8), August 1999. Au: page number available?
17. M. Chiang, D. O'Neill, D. Julian, and S. Boyd, Resource allocation for QoS provisioning in ad hoc wireless networks, *Proc. IEEE GLOBECOM,* San Antonio, November 2001, pp. 2911–2915.
18. Chu, H. Haussecker, and F. Zhao, Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks, in *Int. J. High Performance Computing Applications*, 2002. Au: Author initials?
19. I. Cidon, R. Rom, and Y. Shavitt, Analysis of multi-path routing, in *IEEE/ACM Transactions on Networking,* 7(6), 885–896, December 1999. Au: Volume and page numbers?
20. CITRIS, Center for Information Technology Research in the Interest of Society. Web Page. www.citris-uc.org.
21. Compaq iPAQ, Web Page. www.compaq.com/products/iPAQ.
22. Cougar Project; Web page: www.cs.cornell.edu/database/cougar.
23. CSTB Publications, Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers, CSTB Publications, 236 pages, ISBN: 0-309-07568-8, 2001.
24. Datar, A. Gionis, P. Indyk, and R. Motwani, Maintaining stream statistics over sliding windows, Proc. Thirteenth Annu. ACM-SIAM Symp. on Discrete Algorithms, p. 635–644, January 06–08, 2002, San Francisco, CA. Au: Author initials?
25. H. Deng, A. Mukherjee, and D.P. Agrawal, Threshold and identity-based key management and authentication for wireless ad hoc networks, *IEEE Int. Conf. Information Technology (ITCC'04),* April 5–7, 2004.
26. Deng, Q.-A. Zeng, and D.P. Agrawal, SVM-based intrusion detection system for wireless ad hoc networks, *Proc. IEEE Vehicular Technology Conf. (VTC'03),* Orlando, October 6–9, 2003a.
27. H. Deng, Q.-A. Zeng and D.P. Agrawal, Projecting High-Dimensional Data for Network Intrusion Detection, *Proc. Joint Conf. on Information Sciences (JCIS'03),* September 26–30, 2003b, pp. 373–376.
28. H. Deng, Q.-A. Zeng, and D.P. Agrawal, An unsupervised network anomaly detection system using random projection technique, *Proc. 2003 Int. Workshop on Cryptology and Network Security (CANS'03),* Miami, FL, September 24–26, 2003c, pp. 593–598.
29. J. Ding, A. Mukherjee, and D. Agrawal, Distributed authentication and key generation with multiple user trust level (fast abstract), *The Int. Conf. on Dependable Systems and Networks, (DSN 2004).* June 28–July 1, 2004.
30. L. Doherty, K.S.J. Pister, and L.E. Ghaoui, Convex position estimation in wireless sensor networks, in *Proc. IEEE INFOCOM,* Alaska, April 2001, pp. 1655–1663.
31. Ember Corporation, Process Temperature Integrated Sensing and Control, www.ember.com/products/solutions/industrialauto.html.
32. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, Next century challenges: scalable coordination in wireless networks, in *Proc. 5th Annu. ACM/IEEE Int. Conf. on Mobile Computing and Networking (MOBICOM),* 1999, pp. 263–270.
33. D. Ganesan, D. Estrin, and J. Heidemann, DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks?, *Proc. ACM Workshop on Hot Topics in Networks,* 2002a.
34. D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, Highly *Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks,* in *Mobile Computing and Communications Review (MC2R),* Vol. 1, No. 2, 2002b.

Au: Volume? and page number?

35. J. Gehrke and S. Madden, Query Processing In Sensor Networks, in *Pervasive Computing,* 2004.
36. A.K. Gupta, S. Sekhar, and D.P. Agrawal, Efficient event detection by collaborative sensors and mobile robots, in *Ohio Graduate Symposium on Computer and Information Science and Engineering,* Dayton, OH, June 2004.
