Abstract

We present a suite of algorithms for self-organization of wireless sensor networks in which there is a scalably large number of mainly static nodes with highly constrained energy resources. The protocols further support slow mobility by a subset of the nodes, energy-efficient routing, and formation of ad hoc subnetworks for carrying out cooperative signal processing functions among a set of the nodes.

Protocols for Self-Organization of a Wireless Sensor Network

KATAYOUN SOHRABI, JAY GAO, VISHAL AILAWADHI, AND GREGORY J. POTTIE, UCLA

In this article we describe an architecture for self-organizing wireless sensor networks [1]. These are wireless ad hoc networks that connect deeply embedded sensors, actuators, and processors. This combination of wireless and data networking will result in a new form of computational paradigm which is more communication-centric than any computer network seen before. Wireless sensor networks are part of a growing collection of information technology constructs which are moving away from the traditional desktop wired network architecture toward a more ubiquitous and universal mode of information connectivity [2].

A wireless sensor network of the type investigated here refers to a group of sensors, or nodes, linked by a wireless medium to perform distributed sensing tasks. Connections between nodes may be formed using such media as infrared devices or radios. Wireless sensor networks will be used for such tasks as surveillance, widespread environmental sampling, security, and health monitoring. They can be used in virtually any environment, even those where wired connections are not possible, the terrain inhospitable, or physical placement difficult. They may also be used as enabling infrastructure for new sensing/computational paradigms such as those described in [3].

Design challenges encountered in building wireless sensor networks may be categorized into three classes:

- **Hardware**: This category includes the entire range of design activities related to the hardware platforms that make up sensor networks. MEMS sensor technology is an important aspect of this category. Digital circuit design and system integration for low power consumption are also in this category [4], as well as design of a low-power sophisticated radio frequency (RF) front-end and associated control circuitry. For example, we may consider the sequence of generations of wireless integrated network sensors (WINS). A single WINS node combines microsensor technology, low power signal processing, low power computation, low power, and low-cost wireless networking capability in a compact system. Figure 1 gives a description of the WINS node architecture. Piconet [5] is another example of compact node architecture.
- **Wireless networking**: Given the hardware limitations and physical environment in which the nodes must operate, along with application-level requirements, the algorithms and protocols must be designed to provide a robust and energy-efficient communications mechanism. Design of physical-layer methods such as modulation, and source and channel coding also fall in this category. Channel access methods must be devised, and routing issues and mobility management solved. This article focuses on a number of design aspects in this category.
- **Applications**: At the application layer, processes aim to create effective new capabilities for efficient extraction, manipulation, transport, and representation of information derived from sensor data. In most applications, sensor networks have various functional components: detection and data collection, signal processing, data fusion, and notification.

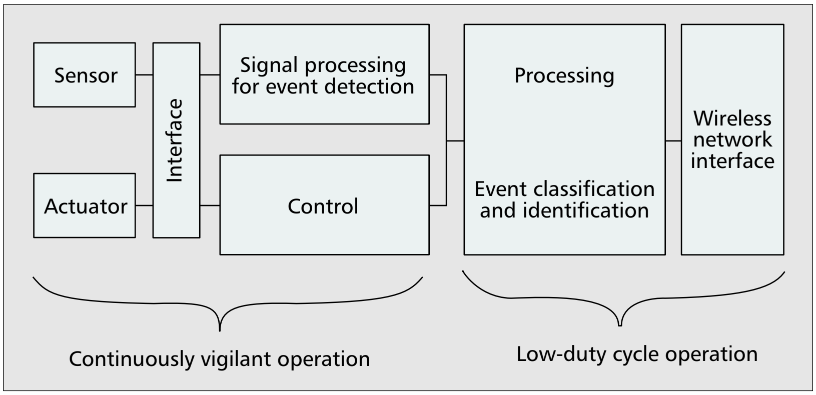
By integrating sensing, signal processing, and communications functions, a sensor network provides a natural platform for hierarchical information processing [6]. It allows information to be processed on different levels of abstraction, ranging from detailed microscopic examination of specific targets to a macroscopic view of the aggregate behavior of targets. Any events in the environment can be processed on three levels: node, local neighborhood, and global. On the node level, data collection and processing occurs in each individual node, requiring no communications except for transmission of the results to some distant information sink. On the local and global levels, internode communication is required for gathering raw or preprocessed data from multiple nodes to a single location for cooperative signal processing such as data fusion or beamforming.

A General Operational Scenario

A sensor network must be able to operate under very dynamic conditions. Specifically, our protocols must be able to enable network operation during *startup*, *steady state*, and *failure*. Note that these terms are used very loosely here. The necessity of operation under these conditions comes about because the sensor network must, in most cases, operate unattended.

Once the nodes have booted up and a network is formed, most of the nodes will be able to sustain a steady state of operation; that is, their energy reservoirs are nearly full, and they can support all the sensing, signal processing, and communica-

This research is supported by DARPA contract no. F04701-97-C-0010, and was presented in part at the 37th Allerton Conference on Communication, Computing and Control, September 1999.


Sensor
Actuator
Interface
Signal processing for event detection
Control
Processing
Event classification and identification
Wireless network interface
Continuously vigilant operation
Low-duty cycle operation

■ **Figure 1.** *The architcture of a sensor node.*

tions tasks required. In this mode, the bulk of the nodes will be formed into a multihop network. The nodes begin to establish routes by which information is passed to one or more sink nodes. A sink node may be a long-range radio, capable of connecting the sensor network to existing long-haul communications infrastructure. The sink may also be a mobile node acting as an information sink, or any other entity required to extract information from the sensor network.

There are instances when collections of nodes need to cooperate together in detection of signals or events, as described in [1]. When a cooperative function is required to extract information about a specific target, a local network is built to facilitate the necessary signaling and data transfer tasks. Typically, cooperative functions involve a small set of nodes near the target location and operate for relatively short time spans. They are required to adapt quickly and efficiently to the appearance of the target and the nature of the signal processing techniques required.

Although the multihop network can operate in both *sensor-to-sink* or *sink-to-sensor* (broadcast or multicast) modes, the bulk of traffic will belong to the former. This will put significant strain on the energy resources of the nodes near the sink, making that neighborhood more susceptible to energy depletion and failure. Nodes may fail due to other reasons such as mechanical failure.

When many nodes have failed, the medium access control (MAC) and routing protocols must accommodate formation of new links and routes to the sink nodes. This may require actively adjusting transmit powers and signaling rates on the existing links to reduce energy consumption, or rerouting packets through regions of the network where nodes have more energy left.

Wireless Sensor Networks Are a New Family of Networks

To illustrate the impact of the physical limits of sensor networks on the design of our wireless networking algorithms we briefly discuss related wireless network models; namely, mobile ad hoc networks, cellular networks, and a number of short range wireless local area networks.

A mobile ad hoc network (MANET) is a peer-to-peer network which usually comprises tens to hundreds of communicating nodes that are able to cover ranges of up to hundreds of meters. Each node is envisioned as a personal information appliance such as a personal digital assistant (PDA) outfitted with a fairly sophisticated radio transceiver. The nodes are fully mobile. A MANET aims to form and maintain a connected multihop network capable of transporting multimedia traffic between nodes. In order to provide quality of service (QoS) in the face of mobility, a MANET must do the following:

- Organize the nodes in such a way that they are able to access the shared communications medium efficiently. This is called *forming an infrastructure* in some cases, and includes the function of providing a means of channel access for the nodes as well.
- Perform routing in the network
- Maintain the network organization and routing in the face of mobility

In a MANET the three-pronged tasks of organization, routing, and mobility management (ORM) are done to optimize for QoS. That is, the network is designed to provide good throughput/delay characteristics in the face of high node mobility. Although the nodes are portable battery-powered devices, energy consumption in this system is of secondary importance, since each device is always attached to a person, and presumably the depleted battery will be replaced when needed (the same way batteries are changed on laptops).

A cellular network is a vast network consisting of both stationary and mobile nodes. The stationary nodes, or base stations, are connected in a subnetwork with a wired backbone, forming a fixed infrastructure. Mobile nodes greatly outnumber stationary nodes (tens to hundreds of mobiles per base station), which are usually situated quite sparsely. The base stations are usually placed to cover a large region with little overlap. The issue of organization is only encountered in terms of cell-to-cell handoffs as the mobile navigates the region. Each mobile node will be only one hop away from any base station. The primary goal here is to provide high QoS, along with high bandwidth efficiency. The base stations themselves effectively have an unlimited power supply, while the mobiles are battery-operated.

Bluetooth [7] is a short-range wireless networking system intended to replace the cable between electronic consumer devices and provide RF connection between them. The Bluetooth topology is a star network where a *master* node is able to have up to seven *slave* nodes attached to it to form a *piconet*. Each *piconet* uses a centrally assigned time-division multiple access (TDMA) schedule and frequency-hopping pattern. The raw signaling rate in this system is 1 Mb/s. All nodes are synchronized to the *master*. There are mechanisms in place for multiple *piconets* to interconnect and form a multihop topology. Typical transmission power is about 1 mW. It is expected to achieve a 10 m range.

