CHAPTER 11

# Energy-Efficient Backbone Construction, Broadcasting, and Area Coverage in Sensor Networks

DAVID SIMPLOT-RYL

IRCICA/LIFL, Universite Lille, Villeneuve d'Asc4, France

IVAN STOJMENOVIĆ

University of Ottawa, Ontario, Canada

JIE WU

Florida Atlantic University, Boca Raton, Florida

*A backbone is a subset of sensors that is sufficient for performing assigned tasks. The exact definition depends on the task or the particular desirable properties of a backbone. We discuss two specific kinds of backbones, neighbor and area* dominating sets, *that we believe are the essential and perhaps only backbones required for sensor networks. A sensor is covered by a backbone if it is in the backbone or is a neighbor to a sensor in the backbone. This type of backbone is referred to here as* neighbor-dominating sets, *or simply dominating sets. A point within a monitoring area is covered by a sensor backbone if it is within sensing range of at least one sensor from the backbone. This type of backbone is called* area-dominating set. *In a* broadcasting *(also known as* data-dissemination*) task, a message is sent from one node, which could be a monitoring center, to all nodes in the network. Sensors, which are randomly placed in an area, decide which of them should be active and monitor an area, and which of them may sleep and become active at a later time. The communication connectivity is important so that the measured data can be reported to a monitoring center. This problem is known as the* sensor-area coverage *problem, and needs to be solved efficiently to enable sensor functioning for a prolonged time. Sensors may also be placed deterministically in an area to optimize coverage and reduce power consumption. Most solutions considered in this chapter are based*

*Handbook of Sensor Networks: Algorithms and Architectures*, Edited by I. Stojmenović
ISBN 0-471-68472-4 Copyright © 2005 John Wiley & Sons, Inc.

*on constructing area-dominating sets for sensor-area coverage. The best known solutions for backbone construction, broadcasting, and sensor-area coverage problems are based on the concept of localized connected dominating sets. For instance, one solution to the broadcasting problem is that only nodes from a connected neighbor-dominating set retransmit the message. This chapter reviews solutions to these three related problems in sensor networks.*

## 11.1 INTRODUCTION

### 11.1.1 Modeling Sensor Networks

A widely accepted basic graph-theoretical model for sensor networks is the *unit-disk* graph model, defined as follows: two nodes, $A$ and $B$, in the network are neighbors (and thus joined by an edge) if the Euclidean distance between their coordinates in the network is at most $R$, where $R$ is the transmission radius that is equal for all nodes in the network. There are two kinds of unit-disk graphs considered in sensor networks, sensing and communication unit-disk graphs, with corresponding sensing and transmission radii, respectively. The relationship between sensing and transmission radii may vary based on the particular hardware or application. There are three basic cases: equal sensing and transmission (communication) radii, transmission radius more than twice the sensing radius, and a communication range that is between sensing and twice the sensing radii. Reasons for the three cases become apparent later in the chapter.

Some solutions make use of sensor ability to adjust transmission radius, instead of using the maximum radius, as determined by the unit-disk graph model. The unit-disk graph model is not fully realistic, but is much better for approximation of a sensor network than the random graph model (with each edge having equal probability of being selected for the graph), studied in ref. [1]. In the unit-disk graph model, the probability of receiving a packet between two nodes suddenly changes from 1 to 0 at distance $R$. A more realistic model is the fuzzy unit-disk graph proposed in ref. [2]. In this model, there are two transmission radii, $r$ and $R$. Two nodes communicate with each other if the distance between them is $\leq r$; they do not communicate with each other if their distance is $\geq R$, and may or may not communicate if the distance is between $r$ and $R$. In a realistic physical-layer model, such as the log normal shadowing model, random signal strength variations lead to a model where the packet reception probability $p(x)$ is a function of distance $x$ between two nodes. The transmission radius $R$ is defined in ref. [3] as the distance at which $p(R) = 0.5$. Two nodes are considered neighbors if the distance $x$ between them is such that $p(x) \geq w$, where $w$ is a threshold parameter (for example, when $w = 0.05$, then $x \approx 1.4R$). In the hitchhiking model [4], two transmission radii, $r$ and $R$, are also used. The receiver can receive a partial packet from the sender if their distance is between $r$ and $R$. The actual percentage of packet that can be decoded depends on a particular signal model. It is assumed that each receiver can assemble several partially received packets to one complete packet.

### 11.1.2 Localized Algorithms and Message Complexity

Among recently developed strategies for constructing small connected dominating sets, *localized protocols* offer the best prospect for achieving energy efficiency. In a localized protocol, each node makes protocol decisions based solely on some available local knowledge (to be more precise, based on the information from neighbors within $k$ hops for certain $k$), without resorting to global network information. Because of the dynamic nature of sensor networks, the topology changes are frequent and unpredictable. The local information must suffice for a sensor node to make protocol decisions; otherwise, the increased communication overhead could offset the energy savings and increase latency. In a *centralized* (or *globalized*) algorithm, one or more nodes (or a central entity like a base station) need to learn global node and/or edge structure, either the whole graph (for instance, to find a route using the shortest-path algorithm), or a global structure derived from the graph (such as minimal spanning tree, which can be used for optimal energy data aggregation). Because of the huge communication overhead involved in gathering such information in dynamically changing sensor networks, such protocols cannot be energy efficient solutions in normally large sensor networks. This chapter consequently discusses primarily localized solutions (some centralized algorithms are described only for the sensor-area coverage problem).

The sensor network may operate with or without time synchronization between sensors. In an *asynchronous* protocol, there is no common clock between the sensors. Therefore, each sensor makes its own decision about being active or going to a sleep state for an arbitrary period based on the overheard communication from other sensors. In a *synchronous* protocol, sensors follow a common clock, and therefore naturally may operate in rounds. In the case of sensor-area coverage, for example, they exchange some messages (at the beginning of each round) in order to decide which of them is needed for coverage in a given round, while the remaining sensors may sleep for the rest of the round and wake up at the beginning of the next round.

We further classify localized protocols according to the amount of information required and to overhead in the construction and maintenance phases. The amount of required information is related to the *message complexity*, which can be defined as the average number of transmitted messages per sensor node in a protocol. Although some protocols appear localized, an extensive message exchange with neighbors amounts to collection and use of global information. In a *strictly localized protocol* [5], all information processed by a node is either local in nature or global in nature, but obtainable in a short constant time by querying only the node's neighbors or itself. In other words, only a small constant number of message exchanges with neighbors is allowed. Strictly localized protocols may need some information that is part of their input (such as destination position in a routing protocol) but cannot use structures that are global in nature (e.g., information about which of the outgoing links belongs to the minimum spanning tree (MST)). An interesting similar definition is given in ref. [6]. An *emergent* algorithm is any computation that achieves formally or stochastically predictable global effects by communicating directly with only a bounded number of immediate neighbors and without the use of central control or global visibility [6].

The sensors may or may not use position information in their decisions. The availability of position information for proper sensor functioning was widely recognized as highly desirable; however, it is a nontrivial problem and the precision of the location information may impact the performance of a protocol. There exists a variety of position-determination protocols [7], with a variety of message complexities. If position information is used, we will make the simple assumption that it was provided to the node message tree (here only messages transmitted by sensors are counted), which is true only if it was provided externally via a global positioning system (GPS) or similar beacons arriving from the environment.

The simplest local information required is certainly no knowledge at all about existing neighbors. The *blind flooding* protocol for broadcasting (used in a typical route discovery in reactive routing protocols), where each node will retransmit the packet after receiving it the first time, belongs to this category. The next, and commonly used, assumption is the knowledge of one-hop neighbors (direct neighbors, or nodes located within transmission radius $R$), and possibly their locations. To collect such knowledge, a periodic "hello" message protocol is normally assumed, where each sensor transmits one message informing neighbors about its existence. Therefore, when message complexities are compared, we assume that one message per node is needed to acquire one-hop information. A further common assumption is of 2-hop neighbors, which are obtained after each node sends a message containing the list of its one-hop neighbors. We will assume therefore that collecting this information requires two messages per node. The actual cost could be higher, since such messages in dense networks could be long and energy-consuming to transmit.