37. J. Hayes, M. McJunkin and J. Kosecka, Communication Enhanced Navigation Strategies for Teams of Mobile Agents, *IEEE Robotics and Automation Society,* Las Vegas, October 2003.
38. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-Efficient Communication Protocols for Wireless Microsensor Networks, in *Proc. Hawaaian Int. Conf. on Systems Science,* January 2000.
39. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, μAMPS ns Code Extensions, www-mtl.mit.edu/research/icsystems/uamps/leach.
40. J.M. Hellerstein, P.J. Haas, and H. Wang, Online aggregation, in *Proc. ACM SIGMOD,* Tucson, AZ, May 1997, pp. 171–182.
41. J.M. Hellerstein, W. Hong, S. Madden, and K. Stanek, Beyond average: towards sophisticated sensing with queries, in *Proc. First Workshop on Information Processing in Sensor Networks (IPSN),* March 2003.
42. J. HighTower and G. Borreillo, Location systems for ubiquitous computing, *IEEE Computer,* 34, 57–66, August 2001.
43. J. Hill, R. Szewczyk, A.Woo, S. Hollar, D. Culler, and K. Pister, System Architecture Directions for Network Sensors, in *Proc. 9th Int. Conf. on Architectural Support for Programming Languages and Operating Systems,* November 2000, pp. 93–104.
44. T. Imielinski and B. Nath, Wireless graffiti — data, data everywhere, in *Int. Conf. on Very Large Data Bases (VLDB),* 2002.
45. C. Intanagonwiwat, R. Govindan, and D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, in *Proc. 6th Annu. ACM/IEEE Int. Conf. on Mobile Computing and Networking (MOBICOM),* August 2000, pp. 56–67.
46. C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, Impact of Network Density on Data Aggregation in Wireless Sensor Networks, Technical Report 01-750, University of Southern California, November 2001.
47. Intel; Intel Exploratory Research; www.intel.com/research/exploratory/digital_home.htm.
48. Intel Research Oregon. Heterogeneous Sensor Networks, Technical report, Intel Corporation, 2003. Web Page. www.intel.com/research/exploratory/heterogeneous.htm.
49. N. Jain, Energy Efficient Information Retrieval in Wireless Sensor Networks, Ph.D. thesis, 2004.
50. N. Jain, D.K. Madathil, and D.P. Agrawal, Energy aware multi-path routing for uniform resource utilization in sensor networks, in *Proc. IPSN'03 Int. Workshop on Information Processing in Sensor Networks,* Palo Alto, CA, April 2003a.
51. N. Jain, D.K. Madathil, and D.P. Agrawal, Exploiting multi-path routing to achieve service differentiation in sensor networks, *Proc. 11th IEEE Int. Conf. on Networks (ICON 2003),* Sydney, Australia, October 2003b.
52. N. Jain and D.P. Agrawal, Current trends in Wireless Sensor Networks, Technical Report, CDMC, University of Cincinnati.
53. E. Jovanov, A. O'Donnell Lords, D. Raskovic, P. Cox, R. Adhami, and F. Andrasik, Stress monitoring using a distributed wireless intelligent sensor system, *IEEE Engineering in Medicine and Biology Magazine,* May/June 2003.
54. K. Kar, M. Kodialam, and T.V. Lakshman, Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications, *IEEE J. Selected Areas in Commun.*, Vol. 18, No. 12, December 2000.
55. B. Karp and H.T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in *Proc. ACM/IEEE MobiCom,* August 2000.
56. R. Kumar, V. Tsisatsis, and M. Srivatsava, Computation hierarchy for in-network processing, in *Proc. WSNA'03,* 2003.

57. S. Lee and A.T. Campbell, INSIGNIA: in-band signaling support for QOS in mobile ad hoc networks, in *Proc. 5th Int. Workshop on Mobile Multimedia Communications(MoMuC'98),* Berlin, Germany, October 1998.
58. S. Lee and M. Gerla, Split multipath routing with maximally disjoint paths in ad hoc networks, *Proc. IEEE ICC, 2001*. pp. 3201–320.