Another short-range commercial system under development is HomeRF [8]. The goals of this system are very similar to those of Bluetooth; however, the networking model is based on the IEEE 802.11 standard. The system is able to handle single-hop ad hoc networks. The radio is a frequency-hopping module. Channel access is possible under TDMA and carrier sense multiple access (CSMA) modes. Raw data rates of up to 2 Mb/s are possible. Transmission power levels are at 100 mW. Typical ranges are distances encountered in the house and yard.

In contrast to all of these networks, our sensor network potentially comprises hundreds to thousands of nodes. These nodes are generally stationary after deployment, with the exception of a very small number of mobile sensor nodes. The traffic will likely have statistical properties unlike the multimedia data streams of conventional wireless networks. Although exact sensor data traffic properties are not known yet, it is clear that, due to the nature of the observed phenomena, the required bandwidth for sensor data will be low, on the order of 1–100 kb/s [1].

The main goal in conventional wireless networks is providing high QoS (i.e., high throughput/low delay) and high bandwidth efficiency when mobility exists. For a sensor network, in contrast, we are interested in prolonging the lifetime of the network. To this end we must conserve energy, and are willing to give up performance in other aspects of operation such as QoS and bandwidth utilization. Each node depends on small low-capacity batteries as energy sources, and cannot expect replacement when operating in hostile or remote regions.

For networks with a fixed infrastructure, loss of connectivity is a statistically rare event and independent of energy usage. On the other hand, in mobile networks topological changes are mostly attributed to the mobility of the nodes, not the energy depletions caused by the execution of various networking protocols. Therefore, in order to raise system performance, mobility management and failure recovery assumes more importance than energy conservation in protocol design. For ad hoc sensor networks, however, energy depletion is the primary factor in connectivity degradation and length of operational lifetime. Therefore, overall performance becomes highly dependent on the energy efficiency of the algorithm.

Energy-Conserving Techniques in Sensor Networks

Energy consumption occurs in three domains: sensing, data processing, and communications. In a wireless sensor network, communications is the major consumer of energy. To better grasp this idea, let us compare energy costs of data transmission via radio and data processing. Taking the example described in [1], for ground-to-ground transmission, it costs 3 J of energy to transmit 1 kb of data a distance of 100 m. On the other hand, a general-purpose processor with the modest specification of 100 million instructions per second (MIPS)/W processing capability executes 300 million instructions for the same amount of energy. Fortunately, it is possible to make trade-offs between data processing and wireless communications. The sensor nodes will do more local processing, as opposed to exchanging raw data over the air. In the same vein, the protocols responsible for ORM must reduce their messaging overhead as much as possible. This leads to the need for highly localized and distributed algorithms for data processing and networking.

Our Protocols

In this section our algorithms, which will perform ORM for sensor networks, are described. Specifically we will describe the Self-Organizing Medium Access Control for Sensor Networks (SMACS) for network startup and link-layer organization. Next, the Eavesdrop-And-Register (EAR) algorithm will be presented. This algorithm enables seamless interconnection of mobile nodes in the field of stationary wireless nodes, and represents the mobility management aspect of the protocol. Finally, we present a Sequential Assignment Routing (SAR) algorithm, which facilitates multihop routing, and the Single-Winner Election (SWE) and Multi-Winner Election (MWE) algorithms, which handle the necessary signaling and data transfer tasks in local cooperative information processing. For in-depth detail about the internal mechanisms of SMACS, SAR, SWE, and MWE, see [9, 10].

Link-Layer Issues — The two major services the link layer provides to higher layers are formation of a link-layer topology (or infrastructure) and regulation of channel access among the nodes. In most existing or proposed ad hoc networks, channel access is done by two different methods: contention or explicit organization in time/frequency/code domains. The various flavors of MACA and MACAW reported widely in the literature are examples of the former. The MAC-layer design for 802.11 is an example.

The second class of channel access schemes, which we term *organized* channel access, attempts to determine network radio connectivity first (i.e., discover the radio neighbors of each node) and then assign collision-free channels to links. The task of assignment of channels (i.e., TDMA slots, frequency bands, or spread spectrum codes) to links between radio neighbors such that they do not collide is a hard problem. To ease the assignment problem a hierarchical structure is formed in the network to localize groups of nodes and make the task of channel assignment more manageable. The problem in this approach is how to determine the cluster memberships and cluster heads such that the entire network is covered while the nodes move. Some examples of solutions are given in [11–13].

The contention-based channel access schemes are clearly not suitable for sensor networks, due to their requirement for radio transceivers to monitor the channel at all times. This is a particularly expensive proposition for the low radio ranges of interest for sensor networks, where transmission and reception have almost the same energy cost. We would like to turn off the radios when no information is to be sent or received.

The organized methods of channel access require nodes in the network to be synchronized with each other at some level (usually at the slot boundary epochs for TDMA systems). In organized schemes, usually a period is set aside for neighbor discovery. If a centralized channel assignment algorithm is to be used, all the connectivity information along with any bandwidth requirements for specific links are passed to a single node in the network for calculation of a schedule. There are distributed assignment methods in place where nodes exchange connectivity data only with some local neighborhood. This network-wide synchronization is again expensive for sensor networks, because it requires extensive message passing over the air to synchronize all the nodes.

Description of the Stationary MAC and Startup Procedure

In our system we assume that nodes are able to turn their radios on and off. They are also able to tune the carrier frequency to different bands. It is assumed that the number of available bands is relatively large.[1] In our protocol a channel is defined as a pair of time intervals, similar to slots in a TDMA schedule. We assume that nodes are deployed by hand or remotely such that they are covering some area randomly. After deployment, each node wakes up at some random time according to some distribution.

SMACS is an infrastructure-building protocol that forms a flat topology (as opposed to a cluster hierarchy) for sensor networks. SMACS is a distributed protocol which enables a collection of nodes to discover their neighbors and establish transmission/reception schedules for communicating with them without the need for any local or global master nodes.

In order to achieve this ease of formation, we have combined the neighbor discovery and channel assignment phases in the SMACS protocol. Unlike methods such as the Linked Clustering Algorithm (LCA) [12], in which a first pass is per-

[1] *This is not an unreasonable assumption. If we assume that radios operate in the 902–928 Industrial, Scientific, and Medical (ISM) band, and that the data rate on each hop is no more than 10 kb/s, we may have something on the order of 2600 distinct frequency bands available from which to choose.*

formed on the entire network to discover neighbors and then another pass done to assign channels, or TDMA slots, to links between neighboring nodes, in SMACS we assign a channel to a link immediately after the link's existence is discovered. This way links begin to form concurrently throughout the network. By the time all nodes hear all their neighbors, they will have formed a connected network. In a connected network, there exists at least one multihop path between any two distinct nodes.

Since only partial information about radio connectivity in the vicinity of a node is used to assign time intervals to links, there is a potential for time collisions with slots assigned to adjacent links whose existence is not known at the time of channel assignment. To reduce the likelihood of collisions, we require each link to operate on a different frequency. This frequency band is chosen at random from a large pool of possible choices when the links are formed.

This idea is described in Fig. 2b. Nodes A and D wake up at times T_a and T_d. After they find each other, they agree to transmit and receive during a pair of fixed time slots. This transmission/reception pattern will be repeated periodically every T_{frame}. Nodes B and C wake up later, at times T_b and T_c, respectively. After they find each other they will assign another pair of slots for transmission and reception. Note that if all the nodes operate on the same frequency band, there is the possibility that some transmissions will collide in the given schedule. For example, a transmission from D to A will collide in time with a transmission from B to C. On the other hand, if different frequency bands are assigned to different links (e.g., f_x to AD link and f_y to BC link), the time schedule of Fig. 2b will work without collisions.[2] When there are many frequencies from which to choose, and frequencies are chosen uniformly at random, the likelihood that the same frequency is chosen by two links within earshot of each other is small.

T_{frame} as described above is fixed for all nodes, and is a parameter of the MAC. T_{frame} is the length of the superframe for our MAC. As new neighbors are found and new links formed, the superframe of each node will start to be filled. From Fig. 2b we see that T_{frame} epochs for nodes A and B, for example, do not coincide. Now if we call each transmission or reception period a slot, we see from the same figure that the protocol will result in slot assignments which do not need to be aligned throughout the entire network. Again, the reason this nonsynchronous assignment is possible is assignment of different frequencies to links. The ability to assign nonsynchronous slots in the network is the key issue that enables the nodes to form links on the fly. We call this concept nonsynchronous scheduled communication (NSC). This spontaneity enables a quick method of scheduling links throughout the network.

After a link is established, a node knows when to turn on its transceiver ahead of time to communicate with another node. It will turn off when no communication is scheduled. This scheduled mode of communication enables energy savings for the node. Since link assignment is accomplished quickly, without requiring accumulation of global connectivity information or even connectivity information that reaches farther than one hop away, the overall effect is significant energy savings.

We now discuss the method by which nodes find each other, and the mechanism by which time slots and operating frequencies are determined. A brief description of this mechanism was given in [14].