### 11.1.3 Does Sleeping Always Conserve Energy?

The importance of placing as many sensor nodes as possible into sleep mode is apparent from the analysis of sensor energy expenditure. A sensor's radio can be in one of three active states—transmit, receive, idle—or in the sleep state. The radio is turned off in sleep state. The power consumption for various types of sensors and ad hoc nodes [8] shows that a sensor in the sleep state consumes 7–20 times less energy than one in the idle state. The power consumption while receiving a message is up to 10% higher than in idle state. Nodes spend 10–100% more energy while transmitting than while receiving messages. For instance, the Windows Internet Naming Service (WINS) seismic sensor consumes between 0.38 W and 0.7 W in the transmit state, 0.36 W in the receive state, 0.34 W in the idle state, and 0.02 W in the sleep state [9]. Sensors in the idle state are listening to the traffic and can be "alarmed" for any action. In the sleep state, however, they cannot receive any message and cannot be alarmed to become active. The importance of placing as many sensors as possible into the sleep state in order to prolong network life is apparent. Shall sensors sleep whenever they know that they are not needed for sensing or communication? Such an assumption is made in ref. [10], which proposed an activity scheduling scheme that assumes that sensor reporting can be done at predetermined times, along predetermined routes. In the route-discovery phase, each node learns about some neighbors and receives some forwarding tasks [10]. In addition, sampling, transmissions, and receptions along the route are also scheduled. This enables sensors to sleep between two scheduled tasks [10].

Suppose, for simplicity, that sensors are changing between active and sleep states on a regular basis, in rounds. The duration of a round cannot be made arbitrary, if prolonged network life is desirable. We will demonstrate this in the case of energy efficient behavior of a single sensor. Assume that there is fixed charge $C$ for transition between the active and sleep periods in sensor networks (this charge is not zero!). Assume also for simplicity that energy consumption in the active state remains consistent regardless of the amount of traffic handled (this, in fact, is not far from the reality [8]). Let $F$ be the ratio of energy consumption between the active and sleep states, and let $S$ be the energy consumed in the sleep state per unit of time (therefore, in the active state the consumption is $SF$ per unit time). Suppose that $T$ is the ratio of sensor reporting (active) and sleep times. If the sensor remains in active mode, its energy consumption is $LSF$ over $L$ time units. If the sensor decides to switch to sleep state between reporting, then there are $2L/(T+1)$ transitions, requiring energy of $2CL/(T+1)$. The consumption for sleep periods is $SLT/(T+1)$, and consumption during the active states is $SLF/(T+1)$. Thus overall consumption is $2CL/(T+1) + SLT/(T+1) + SLF/(T+1) = L[2C + ST + SF]/(T+1)$. This is compared to $LSF$ to conclude that switching to the sleep state between reporting periods is beneficial only if $T > 2C/(S(F-1))$. Therefore, the power needed for frequent transitions may outweigh the benefits obtained from sleeping.

Clustering can be effectively used to minimize internal transmission in a cluster with time division or frequency division. Energy awareness is not only a problem of sleep and awake but also a problem of collision avoidance.

## 11.2 BACKBONE CONSTRUCTION

### 11.2.1 Backbone Construction, Maintenance, and Analysis

Most broadcasting, activity scheduling, and sensor-area coverage algorithms rely on the concept of *backbone.* A backbone is a subset of sensor nodes that is able to perform data communication tasks and to serve nodes that are not in the backbone (because it is close to them). A backbone can also be the set of active sensor nodes, assuming then that the rest of the sensors are sleeping. There is a vast literature about backbone construction (see ref. [11] for a more comprehensive review). The primary backbone concept used in the literature is the dominating-set concept. A *dominating set* has the following property: each node is either in the dominating set or has a one-hop neighbor that is in the dominating set. Further, the connectivity property is often required for proper protocol functioning. A *connected dominating set* (CDS) is a dominating set of nodes that is also a connected set.

Figure 11.1 illustrates the CDS concept. Nodes 13, 12, 11, 10, 9, 4, 3, 1 are in CDS, and any remaining node is a neighbor of one listed. It is obvious that

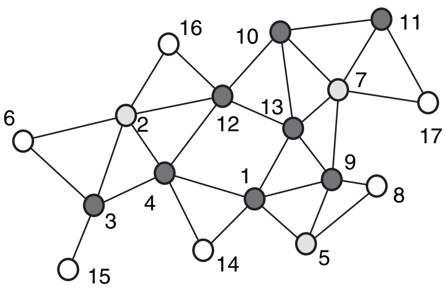


**Figure 11.1** A neighbor CDS consisting of nodes 13, 12, 11, 10, 9, 4, 3, 1.

broadcasting protocol, in which all nodes belonging to a CDS retransmit the message, will reach all nodes in a sensor network (assuming an ideal medium-access layer). This does not mean that all of them indeed need to retransmit, as will be discussed later in the chapter. An activity-scheduling scheme may simply direct all sensors in the backbone to be active, and allow all others to sleep. In this chapter, we have described the two most important backbone concepts: neighbor- and area-dominating sets. They can both be applied in the sensor networks. Once the sensors for area coverage are selected (area-dominating set), their backbone (neighbor-dominating set) can be constructed. Sensors in a connected neighbor-dominating set constructed over a connected area-dominating set can be used for broadcasting in the sensor network. Thus, one can be considered a backbone of another backbone. Note that area-dominating sets provide a network of medium density, which has an impact on the selection and performance of broadcast protocols for use in sensor networks (e.g., blind flooding may be an acceptable option).

The quality of a backbone construction/maintenance protocol is normally evaluated by backbone size with respect to the minimal possible size for the same network. The problem of constructing a CDS of minimal size (with a minimal number of sensors in it) is known to be NP-complete even for centralized algorithms. Therefore it is not surprising that finding good solutions by local means is a difficult task, and one that has attracted significant interest in recent years. The *approximation ratio* of a scheme is the ratio of the number of sensors in the constructed backbone over the minimal possible number of sensors in an optimal backbone. There are other metrics that can be considered [12]: the protocol duration, message overhead, and backbone robustness (does the backbone remains connected if one node fails?). For each metric, the evaluation can be performed, analytically or experimentally, using either *average-case* or *worst-case* performances. The ultimate goal is certainly to have a winner in both categories (such as mergesort or heapsort for the sorting problem). However, so far such a winner has not emerged, and researchers have normally adopted one of the two ways for comparison. Arguably, if a sensor network designer is presented with two protocols, one with excellent average-case, but occasionally quite bad performance, and the other with firm worst case guarantees (e.g., theoretically provable constant bound for the approximation ratio) but considerably inferior in the average case, we believe that the former would be the choice. This is the "philosophy" followed in this chapter.

Before describing some backbone schemes, we discuss their determinism and cost aspects. Backbone construction schemes can be classified as *probabilistic* and *deterministic*, based on whether or not a random number generator was used to construct them. The random number usage here is limited to the network-layer decisions; the underlying medium access scheme may still use random backoff counters, for example, in a deterministic protocol.

The backbone construction protocols described in the literature normally consider only *construction cost.* However, sensor networks are dynamic and the *maintenance cost* cannot be ignored; this is the cost to update the backbone when the network changes. Both construction and localized maintenance protocols can be further divided into *quasi-local* and *local* protocols. In a quasi-local (localized) protocol, all decisions are made based on local knowledge; however, the decisions made in one part of the network may have an impact on decisions made in a distant part of the network. Clustering is a typical example of a quasi-local protocol for both the construction and maintenance phases. The construction phase starts from a few selected "seed" nodes and propagates throughout the network. While this performance is debatable, the maintenance phase of quasi-local protocols is problematic, because of possible "chain effect": a simple change in an edge or addition/deletion of nodes may trigger global backbone updates by propagation. Otherwise, a local localized clustering procedure may have a negative impact on the quality (e.g., size) of the cluster structure. This chapter is therefore inclined toward local (localized) solutions, where, in both the construction and maintenance phases, the backbone status of each node depends solely on the local network configuration, typically one-hop or 2-hop (2-hop neighbors are one-hop neighbors of one-hop neighbors).

We will now describe some localized backbone construction methods and discuss them in the light of mentioned criteria and desirable properties.