59. S. Madden, The Design and Evaluation of a Query Processing Architecture for Sensor Networks, Ph.D. thesis. University of California, Berkeley, Fall 2003.
60. S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, The design of an acquisitional query processor for sensor networks, *ACM SIGMOD Conf.,* San Diego, CA, June 2003.
61. S. Madden, M. Shah, J.M. Hellerstein, and V. Raman, Continuously adaptive continuous queries over streams, in *ACM SIGMOD Int. Conf. on Management of Data,* Madison, WI, 2002, pp. 49–60.
62. I. Mahadevan and K.M. Sivalingam, Architecture and Experimental Framework for Supporting QoS in Wireless Networks Using Differentiated Services, *MONET* 6(4), 2001, pp. 385–395.
63. A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, Wireless sensor networks for habitat monitoring, in *ACM Workshop on Sensor Networks and Applications,* 2002.
64. A. Manjeshwar, Energy Efficient Routing Protocols with Comprehensive Information Retrieval for Wireless Sensor Networks, M.S. thesis, 2001.
65. A. Manjeshwar and D.P. Agrawal, TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, in *Proc. 15th Int. Parallel and Distributed Processing Symp. (IPDPS'01) Workshops,* 2001.
66. A. Manjeshwar and D.P. Agrawal, APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in *Proc. Int. Parallel and Distributed Processing Symp. (IPDPS'02) Workshops,* 2002.
67. A. Manjeshwar, Q. Zeng, and D.P. Agrawal, An analytical model for information retrieval in wireless sensor networks using enhanced APTEEN protocol, in *IEEE Trans. Parallel and Distributed Systems,* 13(12), 1290–1302, December 2002.
68. N.F. Maxemchuk, Dispersity routing in high-speed networks, *Computer Networks and ISDN System 25,* 1993, 645–661.
69. MICA Sensor Mote, Web page: www.xbow.com/Products/Wireless Sensor Networks.htm.
70. MicroStrain Microminiature Sensors. Web page: www.microstrain.com.
71. H. Mistry, P. Roy, S. Sudarshan, and K. Ramamritham, Materialized view selection and maintenance using multi-query optimization, in *ACM SIGMOD,* 2001.
72. G.E. Moore, Cramming More Components onto Integrated Circuits, Electronics, April 1965, pp. 114–117.
73. R. Motwani, J. Window, A. Arasu, B. Babcock, S. Babu, M. Data, C. Olston, J. Rosenstein, and R. Varma, Query processing, approximation and resource management in a data stream management system, in *First Annu. Conf. on Innovative Database Research (CIDR),* 2003.
74. A. Mukherjee, M. Gupta, H. Deng, and D.P. Agrawal, Level-Based Key Establishment For Multicast Communication In Mobile Ad Hoc Networks, submitted to *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC'04),* September 5–8, 2004.
75. A. Nasipuri and S. Das, On-demand multipath routing for mobile ad hoc networks, *Proc. 8th Annu. IEEE Int. Conf. on Computer Communications and Networks (ICCCN),* October 1999, pp. 64–70.
76. J. Nelson and D. Estrin. An Address-Free Architecture for Dynamic Sensor Networks, Technical Report 00-724, Computer Science Department, University of Southern California, January 2000.
77. NEST, A Network Virtual Machine for Real-Time Coordination Services. www.cs.virginia.edu/nest.
78. The Network Simulator - ns-2, www.isi.edu/nsnam/ns.
79. Ojha, H. Deng, S. Sanyal, and D.P. Agrawal, Forming COUNCIL based clusters in securing wireless ad hoc networks, *Proc. 2nd Int. Conf. on Computers and Devices for Communication (CODEC'04),* January 1–3, 2004.
80. V.D. Park and M.S. Corson, A highly distributed routing algorithm for mobile wireless networks, in *Proc. IEEE INFOCOM,* 1997, pp. 1405–1413.