To illustrate this mechanism we will follow the actions of a set of nodes, B, C, and G, as shown in Fig. 2c. These nodes are engaged in the process of finding neighbors. They wake up at random times. Upon waking up, each node will listen to the channel on a fixed frequency band for some random time duration. A node will decide to transmit an invitation by the end of this initial listening time if it has not heard any invitations from other nodes. This is what happens to node C, which will broadcast an invitation or TYPE1 message. Nodes B and G hear this TYPE1 message. Each broadcasts a response, or TYPE2, message addressed to node C during the interval following reception of TYPE1 at a random time. If the TYPE2 messages do not collide, node C will hear both. Node C must choose only one respondent. It will choose node B because its response arrived first. Other selection criteria for choosing a respondent may also be used, such as choosing a node with higher received signal levels or more attached neighbors. Node C will send a TYPE3 message immediately after the end of the interval following the TYPE1 message to notify all respondents which was chosen. Node G, which was not chosen, will turn off its transceiver for some time and then start the search procedure.

If node C is already attached, it will transmit its schedule information, along with the time its next superframe will start, in the body of TYPE3. Node B will read this information, compare the two schedules and time offsets, and arrive at a set of two free time intervals as the slots assigned to the link between C and B. Node B will then send the location of these time slots along with the randomly selected frequency band of operation to node C in the body of a TYPE4 message. At this point the two nodes have a pending link between them. Once a pair of short test messages is successfully exchanged between the two nodes using the newly assigned slots, the link is added to the nodes' schedules permanently.

We define a *subnet* to be a subset of nodes that form a connected graph and have coinciding superframe epochs. There are two or more nodes in each subnet. For example, in Fig. 2b nodes, A and D form a subnet and B and C form another. As time goes on, these subnets grow in size by attaching new nodes. They will eventually become attached to other subnets, until finally almost all the nodes in the network are connected together.[3]

The case when two nodes find each other and attempt to form a link, while they are already members of different *subnets* is the most challenging scenario in our startup procedure. As long as the super frame of both nodes has enough overlap in unassigned regions to allocate a pair of slots for the new link, there is no need for the two nodes to re-organize their respective schedules in order to make room for the new link. If there is no room left, the two nodes will simply give up and search for other nodes.

[2] *In a more general case, in order to combat channel degradations, instead of a fixed frequency, each link will be assigned a distinct frequency hopping pattern. Using frequency hopping will separate transmissions in the frequency domain, and at the same time reduce vulnerability of the links to channel degradations due to intentional and unintentional jamming, such as channel fades and hostile jamming, as well as self-interference. Therefore, our system is really a variation of a hybrid TDMA/code-division multiple access (CDMA) with CDMA realized as frequency hopping. The details of the design of the spread spectrum signaling for this system is beyond the scope of this article.*

[3] *Note that it is possible for some of the nodes in the network to never find a neighbor, and not attach to the wireless network at all. This is an acceptable phenomenon. The goal of the startup algorithm is to automatically form an infrastructure that will support local and long distance transport of sensor information. The percentage of nodes that will not get connected is a function of node density, transmit powers, and terrain type.*

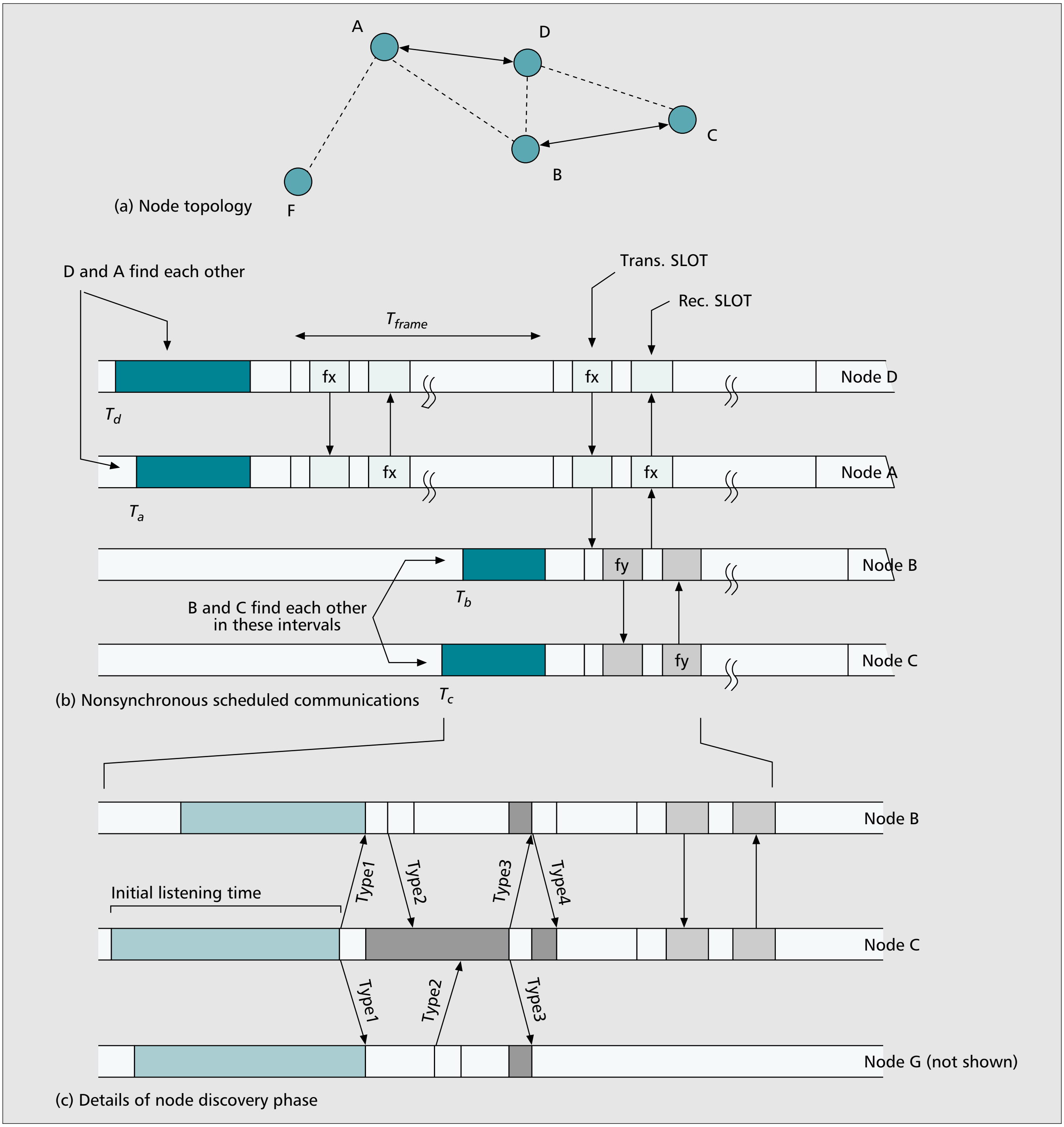


Figure 2. *Link-layer self-organizing procedures.*

List of Startup messages

The following messages are exchanged between nodes when they are searching for new neighbors:

- *TYPE1*: a short invitation containing a node's id and number of attached neighbors. The node which sends it is the *inviter* during the search transaction.
- *TYPE2*: a response to TYPE1. The node that sends it will be an *invitee*. There may be more than one invitee for each inviter. This message gives the inviter and invitee's addresses, and invitee's attached state.
- *TYPE3*: response to TYPE2. It indicates which invitee was chosen. It contains the following additional information depending on the node's attached state:
  –Inviter not attached: none
  –Invitee, inviter attached: inviter's schedule and frame epoch
  –Invitee not attached, inviter attached: proposed channel for the link, calculated by inviter
- *TYPE4*: response to TYPE3. Message contents are as follows:
  –Invitee not attached, inviter not attached: channel determined by the invitee
  –Invitee not attached, inviter attached: none
  –Invitee attached, inviter not attached: channel determined by the invitee

–Invitee attached, inviter attached: channel determined from own and inviter's schedule information

Mobile MAC Issues — As the stationary network becomes fully formed, it is possible that mobile nodes will begin to interact with the network. While adding to the overhead accompanying topological variability, mobile nodes further the functionality of the network, and thus their existence is desired. The goal of the mobile MAC protocol presented here is to provide the required connectivity to mobile sensors as they interact with the static network, while adhering to the constraints for the entire stationary network.

Mobility management within wireless networks has been studied extensively, with each network manifestation resulting in new methods of handling the ORM tasks. The mobile management issues in MANETs, for example, have classically been oriented toward routing issues within the network. Since the network consists solely of mobile nodes, the tasks of routing and mobility within the MANET are generally handled jointly. One way devised to handle these networks is to group the mobile nodes into small clusters, electing a cluster head to which to route information in a local neighborhood [15, 16]. The group of cluster heads in the entire network in turn forms a subnetwork. Information is then routed through this subnetwork. As mobile nodes move from one area to the next, they may decide to register within a new cluster and continue operation as usual.