### 11.2.2 Backbone Construction by Clustering

The distributed clustering algorithm [13] is initiated at all nodes whose id is lowest among all their neighbors (locally lowest id nodes). All nodes are initially undecided. If all neighbors of node *A*, which have a lower id, sent their cluster decisions and none declared itself a clusterhead (CH), node *A* decides to create its own cluster and broadcasts this decision and its id as a cluster id. If a node receives a message from a neighbor that announces itself as a CH, it will send a message (to all its neighbors) declaring itself a non-CH node, to enable more clusters to be created (note that two CHs are not direct neighbors in the algorithm). Thus each node broadcasts its clustering decision after all its neighbors with lower ids have already done so. Non-CH nodes that hear two or more CHs will declare themselves as *gateway* nodes. A sophisticated maintenance procedure for cluster formation when nodes move is described in ref. [13]. To minimize the number of clusters, ref. [14] proposed that node degree be applied as the primary key in clusterhead decisions. Nodes with more neighbors are then more likely to become a CH. In the case of ties, lower id nodes have an advantage. The clustering process requires one message per node in the construction phase (after one "hello" message to find the ids of neighbors or two "hello" messages to learn their degrees). Basagni [15] proposed variants of the clustering algorithm [13], which uses a variety of weights for selecting best CHs.

In the protocol described in ref. [16], after the clustering process is completed, each CH contacts neighboring CHs (up to three hops away) in order to eliminate some gateway nodes, and to use only essential gateway nodes to preserve overall connectivity. The construction and maintenance are fully localized. The protocol in ref. [16] produces an excellent approximation ratio, but the message overhead is significant, due to the overly high complexity of the election phase of the protocol leader, which requires information to be propagated to the fragment members and to nodes in adjacent fragments every time two fragments are merged into a new one. The simulation results [12] show that approaches with nice theoretical features, such as that presented in ref. [16], may hardly be applicable in practice due to the message complexity of their operations.

Basagni, Carosi, and Petrioli [17] described such a clustering based backbone scheme where nodes with more energy have higher chances to be clusterheads. Their construction and maintenance procedures are ongoing process with decisions based on received "hello" messages from neighbors. A node declares itself a CH if it did not receive a "hello" packet from a CH with energy that differs by more than certain threshold ("older" decisions have priority).

Chan and Perrig [5] described a localized clustering algorithm. New clusters are spawned in a self-elective process, when no messages from other CHs are received. Migration of an existing cluster is controlled by its CH. Each CH will periodically poll all its followers (neighbors) to determine which is the best candidate to become the new CH. The best candidate is the node that, if it were to become a CH, would have the greatest number of nodes as followers while minimizing the amount of overlap with existing clusters. The algorithm achieves a packing, efficiency close to hexagonal packing, but is quasi-local because chain effect is not prevented. It also has significant message overhead compared to other clustering protocols.

Wu and Dai [18] proposed a simple cluster formation in a dense network. First, the neighborhood detection is done using Hello messages with shorter transmission ranges than the normal one. The regular clustering algorithm is used to find CHs. However, CHs are directly connected using the normal transmission range. There are two versions of this approach. In the first version, the range of the Hello message is $\frac{1}{3r}$, where $r$ is the normal range. In this way, all CHs within three hops are connected, and CHs are globally connected. In the second version, the range of Hello message is $\frac{1}{4r}$. During the transmission using the normal range $r$, $\frac{3}{4r}$ is used to connect all CHs within three hops and $\frac{1}{4r}$ is used to cover the member in the cluster with a radius of $\frac{1}{4r}$.

In the protocol by Kuhn, Moscibroda, and Wattenhoffer [19], sensors may wake up asynchronously at any time and do not have collision detection capabilities. They only know the limit on the total number of sensors, and have no knowledge of possible neighbors. The algorithm computes asymptotically optimal clustering. The main idea is that nodes, after some initial waiting, compete to become dominators by exponentially increasing their sending probability on one channel. Two other channels are then used to guarantee that the number of further dominators emerging in the neighborhood of an already existing dominator remains small. The algorithm can be simulated to work by using only one channel.

### 11.2.3 Backbone Construction by Nominating and Grid Partitioning

This section will describe two very simple schemes for backbone construction. In ref. [20], the authors propose a simple method for determining the dominant set (not necessarily connected). Each node nominates the neighbor with the largest id among its neighbors to be in the dominant set (assuming that each node has a unique identifier). This can produce the $O(n)$ approximation ratio in the worst case, but works well in the average case. An example of bad performance is a linear chain of nodes with increasing identifiers. Each node needs one message to learn the identifiers of neighbors, and possibly the second message to nominate a neighbor into the dominant set. Connecting the dominant nodes is, unfortunately, a nontrivial problem (e.g., the protocol in ref. [16] could be used for it).

Xu, Heidemann, and Estrin [21] discuss the following backbone construction scheme called GAF. The given two-dimensional space is partitioned into a set of squares (called *cells*), such that any node within a square can directly communicate with any node in an adjacent square. Therefore, one representative node from each cell is sufficient for a connected backbone. Each node transmits its id (which may depend on its remaining energy) plus its coordinates (this requires one message per node). In each cell, the node with maximal id is selected for the backbone. The selected nodes in ref. [21] make a dominant set, but its average size (which depends on the selected size of the square) may be higher than for other methods considered here. Further, the dominant-set concept needs some parameters, such as the size and position of squares, which have to be propagated in the network. The method is simple, has no chain effect, and has a constant approximation ratio. When crossing a boundary, nodes need to retransmit their information to maintain the dominant set. When crossing the second boundary in a larger movement, this is not sufficient, as the moving node has no information about nodes in the new cell. This can be resolved by triggering a round of Hello messages in that cell. The most significant problem is that, for any ratio of transmission radius and grid size, the dominant set obtained may disconnect the graph [17]. Although the network topology is connected, Basagni, Carosi, and Petrioli [17] observed that, for instance, on a network with 50 nodes, GAF [21] get disconnected $\geq 40\%$ of simulation time for any grid size that produces a meaningful backbone size. An example illustrating that the partition may occur even for range transmission radii with respect to grid size is given in Figure 11.2.

### 11.2.4 MPR-Based Backbone

Several broadcasting schemes are based on the concept of multipoint relays (MPR) of a node $S$, defined as a minimal-size subset of neighbors of a given node $S$ that will

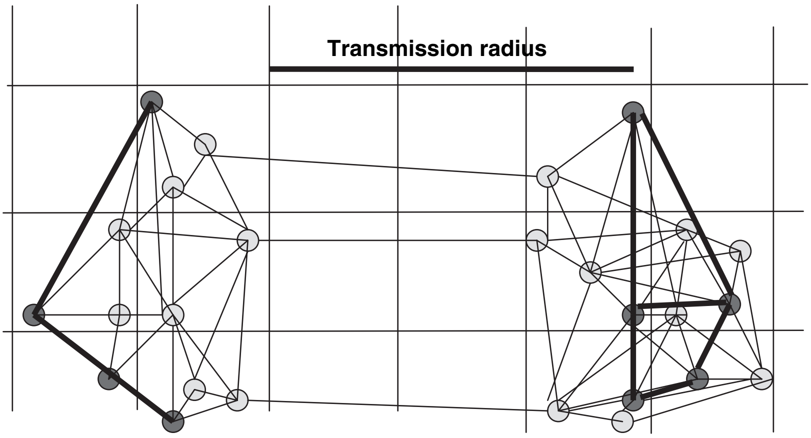


**Figure 11.2** Leaders in a grid partitioning may be disconnected.

"cover" all 2-hop neighbors of *S*. A node is called *covered* if it can receive (directly or via retransmissions by relay nodes of *S*) messages originating at *S*. Relay points of *S* are one-hop neighbors of *S* that cover all 2-hop neighbors of *S*. The goal is to minimize the number of relay points of *S*. The computation of an MPR set with minimal size is an NP-complete problem. A heuristic algorithm, called a greedy set cover algorithm, is proposed in ref. [22]. This algorithm repeats selecting node *B*, which maximizes the number of neighbor nodes that are not yet covered.

Adjih, Jacquet, and Viennot [23] proposed to combine MPR and dominant-set approaches. Each node computes its set to be forwarded to its neighbors and transmits it to its neighbors. It then determines whether it belongs to the "MPR-dominating set" if it either has the smallest id in its neighborhood, or the node is a forwarding (relay) neighbor of the neighbor with the smallest id. Wu [24] observed that a node with a smaller id than all its neighbors, but without two unconnected neighbors, can be eliminated. The construction of an MPR-based backbone requires 2-hop neighbor knowledge, plus a message containing the list of relay nodes of each node. This can be treated overall as CDS construction requiring three rounds of messages, plus another round if the CDS decisions are to be communicated to neighbors.