81. M.R. Pearlman, Z.J. Hass, P. Sholander, and S.S. Tabrizi, On the impact of alternate path routing for load balancing in mobile ad hoc networks, in *Proc. IEEE/ACM MobiHoc,* 2000.
82. J. Pinto, Intelligent Robots will be everywhere, *Robotic Trends,* December 2003.
83. V. Raman, B. Raman, and J. M. Hellerstein, Online dynamic reordering, *The VLDB Journal,* 9(3), 2002.
Au: Author initial?
84. Poosarla, H. Deng, A. Ojha, and D.P. Agrawal, A cluster based secure routing scheme for wireless ad hoc networks, *The 23rd IEEE Int. Performance, Computing, and Communications Conf. (IPCCC'04),* April 14–17, 2004.
85. K. Römer, O. Kasten, and F. Mattern, Middleware challenges for wireless sensor networks, in *ACM SIGMOBILE Mobile Computing and Communication Review (MC2R),* Vol. 6, No. 4, October 2002.
86. E.M. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks, in *IEEE Personal Communications Magazine,* April 1999, pp. 46–55.
87. SCADDS, Scalable Coordination Architectures for Deeply Distributed Systems. www.isi.edu/div7/scadds.
88. P.S. Schenker, T.L Huntsberger, and P. Pirjanian, Robotic autonomy for space: ooperative and reconfigurable mobile surface systems, 6*th Int. Symp. on Artificial Intelligence,* Montreal, Canada, June 2001.
89. L. Schwiebert, S.D.S. Gupta, and J. Weinmann, Research challenges in wireless networks of biomedical sensors, *MobiCom* 2001.
90. C. Schurgers and M.B. Srivastava, Energy efficient routing inwireless sensor networks, *MILCOM'01,* October 2001.
91. S.D. Servetto and G. Barrenechea, Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks, *in Proc. 1st ACM Int. Workshop on Wireless Sensor Networks and Applications,* September 2002, pp. 12–21.
92. R.C. Shah and J. Rabaey, Energy aware routing for low energy ad hoc sensor networks, *IEEE Wireless Communications and Networking Conference (WCNC),* March 2002.
93. Smart Messages Project. www.rutgers.edu/sm.
94. K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, Protocols for self-organization of a wireless sensor network, *IEEE Personal Communications,* 7(5), 16–27, October 2000.
95. L. Subramanian and R. Katz, An architecture for building self-configurable systems, in *1st Annu. Workshop on Mobile and Ad Hoc Networking and Comp.,* 2000, pp. 63–78.
96. H. Suzuki and F.A. Tobagi, Fast bandwidth reservation scheme with multi link multi path routing in ATM networks, in *Proc. IEEE INFOCOM,* 1992.
97. TinyOS: Operating System for Sensor Networks Web Page. http://tinyos.millennium.berkeley.edu.
98. R. T. Vaughan, K. Støy, G. S. Sukhatme, and M. J. Matarı́c, Blazing a trail: insect-inspired resource transportation by a robot team, *DistributedAutonomous Robotic Systems,* DARS 2000, Tennessee, October 2000, pp. 111–120.
99. S. Vutukury, Multipath Routing Mechanisms for Traffic Engineering and Quality of Service in the Internet, Ph.D thesis, March 2001.
100. H. Wang, D. Estrin, and L. Girod, Preprocessing in a Tiered Sensor Network for Habitat Monitoring, in *EURASIP JASP* Special Issue on Sensor Networks, v2003(4), 392–401, March 15, 2003.
101. Xu, J. Heidemann, and D. Estrin, Geography-informed energy conservation for ad hoc routing, in *Proc. ACM/IEEE Int. Conf. on Mobile Computing and Networking,* Rome, Italy, USC/Information Sciences Institute, July, 2001, pp. 70–84.
102. Y. Yu, R. Govindan, and D. Estrin, Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks, UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, May 2001.