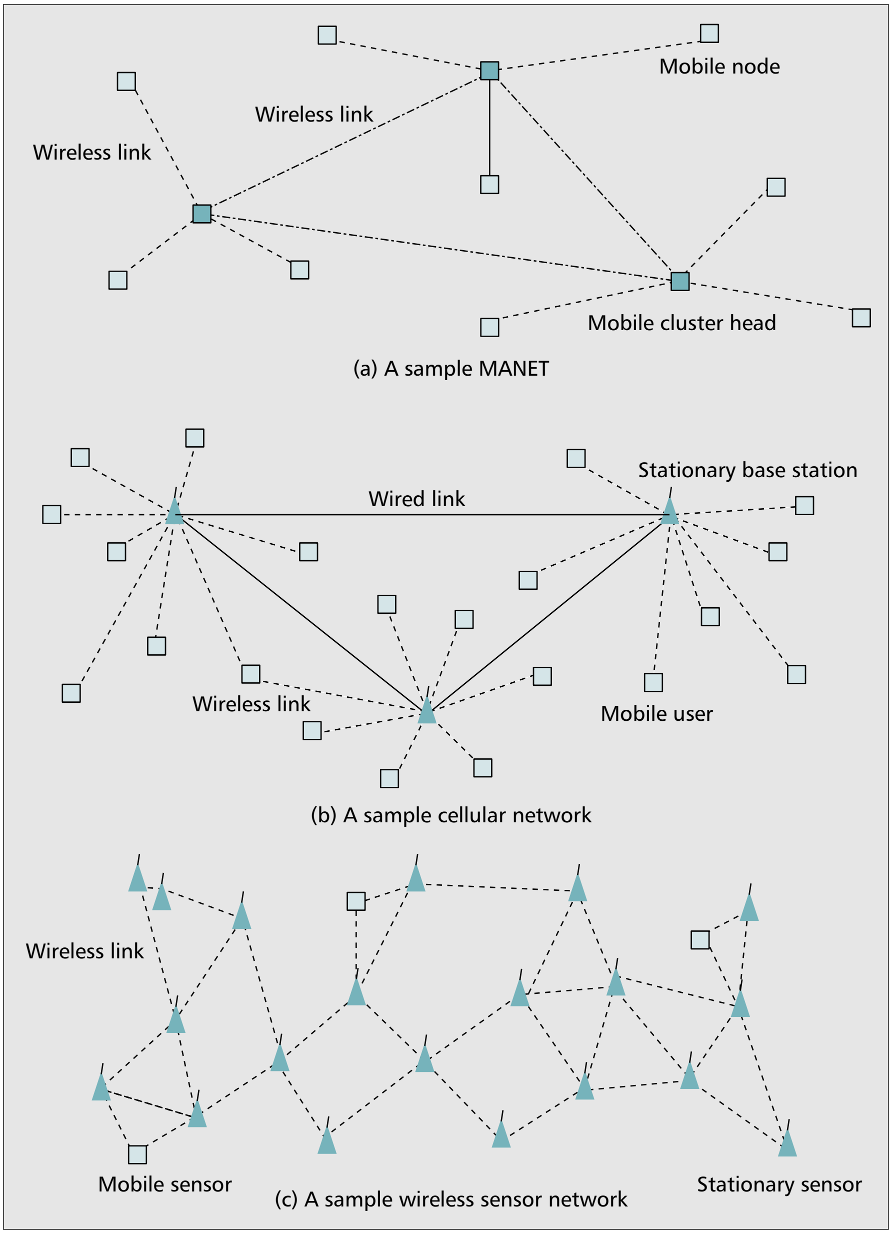


Figure 3. *Various wireless networks.*

Cellular systems are structurally quite different than conventional MANETs. The wired backbone on stationary nodes facilitates routing, as the wireless channel is avoided. Consequently, it is only the single hop from a mobile node to the stationary base station that needs to be considered. Thus, mobility management is primarily considered here from the point of view of forming connections with the best base station. As mobile users travel from the vicinity of one base station to the next, the desired connection is simply updated using handoff techniques, and communication continues as normal [17, 18]. Since the base stations are assumed to have a large energy reservoir, they take on much of the responsibility for mobile management (setting up new routes to the mobile nodes, informing mobile nodes of handoffs, etc.).

Although studies have been done to explain the handling of the ORM tasks for various networks, the properties of the networks are vastly different than those being investigated here. MANETs, in particular, are in the true sense ad hoc networks, but the absence of stationary nodes makes it difficult to simply use their algorithms for handling our mobility management. The nodes themselves are assumed to have a large range (on the order of hundreds of meters), focusing less on power consumption and more on network connectivity since the topology changes quite rapidly. Although cellular systems do introduce a stationary infrastructure, the mobile nodes greatly outnumber the stationary base stations. This implies that the base station will assume many of the tasks in maintaining the required connectivity between the mobile nodes and their serving base station. Figure 3 shows typical scenarios for each of the three system types mentioned here.

The EAR Algorithm's Motivation

Mobiles that have been introduced into the system function as extensions to the stationary sensor network. It cannot be assumed that each mobile node is aware of the global network state and/or node positions. Also, a mobile node may not be able to complete its task (data collection, network instruction, information extraction) while remaining motionless. Thus, the EAR protocol attempts to offer continuous service to these mobile nodes under both mobile and stationary constraints.

Mobile connections to a vast wireless sensor network can arise in many scenarios where either energy or bandwidth is a major concern. Where there is the constraint of limited power consumption, small, low-bit-rate data packets can be exchanged to relay data to and from the network whenever necessary. In this way, the low-power EAR protocol allows for operations to continue within the stationary network while intervening at desired moments for information exchange.

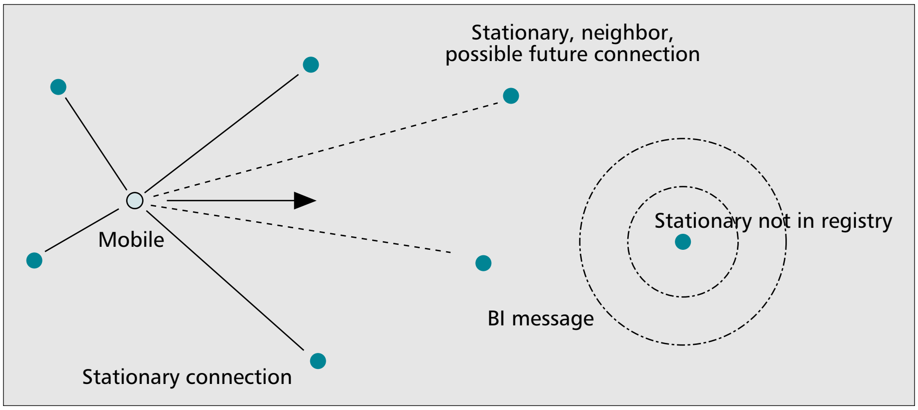

Stationary, neighbor, possible future connection
Mobile
Stationary not in registry
BI message
Stationary connection

Figure 4. *General mobile activity.*

Network Constraints

Because the primary limitation is that of battery power on stationary nodes, the communication channels between the mobile and stationary sensors must be established with as few messages transmitted by the stationary sensors as possible. This can be accomplished by allowing the mobile node to determine when to invite the stationary node as a connection, as well as when to drop a connection.

The network is assumed to consist primarily of stationary nodes, with few mobile nodes, all of which are randomly distributed. Such an assumption leads to the notion that only a select few stationary sensors will be within the vicinity of a mobile sensor at any given time. Giving the ability to form connections to the stationary nodes would result in constant specialized signaling with the intent of inviting mobile nodes to join the network. To avoid the unnecessary use of power associated with lost messages, the mobile nodes assume full control of the connection process. Furthermore, the overhead associated with acknowledgments can be eliminated. This is possible since the proximity between sensors almost surely ensures message reception.

In many situations, a handoff may not even be required. By exploiting the tight stationary sensor packing (within 10–20 m of each other), the mobile sensor can maintain its connections while being aware only of the sensors in the near field, handing off when one of the received signal-to-noise ratio (SNR) values along a current connection drops below a predetermined threshold. Thus, the mobile sensor will keep a registry of the surrounding nodes, selecting a new connection only when absolutely necessary.

Since there will be few stationary nodes aware of the presence of the mobile nodes, the EAR protocol will be transparent to the existing stationary protocol. This allows the functionality of the stationary protocol to remain fixed until the interjection of a mobile node. Also, by placing the mobile MAC protocol in the background, very few specialized messages need to be invented to establish, or drop, connections. Also, we consider here the prospect of giving the mobile nodes higher priority for forming connections. We assume that the stationary nodes are using a TDMA-like frame structure, within which slots are designated for inviting neighboring nodes into the network. By reserving the first slot following an invitation for mobile sensor connections, we can effectively assign higher priority to the mobile nodes.

The EAR Algorithm

During some predetermined slot in the TDMA-like frame structure in the stationary MAC algorithm, the stationary node should transmit some type of invitation message to the surrounding neighborhood, with the intent of inviting new stationary nodes to join the local network. This message need not occur in every epoch of the TDMA structure; it is only needed at some semi-regular interval, and serves as the “pilot signal” for the mobile nodes. Thus, no specialized message is required to initiate the connection procedure. Since the stationary node does not require a response to this message (although it will wait for a predetermined time for a response), the mobile node is simply “eavesdropping” on the control signals in the stationary MAC protocol. It must then decide the best course of action regarding the transmitting stationary sensor; hence, this invitation message will act as the trigger for the EAR algorithm.