### 11.2.5 Wu's Backbone

In a series of articles (the first one being ref. [25]), Wu et al. described, a lightweight backbone construction scheme. We will use a modified definition from refs. [14] and the [26] of basic concept [25], because of its reduced message overhead. A node is an *intermediate* node if it has two unconnected neighbors [25]. In the example in Figure 11.1, nodes *C* and *K* are the only nodes that are not intermediate. A node *A* is covered by a neighboring node *B* if each neighbor of *A* is also a neighbor of *B*, and key(*A*) $<$ key(*B*). Assuming that the keys in Figure 11.1 are ordered alphabetically, node $H$ is covered by node $I$, $G$ is covered by $L$, while $A$ and $B$ are covered by $E$. Nodes not covered by any neighbor are *intergateway* nodes. A node $A$ is covered by two connected neighboring nodes $B$ and $C$ if each neighbor of $A$ is also a neighbor of either $B$ or $C$ (or both), key($A$) $<$ key($B$), and key($A$) $<$ key($C$). An intermediate node not covered by any neighbor becomes an intergateway node. An intergateway node not covered by any pair of connected neighboring nodes becomes a *gateway* node.

Dai and Wu [27] introduced a generalized dominant set, where coverage can be provided by an arbitrary number of connected one-hop neighbors (instead of 1 or 2 as in the original definitions). The definition was modified in ref. [11] to the following form to avoid similar message exchanges between neighbors. Node $A$ is covered by its one-hop neighbors $B, C, D, \ldots$ if the neighbors $B, C, D, \ldots$ are connected, any neighbor of $A$ is a neighbor of at least one of nodes $B, C, D, \ldots$, and key($A$) $<$ min (key($B$), key($C$), key($D$), . . .). It is then further computationally simplified by Carle and Symplot-Ryl [28], as follows. First, each node checks if it is an intermediate node. Then each intermediate node $A$ constructs a subgraph $G$ of its neighbors with higher key values. If $G$ is empty or disconnected, then $A$ is in the dominating set. If $G$ is connected, but there exists a neighbor of $A$ that is not a neighbor of any node from $G$, then $A$ is in the dominant set. Otherwise $A$ is covered and is not in the dominant set. Dijkstra's shortest-path scheme can be used to test the connectivity. This procedure is generalized since it allows coverage by any number of neighbors. It is computationally even less expensive than the two-nodes coverage case.

The CDS concept [25,27] is illustrated in Figure 11.1, where the keys are assumed to be ordered by their numerical id values: "1" $<$ "2" $< \cdots <$ "16." Nodes 6, 15, 16, 17, 14, and 8 do not have two unconnected neighbors that are not in CDS (they are not intermediate). Node 2 is dominated by three connected neighbors (nodes 3, 4, 12), since they have higher key values, and the remaining neighbors 6 and 16 are "covered" by 3 and 12, respectively. Node 7 is covered by four connected neighbors with higher keys 9, 13, 10, and 11 (the remaining neighbor 17 is covered by 11). Node 5 is covered by its neighbor 9, since other neighbors (1 and 8) are neighbors of 9, and "5" $<$ "9." Node 1 remains in CDS because neighbors with higher keys (4, 14, 5, 9, 13) are disconnected.

Wu's concepts require either one-hop knowledge of neighbors with their position, or 2-hop neighbor topology information. This can be obtained after one or two Hello messages from each node. Experimental data from several sources (e.g., ref. [12]) confirm that Wu's concepts provide small-size CDS on average. It was proved in ref. [27] that the generalized CDS concept has a constant approximation ratio on average, and very low probability of having an infinitely large approximation ratio. An example of a "bad" approximation ratio is the case of a linear chain of nodes with increasing keys, where almost all nodes are selected into the CDS.

Each node makes decisions about CDS membership (in Wu's concept) without communications between nodes beyond the message exchanges that nodes use to discover each other and establish neighborhood information. If knowledge of neighbors that are in the CDS is needed, then one message from these nodes suffices.

In that case, such a message can be used to further reduce the size of the backbone. As soon as one node decides to be in the CDS, it sends a packet informing neighbors about the decision. Neighbors (which did not yet decide their membership) will then consider such decided CDS neighbors as having higher key values, which may help them in withdrawal from the CDS decision [28].

### 11.2.6 Enhanced Dominating Sets

The number of nodes in a CDS following Wu's concepts can be reduced by applying some enhanced concepts [29,30]. The first observation [29] is that if 2-hop topological knowledge is already required, it can be used to eliminate a few more nodes from the CDS. Consider the example in Figure 11.1. Node $D$ is in the CDS although it is actually covered by nodes $E$, $I$, and $L$. The later three nodes all have higher key values, and are connected. Node $L$, however, is not a one-hop neighbor of $D$. This does not prevent node $D$ from verifying whether any of its neighbors are neighbors of $L$, or whether $E$, $I$, and $L$ are connected, since $L$ appears in the list of neighbors sent to $D$ by its one-hop neighbors; therefore, such a conclusion can be made. The new definition therefore can be given as follows [29]. Node $A$ is covered by its 2-hop neighbors $B$, $C$, $D, \ldots$ if the neighbors $B$, $C$, $D$, $\ldots$ are connected (according to information available to $A$), any neighbor of $A$ is the neighbor of at least one of nodes $B$, $C$, $D, \ldots$ and $\text{key}(A) < \min\ (\text{key}(B), \text{key}(C), \text{key}(D), \ldots)$. Note that $A$ is not aware of possible links between its two 2-hop neighbors, and therefore may declare the set disconnected although in reality it may be connected. Note that Rule $k$ in ref. [27] is general, allowing coverage by a set of one-hop and 2-hop "marked" neighbors that are "glued" together by other "marked" nodes ("marked" nodes are those that consider themselves to be in the dominant set), which can be at an arbitrary hop distance. However, in algorithm 2 from ref. [27], implementing Rule $k$, nodes send their dominating status only to their one-hop neighbors; therefore, the information about the dominating status of 2-hop neighbors and beyond is not made available for use in making a decision. While implementation [27] is based on sending messages from each node (informing about withdrawal from the dominant set), the algorithm set forth in ref. [28] does not use any message between nodes after gaining 2-hop topological knowledge. Further, the observation described in ref. [29] is based on coverage by nodes that may or may not be in the dominant set, while the definition given in ref. [27] refers to only coverage by nodes that are in the dominant set.

The second observation [29] is that key values often present obstacles to selecting proper nodes in the CDS. A definition that will allow key reversal may be beneficial. Suppose that, in Figure 11.1 node $G$ was actually renamed node $M$ for a reason (e.g., high energy value). Then node $M = G$ will be in the CDS, because of the highest-key value. But this does not eliminate any other node from the CDS; therefore, its inclusion is superfluous. How then can the key of $M = G$ be reversed? All the neighbors of $M = G$ are neighbors of $L$, and $L$ has a neighbor that is not a neighbor of $M = G$. This is sufficient for node $M = G$ to realize that $L$ will not declare it as a covering node, and therefore can safely withdraw from CDS. This concept can be formalized as follows. Node $u$ is covered by node $v$ if and only if one of the following two conditions is satisfied:

(1) $N(u) \subset N(v)$, where $N(u)$ is a proper subset of $N(v)$, that is $N(v) \neq N(u)$ is part of this condition), and
(2) $N(u) = N(v)$ and key($u$) < key($v$).

Note that the preceding extended rule cannot be used jointly with other rules, such as Wu and Li's Rule 2 [25]. The generalization to coverage by several nodes and the corresponding algorithms for backbone construction are presented in ref. [29].

The two enhancements described can be combined into a single one, by allowing the node to be covered by either of the two ways [28].

### 11.2.7 Activity Scheduling in Ad Hoc Networks

In an ad hoc network that is not a sensor network, area coverage may not be required. In such a case, *activity scheduling* (deciding which nodes should be active, and which should go to sleep mode, so that the ad hoc network life is prolonged) can be performed by applying the connected-neighbor dominant set concept. Nodes in the connected-neighbor dominant set are active, while the rest of the nodes can be put to sleep. However, in order to increase network lifetime, such decisions need to be periodically reevaluated, as nodes that are saving energy need to contribute at a later time. Each node in an asynchronous ad hoc network may wake up at its predetermined time and evaluate whether it needs to be active based on a message exchange with currently active neighbors. In the case of synchronous nodes, such decisions are made in rounds. All nodes wake up at the same time, exchange Hello messages, and then decide which of them will create a backbone. Any described backbone decision process can be applied. If Wu's concept is applied, a suitably selected key value, which depends on the remaining node energy, is selected and used. The other important parameter for making decisions is the average number of neighbors (average degree) of each node. The choice of such a best metric for prolonged network life was investigated in ref. [31].

## 11.3 BROADCASTING IN SENSOR NETWORKS

### 11.3.1 Taxonomy

In addition to the taxonomy discussed for the backbone construction, the broadcasting protocols can be further classified. The next division is whether or not they are *reliable*. Reliability is the ability of a broadcast protocol to reach all the nodes in the network, assuming that the medium-access control (MAC) layer is ideal (every message sent by a node reaches all its neighbors), location update protocol provides accurate desired information to all nodes about their neighborhood, and the network is connected. The *blind-flooding* protocol, where each node receiving the packet for the first time will retransmit it, is a reliable protocol at the network layer. However, as observed in seminal work [32], due to excessive retransmissions for dense networks, collisions and contentions actually can make it very unreliable at the MAC layer, plus there exists a large amount of redundancy. The probabilistic (retransmissions with certain fixed probability), counter (retransmitting if the number of received copies does not exceed a constant), and distance (retransmitting if the distance to all senders exceeds certain threshold distance) solutions proposed in ref. [32] are not reliable at the network layer, and also have inferior rebroadcast savings (percentage of nodes that do not retransmit the packet) to the backbone-based reliable solutions reviewed here. Note that the MAC layer cannot be reliable (at least those currently considered for adoption in sensor networks), due to the hidden-terminal problem (a node simultaneously receiving messages from two other nodes that are not aware of each other's transmission) and the probabilistic nature of the protocols used.

The final classification of broadcasting schemes is determined according to the *packet content* during the broadcasting process. The broadcast message sent by the source, or retransmitted, might contain a broadcast message only. In addition, it may contain a variety of information needed for proper functioning of the broadcast protocol, such as its own id, its position, one bit about its backbone status, a list of one-hop neighbors, degree (number of its neighbors), or list of forwarding neighbors, informing them whether or not to retransmit the message.

### 11.3.2 Backbone and Neighbor Elimination–Based Broadcasting

In ref. [14], the following framework and general algorithm were established for reliable broadcasting. The algorithm is based on two concepts: CDS as the particular type of backbone that provides reliability, and neighbor-elimination scheme. Backbone formation was already discussed in Section 11.2. Connectivity provides propagation through the whole network, while domination assures reachability by all nodes. Excess messages in any protocol affect node power and the bandwidth available; thus, the main goal is to describe a reliable broadcast protocol with a minimal number of retransmissions, that is, to construct a connected dominating set of minimal size.

The *neighbor-elimination* scheme was independently proposed in three papers in August 2000 [33–35]. In this scheme, a node does not need to rebroadcast a message if all its neighbors have been covered by previous transmissions. After each copy of the same message is received, a node eliminates, from its rebroadcast list, neighbors that are assumed to have correctly received the same message (based on one-hop positional or 2-hop topological knowledge that the node has about its neighbors). If the list becomes empty before the node decides to rebroadcast, the rebroadcasting is canceled.

The general algorithm [14] for intelligent flooding is then the following one. The source node transmits the packet. Upon receiving the first copy of the transmitted packet intended for broadcasting, the node will not retransmit it if it is not in the CDS. If it is in the CDS, it will select a time-out based on some criteria and some random number. It will also eliminate all neighbors that received the same copy of the message from its forwarding list (originally containing all one-hop neighbors).

While waiting, more copies of the packet could be received. For each of them, all neighbors receiving it are eliminated from the forwarding list. When timeout expires, the node will retransmit if its forwarding list is non-empty, otherwise it will cancel retransmission. This framework was applied in ref. [14] using clustering based and Wu's concept based backbones. Wu and Dai [36] propose a general algorithm that unifies many neighbor elimination schemes.

Figure 11.3 illustrates the broadcasting algorithm [14], with *C* being the source node, and nodes *F*, *A*, *G*, and *H* being in the connected dominating set following definition [25,27], with key = (degree, id). Node *E* is covered by node *G*, while node *L* is covered by connected neighbors with higher keys *G* and *H*. Similarly, node *I* is covered by *A* and *H*. Other nodes are not intermediate (do not have two unconnected neighbors). Covering relations are drawn in the dashed bolder edges. Let the time-out be defined as time-out = (1/(number of uncovered neighbors), id). Note that id is added to decide which node retransmits first in case of ties. Node *F* then sets the time-out to $\frac{1}{3}$ (three uncovered neighbors by source transmission are *A*, *G*, *E*) and retransmits at the time-out expiration. Neighbors from CDS are *A* and *G*, and they set time-outs to $\frac{1}{3}$ and $\frac{1}{2}$, respectively, based on the number of neighbors not receiving that transmission (based on their knowledge; it is possible that some neighbors treated as uncovered actually already received the message from nonneighbors in the process). Node *A* then retransmits because of shorter timeout. After this retransmission, *G* changes the original time-out to $\frac{1}{1}$ (only neighbor *L* remains uncovered), and the remaining time-out is $\frac{1}{1} - \frac{1}{3}$, since $\frac{1}{3}$ of the time already lapsed. The time-out at *H* is $\frac{1}{3}$ and is shorter, so it retransmits first. Node *G* then cancels retransmission.

To increase reliability at the MAC layer, Stojmenovic et al. [14] proposed the retransmissions after negative acknowledgments (RANA) protocol. Collision

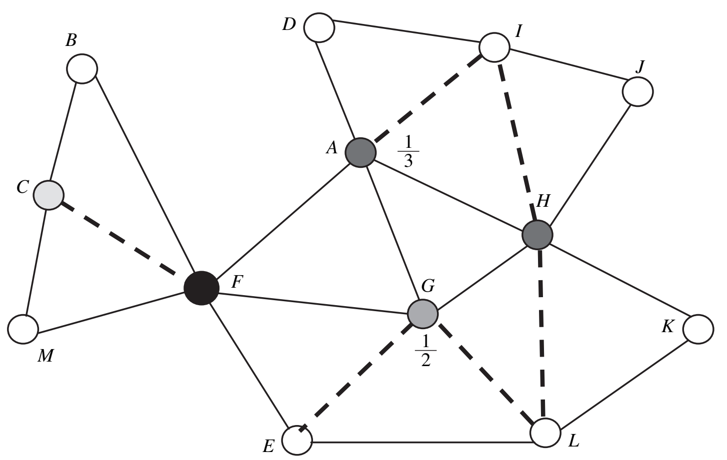


**Figure 11.3** *F*, *A*, and *H* retransmit in the neighbor elimination and dominating set–based broadcasting [14].

between two packets normally occurs after the initial portion of the first packet, containing the sender's information has already been received. The receiver node can then send a negative acknowledgment back to the sender node, asking it to retransmit again.

### 11.3.3 MPR

Several authors [33,37–39] independently proposed reliable broadcasting schemes in which the sending node selects adjacent nodes that should relay the packet to complete the broadcast. The ids of the selected adjacent nodes are recorded in the packet as a forward list. An adjacent node that is requested to relay the packet again determines the forward list. This process is iterated until the broadcast is completed. The methods differ in the details on how a node determines its forward list. The general principle was already outlined in the section on MPR-based backbone.

The adaptation of multihop relaying presented in ref. [40] improves its performance by the following observations: the broadcasting node transmits a list of its neighbors at the time of broadcast packet transmission, not as part of any Hello message. Knowledge of the 2-hop neighbors is used to determine which neighbors also received the broadcast packet in the same transmission, and these nodes are already covered and are removed from the neighbor graph used to choose the next hop relaying nodes. Finally, if a broadcast message is received from a node that is not listed as a neighbor, the message is retransmitted to deal with high mobility issues. In connected dominant set–based broadcast algorithm [41], the sender node establishes priorities between the forwarding nodes and each forwarding node should eliminate from consideration not only neighbors of the sender node, but also neighbors of each relaying node with higher priority. Wu and Lou [43] proved several extensions of MPR to generate a smaller CDS using 2-hop neighborhood topology information to cover each node's 2-hop neighbor set. Note that 2-hop neighborhood topology includes all nodes within two hops and their connections. In addition, they extended the notion of coverage in the original MPR and showed that the extended MPR has a constant local approximation ratio compared with a logarithmic local ratio in the original MPR.