In order to keep a constant record of neighboring activity, the mobile node will form a registry of neighbors. This registry will hold only the required information for forming, maintaining, and breaking connections. Since the registry will only comprise stationary nodes corresponding to signals received by the mobile, the mobile node will have information about the stationary nodes in the immediate neighborhood. From the transmitted invitation message the mobile can extract the received SNR, node ID, transmitted power (in a power-controlled scheme), and so on. Making, or breaking, a connection is based on the status of connections, as well as the location and mobility information inferred from the entries in the registry. Figure 4 depicts a typical situation of a mobile node, showing current as well as future connections.

The stationary node will maintain a registry as well, although its role is minimal compared to that of the mobile node. The stationary node will simply register mobile sensors that have formed connections and remove them when the link is broken, effectively limiting participation in the connection procedures.

To design a system in which the mobile assumes full responsibility for making and breaking connections, a novel signaling method must be defined. If the invitation message, which is inherently part of the stationary MAC algorithm, is included as a shared message, the EAR algorithm makes use of the following four primary messages:

Broadcast Invite (BI) — The stationary node invites other nodes to join.

Mobile Invite (MI) — The mobile responds to BI to request a connection.

Mobile Response (MR) — The stationary node accepts the MI request.

Mobile Disconnect (MD) — The mobile informs the stationary node of a disconnect; no response is needed.

Acknowledgments are avoided by taking appropriate precautions, such as timeouts, to prevent lost messages from incorrectly identifying connections and neighbors. The stationary nodes are only responsible for the transmission of one specialized message within the EAR algorithm. This reduces the power expense in forming and breaking connections between mobile and stationary nodes.

A newly introduced mobile node will begin its connection protocols upon reception of the stationary node’s BI message. The stationary node is registered, and a decision is made, depending on the present connection status of the mobile node as well as the potential link quality between the mobile and stationary nodes, on whether to request a new connection. If a connection is not requested, the associated stationary node is simply held in the registry. If a connection is in fact requested, the mobile node awaits a response, while continuing to listen for invitation messages. The mobile node will continue to register every stationary node encountered, until its registry becomes full (a registry size is predetermined). At this time, new stationary nodes will have to contend for a place within the registry by a simple comparison scheme, possibly replacing a node with inferior channel quality.

Upon receipt of the MI message, the stationary node will determine whether a connection is possible. If so, slots are selected along the TDMA frame for communication, and a reply is sent to the mobile node accepting the connection. Simultaneously, the stationary node will enter the mobile node in its own registry. It is possible, but unlikely, that the stationary node will reach the entry limit in its own registry (again, its size is predetermined). Similarly, it may not have a communications slot available that coincides with those presented to it by the mobile node. In such cases, a decline is sent to the mobile node.

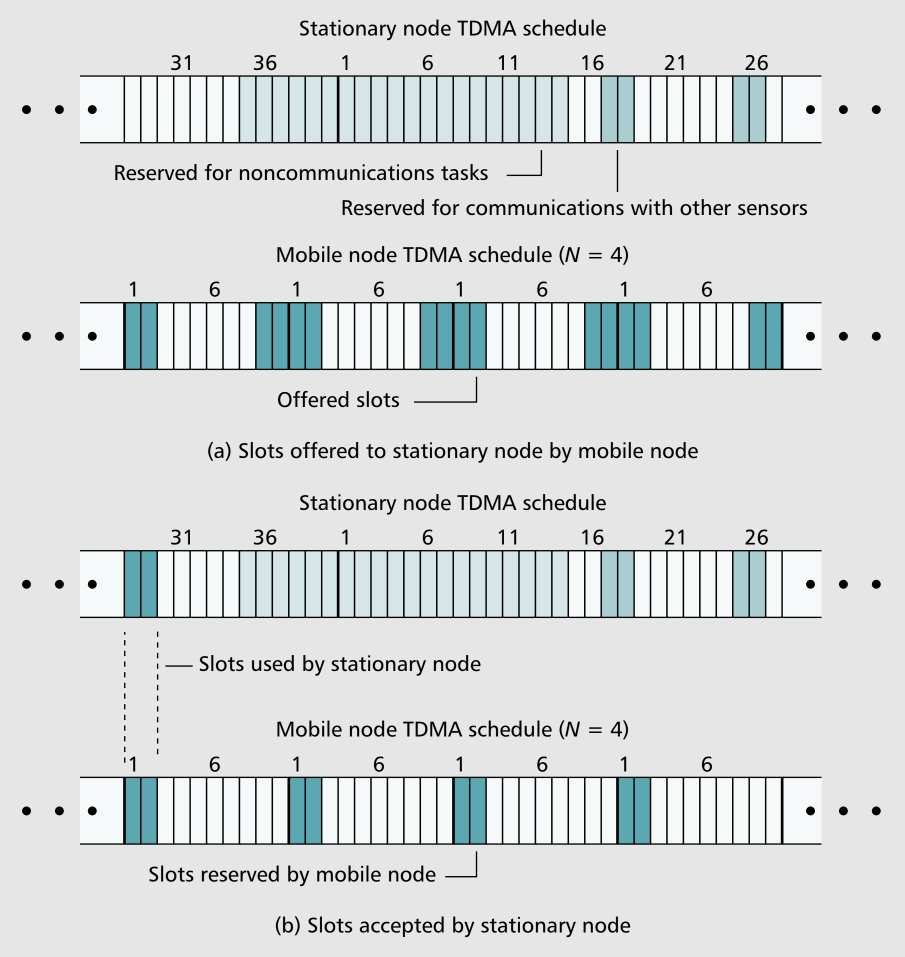

Stationary node TDMA schedule
31
36
1
6
11
16
21
26
Reserved for noncommunications tasks
Reserved for communications with other sensors
Mobile node TDMA schedule (N = 4)
1
6
1
6
1
6
1
6
Offered slots
(a) Slots offered to stationary node by mobile node
Stationary node TDMA schedule
31
36
1
6
11
16
21
26
Slots used by stationary node
Mobile node TDMA schedule (N = 4)
1
6
1
6
1
6
1
6
Slots reserved by mobile node
(b) Slots accepted by stationary node

Figure 5. *Slot assignment for mobile connections.*

It is likely that the mobile node will receive many BIs from registered stationary nodes. Instead of simply dropping the message, the mobile node uses this new information to extract information about the channel quality, and thus its general proximity to the stationary sensor. As the received SNR along the channel improves or degrades, the mobile sensor may wish to request a connection or disconnection (with an MD). The mobile decides to which nodes to request connections, and from which nodes to disconnect, based on predetermined thresholds.

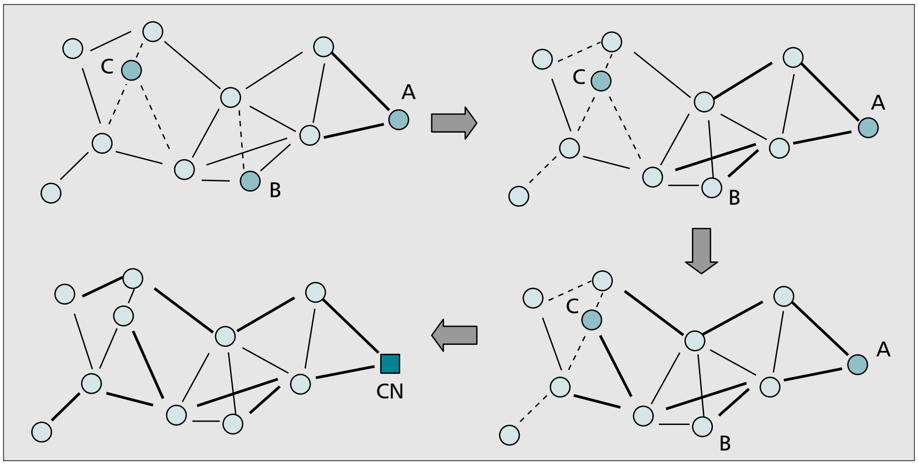
In the EAR algorithm, two threshold values are used to avoid the ping-pong effect: connect and disconnect thresholds. As an unconnected stationary sensor's received SNR rises above the connection threshold, a connection is considered. Similarly, as a connected stationary sensor's SNR drops below the disconnection threshold, an MD is sent. A high connection threshold will generally yield an overall higher quality of link within the network since the received SNR is forced to be higher; but the probability of outage is increased because the requirements for forming a connection are more stringent. By raising the disconnection threshold, again a higher average SNR is attained within the network, although the mobile sensor will drop the connection more often, resulting in higher overhead due to signaling.