Compared to backbone-based broadcasting, MPR broadcasting has a similar or somewhat better performance in terms of rebroadcast savings, but has message overhead due to the inclusion of the forwarding list in the packet, which may be significant for energy-limited tiny sensors.

### 11.3.4 Broadcasting and Dominating Sets with Realistic Physical Layers

We now describe the corresponding coverage, backbone notions, and broadcasting process when the impact of the physical layer is considered. Let $A_1, \ldots, A_k$ be active neighbors of given node $B$, and let $x_1, \ldots, x_k$ be their respective distances to $B$. Then $p(x_1), \ldots, p(x_k)$ are their packet reception probability rates for packets sent by $B$.

The probability $q(x)$ that at least one of the packets from the active nodes is received by $B$ is then $q = 1 - (1 - p(x_1))(1 - p(x_2)) \cdots (1 - p(x_k))$. Node $B$ is $m$-covered by active nodes $A_1, \ldots, A_k$ if $q \geq m$ [3]. A set of nodes is the $m$-dominating set if each node is either in the set or is $m$-covered by nodes from the set [3]. Note that, for $m = 1$, and the unit-disk graph model, the well-known definition of dominant sets follows.

These definitions can be used as a basis to generalize some well-known types of dominating sets for the unit-disk graph to be applied under a realistic physical layer. For example, the following definition is proposed in ref. [3] as a generalization of the concept proposed by ref. [27]. Let $A_1, \ldots, A_k$ be the set of higher id neighbors of $B$. If the set is empty or disconnected, then $B$ is in the dominating set. If the set is connected and each neighbor of $B$ is $m$-covered by them, then $B$ is not in the dominating set. Finally, if any neighbor of $B$ is not $m$-covered by the set, then B is in the dominating set.

The broadcasting process with any notion of dominating sets and neighbor elimination [14] can proceed as follows [3]. After receiving a broadcast message, node $A$ will set a time-out short if it is in the dominant set, and long if not. It calculates the probabilities of each neighbor for receiving the same message, and eliminates $m$-covered neighbors from the list. This lists is updated for any further copy of the received message. The update includes the time-out that can be extended with more received messages. At the end of time-out, if all neighbors are $m$-covered, retransmission is canceled. Otherwise, the node retransmits the packet.

Since the reception of any message is a probabilistic event, one retransmission by any particular node may not suffice. To learn about the existence of neighbors, each node may need to send several packets. The number of retransmissions needed for learning about the satisfactory number of neighbors depends on density. In ref. [3], it was proposed that each node retransmit Hello messages until a certain fixed number of such packets or responses is received from neighbors, as an indirect measure of density. A similar protocol also can be applied for the broadcasting task, modifying any existing protocol originally designed for the unit graph as follows. Instead of retransmitting only once, a given node can keep retransmitting until a certain fixed number of packets (carrying the same packet) has been heard from neighbors, before or after the first retransmission, or until a certain time-out expires (to handle the case of low-degree nodes). If density is known, then a fixed number of retransmissions can be replaced by a number depending on local density. Nodes that are, by original protocol, supposed not to retransmit may also contribute by retransmitting the message, but fewer times than other nodes. Further investigation and simulation is needed to find a precise description of the winning protocols, following this general design principle.

### 11.3.5 Minimum Energy Broadcasting

Suppose that nodes in an ad hoc network can adjust their transmission radii, and that they are aware of their own and the geographic position of their neighbors. The problem is to broadcast a packet to all the nodes in the network so that the sum of all transmission power used is minimized. The power consumption for two nodes at distance $r$ is $r^{\alpha} + c$, where $\alpha \geq 2$ and $c$ is a constant that includes signal processing and minimal reception power. It is shown in ref. [43] that, for $c > 0$ (which is a realistic assumption), it is not optimal to minimize transmission range. Furthermore, it was demonstrated that there exists an optimal radius, computed with a hexagonal tiling of the network area, that minimizes the power consumption. For $\alpha > 2$ and $c > 0$, the optimal radius is $r = (2c/(\alpha - 2))^{1/\alpha}$, which is derived theoretically and confirmed experimentally.

A localized broadcast algorithm, called *TR-LBOP* is proposed [43], which takes this optimal radius into account. This protocol is experimentally shown to be have limited energy overhead with respect to globalized algorithms for all network densities.

## 11.4 SENSOR AREA COVERAGE

In area-coverage problems, a set of sensors is given and distributed over a given area. Each sensor is able to cover a circle with radius centered at it. The problem is to determine a small number of sensors that still cover the same area and are connected, so that the sensor can report the detected information to a monitoring center. The maximum network lifetime is certainly a related goal. *Full coverage*, *energy efficiency*, and *connectivity* are critical requirements of any area-coverage protocol. The objective of any area-coverage protocol is to achieve full area coverage, and protocols can be classified into those that guarantee full area coverage (provided such coverage exists) and those that do not guarantee it. A set of sensor nodes that together fully cover a given area is called *area-dominating set.* Protocols can also be divided into those that guarantee connectivity of selected active sensors and those that do not.

There is a variety of problem statements, assumptions, and solution approaches for the sensor area coverage. We will review them before presenting some solutions. The problem is centered on a fundamental question: How well do the sensors observe the physical space? This chapter discusses only the *area-coverage* problem, meaning that each point in a given geographic area needs to be covered by at least one sensor. Alternative formulations include covering certain points instead of area (*point coverage*) and *barrier coverage.* Examples of barrier-coverage problems are, minimizing the probability of undetected penetration through the sensor barrier and minimal exposure path, measured as sensing time, with sensing ability diminishing with distance. A survey of point- and barrier-coverage solutions is given in ref. [44].

The area-coverage problem can be further divided into *single* and *multiple* area coverage. In single area coverage, each point in the area is required to be covered at least by one sensor. In multiple area coverage, each point needs to be covered multiple times, which could be a fixed $k$ times coverage at a given time, or division of sensors into maximum number of layers of area coverage. These layers can then either alternate in time for coverage, or several layers can be used to cover an area simultaneously for increased reliability.

The sensor deployment mechanism can be *random* or *deterministic.* A deterministic sensor placement (placing sensors at desired locations) may be feasible in friendly and accessible environments. Random sensor distribution is generally considered in remote or inhospitable areas, or when a fast deployment of a large amount of sensors is desirable. We will consider only random placement in this chapter. An example of deterministic placement is considered in ref. [45], where the authors proposed path exposure (the likelihood of detecting a target traversing the region using a given path) as the measure of goodness of the sensors that are deployed to perform collaborative target detection. A centralized algorithm for placing sensors at selected locations to minimize path exposure is described in ref. [45].

In most articles in the literature, all sensing radii are equal, while a few articles consider coverage with different sensing radii. We will consider only the case of equal sensing radii at each node, since there is limited research done for the case of adjustable ranges [46,47]. Wu and Yang [47] considered the cases where each sensor is able to select one of two or three adjustable ranges, with the goal of minimizing the overlapped sensing area, extending results from ref. [48].

There are also several variants regarding the relation between sensing and transmission ranges. One common assumption is that sensing radius and communication radius are *equal* [28]. However, some physical measurements indicate that the communication range is normally *larger* than the reliable sensing range. This has implications on the selection of sensors for coverage, and also on the performance of other relevant protocols. For example, Xing, Lu, Pless, and Huang [49] show that greedy routing always works when the communication range is twice or more the sensing range, and the area is covered and convex. They also consider restricting greedy routing to nodes whose Voronoi regions intersect the source–destination line.

Most literature uses the *unit-disk graph* model for sensing, which is similar to the unit-disk graph model used for communication. In this model, the sensor is able to monitor an event if and only if the distance from the sensor to the event is at most $S$, where $S$ is its sensing radius. However, a closer look at the *physical layer* reveals that sensing ability decreases with distance. Instead of the unit-disk graph model, it is more realistic to use a model where the probability of sensing an event depends on the distance from the sensor to that event. Liu and Towsley [50] approached the coverage problem from a theoretical perspective and explored the fundamental limits of the coverage of a large-scale sensor network.

Zhang and Hou [51] studied the fundamental limits of sensor network lifetime that all algorithms can possibly achieve. If the lifetime of a sensor is $T$, they derived analytically and by simulation the minimum sensor density needed to achieve network lifetime $kT$. They observed that the increase in lifetime per unit of nodal density becomes marginal when the density exceeds a certain threshold.

### 11.4.1 Threshold-Based Protocols

Ye et al. [52] proposed a simple localized protocol (called PEAS) for dynamically selecting an area-dominating set in asynchronous sensor networks. Each sensor has the same probing radius $P$ and the same maximum transmission radius $R$, which is also the monitoring radius. Any two active sensors must be at a distance of at least $P$, which is enforced by the scheme. Initially all sensors are in sleeping mode, with an exponentially distributed sleep-duration function. When sleeping time expires, the sensor sends a probing message using transmission radius $P$. Each active sensor that overhears this probing message should estimate whether or not its distance to the probing sensor is below $P$. Since they are able to detect signals from a greater distance, up to $R$, they should apply signal strength (which is considered an unreliable measurement due to fading effect) or time-delay measurements to make the judgment. If the distance is below $P$, then it sends (a/the) message to the probing sensor informing it about its activity. Upon receiving such a response, the probing sensor again selects a new sleeping duration and continues to sleep, waking up at a later predetermined time to reevaluate the decision. If the distance is above $P$, then no response is generated. If the sensor does not receive any response to its probing signal, it decides to wake up and monitor the area, up to radius $R$. Once a sensor wakes up, it continues to work until it dies. For this protocol, the probability of having full coverage of a monitored area is close to 1 if the threshold $P$ is less than $1/(1+\sqrt{5}) \approx 0.3$ of the sensing area's radius $S$, that is, $P < 0.3S$. The rationale is that otherwise activating the sensor has an insufficient contribution toward covering some new area, due to it being too close to an already active sensor. The method presented has a high degree of fault tolerance. However, this protocol is probabilistic and does not ensure full area coverage. Figure 11.4 illustrates this protocol, with the black nodes being active and the white nodes being in sleep mode, because each of them is contained within the threshold distance (smaller circles) to one of the active nodes. The larger circles indicate the communication radius for active nodes.

In ref. [53], three sensor-area covering schemes are proposed. In the probabilistic-based scheme, each node decides whether or not to remain active with a fixed probability, whose optimal value is derived based on the expected percentage of the sensing area coverage, which in turn depends on the number of neighbors

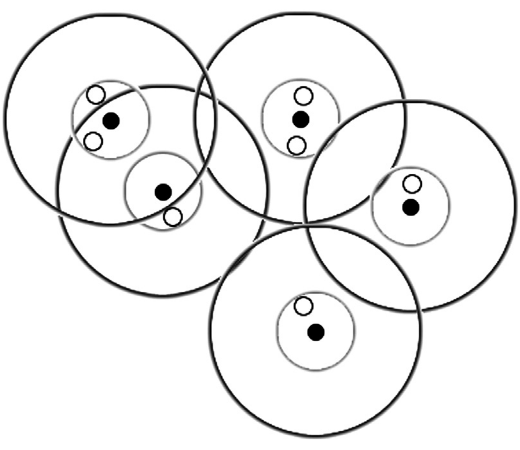

**Figure 11.4** Threshold-based area coverage.

within transmission radius that announced active status and expected distance to them. In the nearest neighbor–based scheme, the decision is based on the expected distance (whose value is derived as a function of the ratio of the transmitting and sensing radii and the number of active neighbors) to the nearest of the active neighbors (this is similar to the PEAS scheme [52]). In the neighbor number–based scheme, the decision is based on a counterthreshold compared to the number of active nodes. All mentioned threshold values are determined numerically and experimentally, for use in the schemes, and do not guarantee area coverage.

### 11.4.2 Some Covering and Connectivity Properties

In refs. [48] and [54], it is proved that if the transmission range is at least twice the sensing range, and the area to be covered is convex, then the area coverage also implies connectivity among the covering sensors. This follows from observing that the distance between the centers of two intersecting circles of the same radius cannot exceed twice the radius, therefore two sensors whose sensing radii intersect are also communication neighbors. The distance between two nodes whose sensing ranges $S$ intersect is $<2S$, which is within the transmission range $R$ for $R > 2S$. Di Tian [55] generalized this proof by eliminating the need for the convexity condition.

When the sensing and transmission radii are equal, the coverage property can be tested by verifying whether or not the perimeter of the sensing circle is covered by other circles. This is illustrated in Figure 11.5. The number of uncovered arcs of a circle can be at most two.

When the communication range exceeds the sensing range, this simple test cannot be used. Finding the exact regions of intersection, or their size, is computationally sophisticated and time-consuming. However, one can apply the following well-known geometric theorem [48,54] to efficiently confirm that a sensing circle is fully covered by other sensing circles: It is shown that if there are at least two covering circles and any intersection point of two covering circles inside the sensing area is covered by a third covering circle, then the sensing area is fully covered.

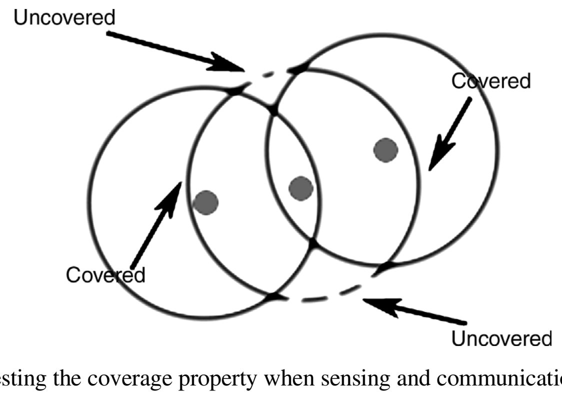


**Figure 11.5** Testing the coverage property when sensing and communication radii are equal.

This preceding result is illustrated in Figure 11.6. Sensors within a communication edge of each other are joined by an edge. The central darker sensing circle is covered by other circles, and all the intersection points of other circles, which are inside it, are covered by a third circle.

This result provides an efficient method for testing the full-coverage criterion. However, it does not provide direct information about the possible size of the uncovered region. One possible estimate is to generate a certain number of points at random, test each for coverage with existing circles, and take the percentage of covered points as the estimate (this method can be computationally expensive if satisfactory precision is required). Another alternative is to make an estimate based on the distances and positions of the active sensors. There exists a need for designing more accurate and fast-coverage size-estimation protocols, including an efficient test for confirming full coverage.

### 11.4.3 Hexagonal Area Coverage

Zhang and Hou [47] described an efficient algorithm for selecting covering sensors. Sensors are assumed to be time synchronized, and they periodically make new decisions about sensors that remain active to cover the area. In each round, a single sensor starts the decision process, which then propagates to the whole network. New sensors are selected so that the priority is given to sensors located near optimal hexagonal area coverage, obtained when the area is ideally divided into equal regular hexagons. The coverage is indeed very optimal, given the *distributed* nature of the decisions. However, the need for a single sensor to start the process may cause problems in applying it. Since time is synchronized, and rounds are well defined, perhaps it is better to allow all sensors to make localized decisions without waiting for any specific sensor to start the process (especially if

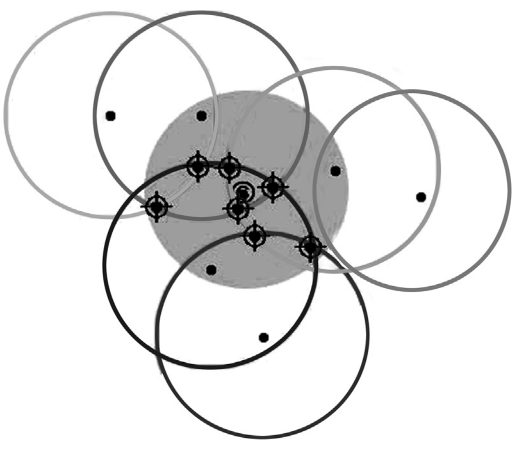

**Figure 11.6** A circle is covered when all intersection points are covered.

the sensor somehow decided to start the process and failed to do so because of malfunctioning). The original sensing coverage may not be preserved (as shown by experimental results).