Since it is sometimes difficult to adjust the registries due to inconsistencies in signal reception, the mobile node employs a set of timeouts to limit registry errors. When a connection to a stationary node is requested, the mobile node updates the connection status to PENDING. It is possible that this invitation message is lost in transmission, resulting in the mobile maintaining the PENDING status indefinitely. Thus, if a response is not received within a specified time frame, the mobile node will downgrade the stationary node's status to NOT-CONNECT. Furthermore, once a connection has been established, if information is not readily available for extraction from the network, the mobile node will rely on reception of the BI messages to update the connection status. Since the BI messages are not sent regularly, it is possible that the mobile node will quickly move out of range of a neighboring stationary node. If this happens, the mobile sensor will drop the connection after a predetermined waiting period.

MAC/TDMA/Bandwidth Utilization

Because the mobile node primarily uses its schedule for mobile–stationary communications, it is inefficient to use similar TDMA schedules for each type of node. A possible solution is to allow the mobile node's frame length to be an integer fraction (N) of that of the stationary node. The mobile node may offer R slot pairs for communication, resulting in $R * N$ options for the stationary node, any number of which may be chosen. Although communications may not occur during each of the mobile node's N frame repetitions, the associated slot is always reserved. Figure 5 depicts a typical request for a connection by a mobile node, with $N = 4$ and $R = 2$. Here, only one slot pair is accepted, causing the mobile node to make the reservation and communicate during every fourth instance of its frame period.

Routing — Because the mobile nodes interact with the network, it is possible that they become involved in the routing paths calculated at the network layer. For information sources, such as robotic data collectors and instructional personnel, routing is not an issue since the only goal is to place the information on the network, allowing the stationary nodes to route the information to the required destinations. If the mobile node is used as an information sink, though, routing tables have to be devised to allow information to efficiently reach



■ **Figure 6.** *Diffusion of candidates' information under SWE.*

the user. If the degree of mobility is relatively slow, new routing trees can be calculated as the mobile moves from location to location. To avoid unnecessary recomputations, however, it is possible to simply recompute the routing trees in the locale of the mobile node. Since this tends to become inefficient when the mobile moves some distance from its original location, a new complete routing tree will be calculated only when necessary. For both multihop as well as cooperative network routing, efficiency can be improved in three areas: route setup, route maintenance, and service

However, there is generally a trade-off among them. Complex route computations may find energy-efficient paths, but are expensive to maintain as network topology changes. Therefore, energy efficiency should be emphasized in each area to the degree appropriately matching its importance in meeting the overall objective. For multihop routing, the objective is to provide priority service with robustness on a long-term basis; therefore, more energy will have to be spent on route setup and route maintenance to meet these requirements. On the other hand, for a noncoherent cooperative function network, where data traffic is light, optimization of energy cost on each route is not nearly as important as reducing overhead during route setup.

Multihop Routing

Two multihop routing algorithms have been proposed for MANET: Ad Hoc On Demand Distance Vector (ADOV) routing and Temporally Ordered Routing Algorithm (TORA). Both are examples of demand-driven systems that eliminate most of the overhead associated with table update in high-mobility scenarios. However, it has high energy cost during route setup (path discovery). Since our system does not deal with high mobility, it is in the interest of energy efficiency to go with a table-driven system. Another algorithm, called Power-Aware Routing [19], finds the minimum metric paths on two different power metrics:

- Minimum energy per packet
- Minimum cost per packet

The first metric is intuitive and produces substantial energy saving *while the network retains full connectivity*; however, performance degradation due to node/link failure is not accounted for. The minimum cost metric is obtained by weighting the energy consumption by the energy reserve on each node. It has the nice property of delaying failures by steering traffic away from low-energy nodes; however, overhead for path maintenance could be high.

To improve energy efficiency in a low-mobility network, we turn to a table-driven multipath approach. The degree of failure protection is directly related to the degree of disjointedness, k, of the paths joining a node to a data sink (i.e., the number of paths with no common branches). A k-disjoint structure can protect against failures of k links or nodes. As a rule of thumb, to generate a k-disjoint structure requires about k times the overhead complexity of a shortest path algorithm [20]. However, the disjoint property creates strong coupling between routing tables that makes a localized recovery scheme nearly impossible. The key to reducing overhead is to loosen up this coupling effect by relaxing the disjoint requirement outside the one-hop neighborhood of the sink. Although the degree of failure protection is lower, it can be compensated for by a localized path restoration procedure at much lower energy cost.

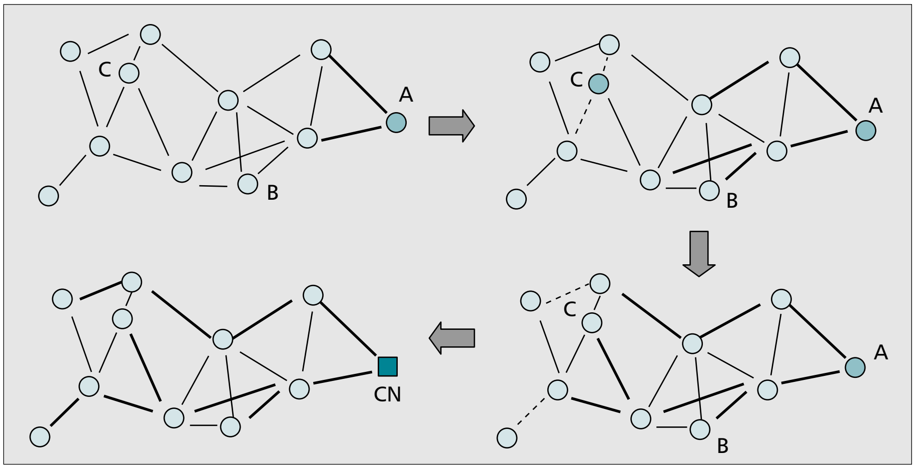
To create multiple paths from each node to the sink, multiple trees, each rooted from a one-hop neighbor of the sink, are built. Each tree will be forced to grow outward from the sink by successively branching, whenever possible, to neighbors at higher hop distances from the sink while avoiding nodes with very low QoS and energy reserves. At the end of the tree-building procedure, most nodes will belong to multiple trees and thus have multiple paths disjoint inside the one-hop neighborhood of the sink. The advantage of this structure is that it allows each sensor indirect control of which one-hop neighbor of the sink will relay a message. For each node, two parameters are associated with each path:

- Energy resources estimated by maximum number of packets that can be routed without energy depletion if it has exclusive use of the path
- Additive QoS metric where a higher metric implies lower QoS

Having multiple paths to the sink node, each sensor uses an SAR algorithm for path selection. It takes into consideration the energy resource and QoS on each path, and the priority level of a packet. Path selection is made by the node that generates the packet, unless topology change down the path requires the packet to be diverted. Each link contributes an energy cost and delay, and thus a resistance to packet flow that can be captured in an additive metric for any given path. Against this a packet will have credits so that it can achieve priority in using, for example, paths that are low latency, but traverse nodes with depleted energy. For each packet routed through the network, a weighted QoS metric is computed as the product of the additive QoS metric and a weight coefficient associated with the priority level of that packet for purposes of performance evaluation. The intuitive interpretation of this weighted QoS metric is that *it measures the QoS provided to each packet relative to the priority level of the packet*. Therefore, to maintain the same weighted QoS metric, higher QoS (lower QoS metric) will be used for higher-priority (higher weight coefficient) packets. The objective of the SAR algorithm is to minimize the average weighted QoS metric throughout the lifetime of the network.

As each path is used over time, the available energy resource will change. There are also possible changes in the QoS on each path. These changes will be accounted for by a periodic metric update triggered from the sink node. Simulation study [21] shows SAR has better performance than the minimum metric algorithm, which optimizes performance by focusing, very singularly, on lowering energy consumption for each packet without considering its priority.

Failure recovery is implemented by a handshaking procedure that enforces routing table consistency between the upstream and downstream neighbors on each path so that any local failure will automatically trigger a recomputation procedure locally. This procedure will converge as long as a path exists in the network topology [10]. In order to prevent the possibility of slow convergence (i.e., the counting to infinity problem), a threshold method detects rapid increase of path metric and speeds up convergence to infinity, which effectively marks the erasure of a path. This can conserve energy for nodes that are separated from the sink but may later reestablish connection again.



■ **Figure 7.** *Another SWE election process.*

Adaptive Local Routing for Cooperative Signal Processing

We assume that an application-level algorithm or outside agent will determine which cooperative function is needed and trigger the network formation process. In the following section, the term *network* refers specifically to a connected set of sensors that detected a common target. Before describing the network formation algorithm, a few remarks on the basic categories of environmental stimuli and cooperative functions are warranted.

In general, environmental stimuli can be separated into two major categories: near-field (NF) and far-field (FF). Near-field stimuli have short range relative to the baseline width of sensor groups within detectable distance. Signal propagation is dominated by the line-of-sight component; therefore, SNR of sensor data can be modeled in the form: $k\ d^{-r}$, where d is the distance between the sensor and the signal source, and k and r are constants determined by the propagation medium. Accurate localization and identification are possible if the target is located inside the convex hull of the network. Far-field targets are located at much farther distances relative to the baseline width of the network. For these targets, source localization and range estimation are much more challenging. Due to greater physical distance from the network, signals encounter both increased dispersion and attenuation.

There are two types of cooperative signal processing techniques: noncoherent and coherent. For noncoherent processing, raw sensor data will be preprocessed at each node to extract a small set of parameters to be forwarded to a central node (CN) for further processing; for coherent processing like blind beamforming [22], raw sensor data, after minimal preprocessing, will be tagged with a timestamp and uploaded through the local network to the CN for more intensive computations. Although energy efficiency is the ultimate goal, different approaches can be used depending on which cooperative functions are used. Noncoherent functions have fairly low data traffic loading; therefore, we will focus our effort on improving algorithmic efficiency. On the other hand, since coherent processing generates long data streams, energy efficiency must be achieved by path optimality. For clarity of presentation, we separately discuss coherent and noncoherent processing networks.

Noncoherent Cooperative Function — In general, there are three phases in the processing network formation process:

- Phase I: Target detection, data collection, and preprocessing
- Phase II: Membership declaration
- Phase III: Central node election

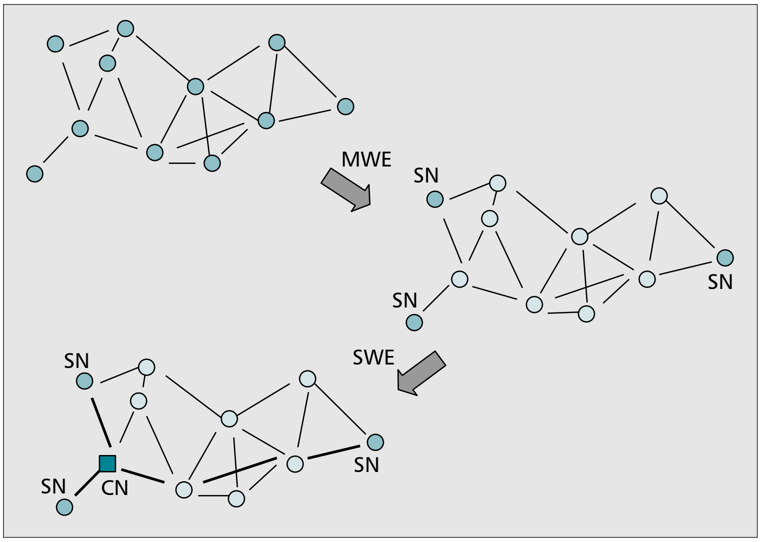
During phase I, a target is detected, its data collected and preprocessed. Although the sink node can override any decision made on the local level, the results of preprocessing can serve as good indicators of whether a node should participate in a cooperative function. One such indicator is the SNR. When a node decides to participate in a cooperative function, it will enter phase II and declare this intention to all neighbors. This should be done as soon as possible so that each sensor has a local understanding of the network topology. Phase III of the formation process is the election of the CN. Since the CN is selected to perform more sophisticated information processing, it must have sufficient energy reserves and computational capability. It can also be selected based on SNR, which is a good estimator for distance to the target in the NF case. The CN election algorithm has two components:

- The Single Winner Election (SWE) algorithm
- The Spanning Tree (ST) algorithm

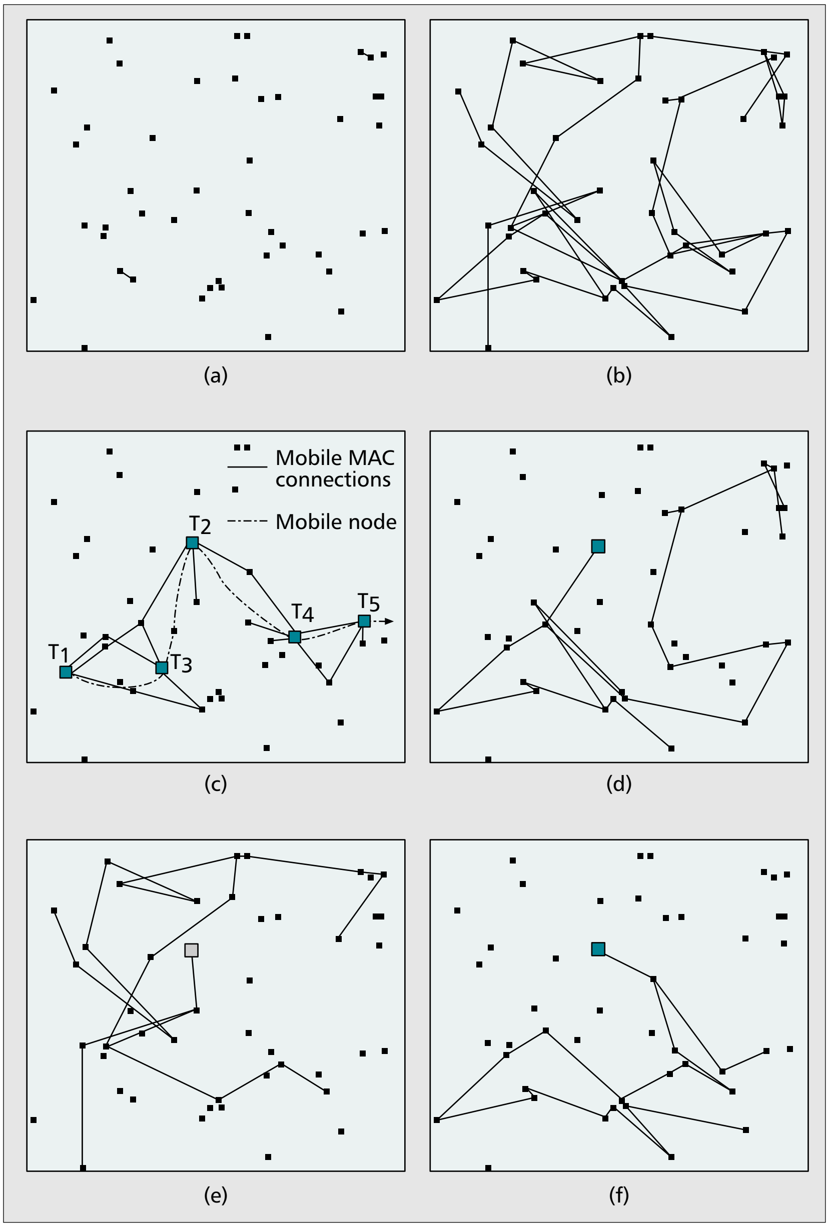
The first component handles the necessary signaling to facilitate the exchange of candidate information; the second component computes a minimum-hop spanning tree rooted at the CN. By piggybacking election and routing information together in an Elect message, it is possible to execute both algorithms concurrently.

Each Elect message identifies a potential CN candidate and a set of parameters that serve as the election criteria by which candidates are compared. In the initial stage of the SWE process, each node may impose a voluntary delay of varying length before announcing itself as a CN candidate by broadcasting Elect messages. In response to the first batch of Elect messages, those nodes that received them start comparing the proposed CN candidates with themselves and respond with a second batch of Elect messages, which carries the result of this initial comparison. The second batch of message passing will likely spawn further exchange of messages. During this process, for each message that presents a better candidate, its information will be recorded in the registry and then forwarded to all neighbors; otherwise, the message is discarded. Figure 6 shows how the continual exchange, forwarding, and discarding of Elect messages allows the winning candidate's information to "diffuse" throughout the network. Together with this diffusion process, a minimum-hop spanning tree rooted at the winning candidate will gradually increase its coverage. By the end of the SWE process, a minimum-hop spanning tree will completely cover the network.

An overhead–delay trade-off exists such that if each candidate voluntarily delays itself based on its likelihood to win the election (i.e., value of the election criterion used,) the diffu-



■ **Figure 8.** *The formation process for coherent routing.*



■ **Figure 9.** *Simulation of the behavior of various protocols.*

sion process of the Elect messages for the better candidates will have a head start. This simple mechanism can eliminate many local Elect message exchanges among losing candidates, and greatly reduces overhead (compare Figs. 6 and 7). When sufficient delay difference exists between the best candidate and the rest of the network, Elect messages of the winner can cover the entire network without opposition, thus achieving minimum overhead. Simulation experiments showed that the local network formation process is quite scalable when some formation delay can be tolerated.

Coherent Cooperative Function — The coherent algorithm differs from the noncoherent one in two respects:

- Limited number of sensors generating data
- Explicit computation of minimum energy paths

Since the energy cost of uploading long data stream to the central node is high, an MWE process is used to limit the number of sensor source nodes (SNs) providing the data. The MWE process is a simple extension of the SWE process. Instead of keeping a record of *one* best candidate, each node will now keep up to *n* of them. Just as in the noncoherent case, for each winning SN candidate, a minimum-energy path can be computed by piggybacking link power information on the Elect messages. At the end of the MWE process, each sensor in the network has a set of minimum-energy paths to each SN. Then the total energy consumption to upload data from each SN to each node in the local network can be computed.

Using this energy consumption figure as the election criterion, an SWE process can be used to find the node that yields the *minimum energy consumption*. This node can then serve as the CN for the coherent cooperative function. In general, the formation process has longer delay, higher overhead, and lower scalability than for noncoherent processing networks. Figure 8 illustrates the formation process.

The Simulation Implementation

A simulation testbed for the above protocols was implemented in Parsec [23]. In this simulation, a radio propagation model complete with shadowing and path loss is used. The simulation is capable of running packet-level experiments to test the behavior of the algorithms. The simulation is able to accommodate simulations of hundreds of nodes at the moment. The simulation environment models each node as a separate Parsec entity. The functionality of each layer (MAC, mobile MAC, and network) is implemented as a function inside the node.

A network consisting of 45 nodes, scattered randomly in space, with density $\lambda = 0.04$ nodes/m^2, was simulated, as shown in Fig. 9a. In this simulation, the sensor nodes are using 1 mW transmit power, $T_{\text{frame}} = 8.0$

s, and 100 frequency bands are available. Path loss follows a fourth-power dropoff with distance law, and the shadowing variance is 8 dB. Figure 9b gives the state of the network links at the moment it becomes connected.

In Fig. 9c the behavior of the mobile MAC is shown. The mobile node is traveling at a velocity of 0.1 m/s, with the capability of having 10 neighbors registered, but limited to only three connections. The connection threshold is set at a received SNR level of 12 dB, with the disconnection threshold at 7 dB. The figure shows the track of a mobile and its link-level connections maintained by the Mobile MAC protocol at five sample points $\{T_1, T_2, T_3, T_4, T_5\}$.

Figures 9d, 9e, and 9f show three spanning trees connecting the sensor to the mobile which has declared itself as a sink node at time T_3. Each spanning tree is created from a distinct one-hop neighbor of the sink, and the requirement to branch to higher hop distance is relaxed when the tree is small. At such an early stage of network formation, when the average network degree is only 2.13 (as depicted in Fig. 9b), only 14 of 45 (roughly 31 percent) of the sensors have multiple paths to the sink. However, as SMACS continues to pick up new link-level connections, the average degree as well as the multipath coverage will continue to improve until the topology becomes stabilized. Note that in all these cases, in order to keep the diagrams clear, the existing underlying links are not shown.

Conclusion

We have presented a set of algorithms for establishing and maintaining connectivity in wireless sensor networks. The algorithms exploit the low mobility and abundant bandwidth, while coping with the severe energy constraint and the requirement for network scalability. The algorithms further accommodate slow mobility by a subset of the nodes. However, many important research questions remain, including bounds on the minimum energy required for network formation, especially taking into account the interactions with the signal processing functions. Another issue is the extent to which the algorithms can efficiently deal with more extensive mobility in the nodes and targets.

The most fundamental open question is that of hierarchy in the distributed signal processing and networking functions. It is clear that some layering of signal processing functions is required to produce energy-efficient operation. We can afford neither to constantly run the most expensive signal processing algorithms, nor the poor decision quality that results from relying only on the simplest procedures. Since communications dominates the energy cost when cooperative functions among nodes are needed, the question naturally arises as to the extent to which the signal processing hierarchy demands a corresponding networking hierarchy. We have developed substantially different algorithms for setting up subnetworks to perform cooperative signal processing functions, with the effort involved and scalability depending quite strongly on the signal processing function. However, this is only the first venture in exploring a very rich space of problems. Hardware testing of alternative algorithms in large networks is certain to yield many interesting challenges.

References

[1] G. J. Pottie and W. J. Kaiser, “Wireless Integrated Network Sensors,” *Commun. ACM*, vol. 43, no. 5, May 2000, pp. 51–58.

[2] May 2000 issue of *Commun. ACM*.

[3] D. Estrin, R. Govindan, and J. Heidemann, “Embedding the Internet,” *Commun. ACM*, vol. 43, no. 5, May 2000, pp. 39–41.

[4] G. Asada *et al.*, “Wireless Integrated Network Sensors: Low Power Systems on a Chip,” *Proc. 1998 Euro. Solid State Circuits Conf.*, 1998.

[5] F. Bennet *et al.*, “Piconet: Embedded Mobile Networking”, *IEEE Commun. Mag.*, Oct. 1997, pp. 7–15.

[6] G. J. Pottie, “Hierarchical Information Processing in Distributed Sensor Networks,” ISIT, Cambridge, MA, Aug. 1998, p. 163.

[7] http://www.bluetooth.com

[8] http://www.hoemrf.com/hrfwgtec.pdf

[9] K. Sohrabi, *On Low Power Wireless Sensor Networks*, Ph.D. dissertation, Dept. of Elec. Eng., UCLA, June 2000.

[10] J. L. Gao, “Energy Efficient Routing for Wireless Sensor Networks,” Ph.D. dissertation, Dept. of Elec. Eng., UCLA, June 2000.

[11] M. Gerla and J. T. Tsai, “Multicluster, Mobile, Multimedia Radio Network,” *Wireless Networks*, 1995, pp. 255–65.

[12] D. J. Baker and A. Ephremides, “The Architectural Organization of a Mobile Radio Network via a Distributed Algorithms,” *IEEE Trans. Commun.*, no. 11, Nov. 1981, pp. 1694–1701.

[13] A. D. Amis *et al.*, “Max-Min D-Cluster Formation in Wireless Ad-Hoc Networks,” *IEEE INFOCOM*, Mar. 2000.

[14] K. Sohrabi and G. Pottie, “Performance of a Novel Self-Organization Protocol for Wireless Ad-Hoc Sensor Networks,” *Proc. IEEE VTC*, Amsterdam, Netherlands, Sept. 1999.

[15] A. Iwata *et al.*, “Scalable Routing Strategies for Ad Hoc Wireless Networks,” *IEEE JSAC*, vol. 17, no. 8, Aug. 1999, pp 1369–79.

[16] G. Pei and M. Gerla, “Mobility Management in Hierarchical Multi-Hop Mobile Wireless Networks,” *Proc. 8th Int’l. Conf. Comp. Commun. and Networks*, Piscataway, NJ, 1999, pp. 324–29.

[17] A. Viterbi *et al.*, “Soft Handoff Extends CDMA Cell Coverage and Increases Reverse Link Capacity,” *IEEE JSAC*, vol. 12, no. 8, Oct. 1994, pp. 1281–88.

[18] D. Wong and T. J. Lim, “Soft Handoffs in CDMA Mobile Systems,” *IEEE Pers. Commun.*, Dec. 1997, pp. 6–17.

[19] S. Singh, M. Woo, and C. S. Raghavendra, “Power-Aware Routing in Mobile Ad Hoc Networks,” *MOBICOM ’98*, Dallas, TX, pp. 181–90.

[20] J. W. Suurballe, “Disjoint Paths in a Network,” *Networks*, vol. 4, Wiley, 1974, pp. 125–45.

[21] K. Sohrabi *et al.*, “A Self-organizing Wireless Sensor Network,” *Proc. 39th Annual Allerton Conf. Commun., Control, and Comp.*, Urbana, IL, Oct. 1999.

[22] K. Yao *et al.*, “Blind Beamforming on a Randomly Distributed Sensor Array System,” *IEEE JSAC*, vol. 16, no. 8, Oct. 1998.

[23] http://may.cs.ucla.edu/projects/parsec

Biographies

KATAYOUN SOHRABI [S’87] (sohrabi@ee.ucla.edu) received B.S. and M.S. degrees in electrical engineering from the University of Missouri, Rolla in 1991 and 1993, respectively. She expects to received her Ph.D. from UCLA’s Electrical Engineering Department in 2000. She was IEEE student chapter treasurer at UMR for 1991–1992. Her areas of interest include channel access algorithms and protocols for wireless networks, energy-sensitive algorithms for sensor networks, spread spectrum techniques, and RF channel modeling.

VISHAL AILAWADHI (vishal@ucla.edu) received his B.S. and M.S. degrees in electrical engineering from UCLA in 1996 and 1998, respectively. He is currently pursuing his Ph.D. in electrical engineering from UCLA. His research interests include low-power mobility support algorithms and protocols, including medium access concerns, for wireless sensor networks.

JAY L. GAO received his B.S. and M.S. degree in electrical engineering from UCLA in 1993 and 1994, respectively. He is currently a doctoral candidate in electrical engineering at UCLA, and his research area includes low-energy sensor multihop routing and local adaptive routing for cooperative signal processing.

GREGORY J. POTTIE [S’84, M’89] received his B.Sc. in engineering physics from Queen’s University, Kingston, Ontario, Canada, in 1984, and his M.Eng. and Ph.D. in electrical engineering from McMaster University, Hamilton, Ontario, in 1985 and 1988, respectively. From 1989 to 1991 he worked in the transmission research department of Motorola/Codex in Canton, Massachusetts, with projects related to voiceband modems and digital subscriber lines. Since 1991 he has been a faculty member of the UCLA Electrical Engineering Department, where he is now professor and vice-chair for undergraduate programs. His research interests include channel coding theory, wireless communication systems, and wireless sensor networks. Current projects include development of distributed algorithms for cooperative tasks in sensor networks, including cooperative communication, data fusion, and energy-aware network self-organization and routing. From 1997 to 1999 he was Secretary to the Board of Governors for the IEEE Information Theory Society. In 1998 he was named the faculty researcher of the year for the UCLA School of Engineering and Applied Science. He is a co-founder of Sensoria Corporation.