### 11.4.4 Area Coverage Based on Neighbor Cooperation

The algorithm presented in ref. [56] divides the area into small grids, and then covers each grid with a sensor. Each sensor that can cover a grid maintains a list of other sensors that can also cover it, in a priority order. All sensors covering the same grid can communicate with each other, since the communication range is at least twice the sensing range. When sensor density is significant, sensors need a lot of memory and processing time to maintain priority lists, plus the communication overhead for making covering decisions in cooperative manner is nontrivial.

Hsin and Liu [57] investigated random and coordinated area-coverage algorithms. Each sensor covers a circle of radius *R*. In their coordinated-coverage scheme, a sensor may decide to sleep after receiving "permission" from sponsoring neighbors, for the time such permission is given. A node that sponsors any other node must be active. The decisions are not synchronized, since each sensor can "negotiate" with its sponsors independently, and the scheme allows for several variants with (sophisticated) protocol details. The authors suggest that nodes collect information about residual energy from neighboring sensors. Sensors with high residual energy are more likely to enter the sleep state than sensors with low residual energy. Each sensor maintains its own delay counter, which is used for role alteration. Coordinated schemes performed better in their experiments. Although the Hsin and Liu's [57] coordinated scheme has some desirable properties, such as localized behavior, it may select too many sponsor nodes to be active, since there is no coordination between nodes for the selection of as many as possible common sponsor nodes.

### 11.4.5 Centralized Area-Coverage Protocols

Centralized (and distributed) schemes may be treated as localized schemes with extended communication range, where any node can reach any other node. In this scenario, obviously one node can make all sensing decisions for other nodes and communicate them.

In ref. [58], a centralized algorithm is given for finding a small-size connected sensor cover. A straightforward distributed version of the same algorithm is also given. The sensing circles are not necessarily of the same size. In their greedy algorithm, candidate sensors for inclusion are those sensors that partially (not fully) intersect with sensors previously included in the area coverage. For each such sensor, a shortest path from it to one of the sensors already selected is considered. Note that, if coverage circles were the same, the considered path consists of one hop only, since any two sensors whose coverage circles intersect must be neighbors. Circles of candidate sensors divide the area into subelements (each subelement is a small region belonging entirely to some circles and entirely outside the remaining circles). The length of each path is divided by the number of subelements. All sensors on the path with the maxim of such a ratio are added to the covering set (the sensor that starts the selected path is called “leader” here). In the distributed implementation, the sensor that was last added (the leader) initiates the search for a new sensor/path to add. It broadcasts the search message up to $2R$ hops, where $R$ is the maximal hop distance between any two sensors whose circles intersect ($R = 1$, if all circles are equal). Sensors that receive such a message and have partial coverage perform the described iteration, to select a new path and a new sensor “leader.” This process repeats until the entire query region is covered.

We observe that the algorithm presented in ref. [58] may not converge with full coverage of the region. For example, the corner of a region may just be fully covered by the last leader, and all sensors within distance $2R$ may be fully covered as well. On the other hand, uncovered regions may exist in other corners of the region. This problem, however, can be resolved by some additional protocols, such as time-out at sensors that activate if the region is partially covered, but no news is received within the given time-out. More detail regarding parallel actions by several such sensors needs to be added, and the quality of the final result may not differ significantly from the one obtained by centralized implementation.

For simplicity of analysis, consider the case of equal sensing and transmission radii ($R = 1$). Candidate search broadcast involves transmission from the leader, and retransmissions by several of the neighbors to reach all nodes at distance two (an MPR-like broadcasting method can be used), responses from each candidate sensor, and another broadcast to communicate the decision. In dense sensor networks, many sensors are candidate sensors, thus too much traffic for selecting each next sensor is easily generated. Let $R$ be the transmission radius. Initially, all sensors at the distance in the interval $(0,2R)$ from the first leader are candidate sensors for the next leader. There is, unfortunately, no limit on their number inside this circle. It is also difficult to schedule so many transmissions at the MAC layer.

### 11.4.6 Localized Sensor Area Coverage

Tian and Georganas [46] proposed a solution for sensor area coverage in synchronous networks where sensing range is equal to the transmission range. It requires that every node know all its neighbors’ positions before making its monitoring decision. At the beginning of each round, each node selects a time-out interval. At the end of the interval, if a node sees that neighbors that have not yet sent any messages together cover its monitoring area, the node transmits a “withdrawal” message to all its neighbors and goes into the sleep mode. Otherwise, the node remains active, but does not transmit any message. The process repeats periodically to allow for changes in monitoring status. There are several problems in this protocol. Neighboring active sensors may fail without notice, and neighboring sensors may not activate, believing that the sensor is “alive” and monitoring. This problem can be resolved if neighboring information is exchanged at the beginning of each round. However, this then involves significant communication overhead once sensors start to die between activity periods. The other problem is that covering sensors may not be connected; thus, reporting to a monitoring station may not succeed. The authors also discuss the case of different sensing radii at each sensor.

Jiang and Dou [59] describe several improvements to the algorithm in ref. [46]. They apply the criterion that a circle is covered completely if perimeters of other circles covering it are fully covered by other covering circles (note that it can be further simplified, as discussed later, to consider only intersection points). In the algorithm presented in ref. [59], at the beginning of each round, each node sends a hello message to inform about its position. The algorithm from ref. [46] is then applied (which relies on node withdrawals with negative acknowledgments) for all ratios of sensing and transmission radii, using criteria described here. Experimental data in ref. [59] show that this algorithm outperforms PEAS [52] with respect to the number of nodes needed in the coverage, while completely preserving sensing coverage of the original network.

Carle and Simplot-Ryl [28] described a localized algorithm for area coverage for the case of equal sensing and transmission radii. This approach has been generalized by Carle, Gallais, Simplot-Ryl, and Stojmenovic [60] for an arbitrary ratio of sensing and transmission radii. The approach, in addition to being fully localized, has a very small communication overhead. There are two variants in the approach. One requires each sensor to send exactly one message, while the other requires that only nodes that will remain active for covering the area send exactly one message. The two approaches have a trade-off, since one message sent by each sensor that will move to a sleep mode is expected to leave less active sensors for the area coverage.

The basic principle of the algorithm presented in ref. [60] is that each node selects a time-out and listens to messages sent by other nodes before the time-out expires. The time-out can be selected at random, or may depend on the sensor's remaining battery energy. Each received message provides information that a portion of the sensing range is covered. This information is derived from the position of the transmitting sensor, which is reported in the message. The reduction of the required area coverage for monitoring implies an extension of time-out. Nodes with smaller uncovered areas should receive a longer time-out, hoping that a message by a node that is able to cover more area will cover that small portion as well. At the end of time-out, the node verifies whether or not its sensing area is fully covered. If so, it goes to sleep mode in the current round. The two variants differ in whether or not the node then sends the message informing neighbors about the sleep status decision. These messages are called negative acknowledgments. If such a negative acknowledgment is sent by a node that will enter a sleep state, it still informs neighbors about a certain area that has already been covered by sensors that will remain active. The benefits of the negative acknowledgment message are illustrated in Figure 11.7 (where sensing and communication radii are equal). Assume that nodes 1–4 announced their active status. Although node 5 then decides to sleep, its withdrawal message reduces the area to be covered by node 6 by the shaded area in Figure 11.7 (more precisely, node 6 may now conclude that its sensing area is fully covered, which enables it to select sleep mode). That shaded area is covered by active nodes 3 and 4, which are not communication neighbors of 6. Sensor nodes whose sensing area is not fully covered (or fully covered but with a disconnected set of active sensors) when

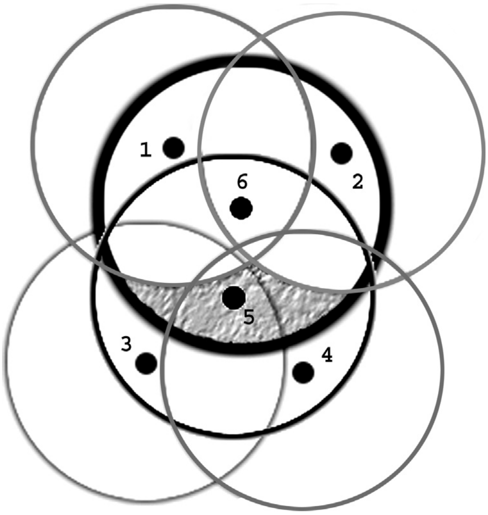


**Figure 11.7** Negative acknowledgment by node 5 reduces the area to be covered by node 6.

the deadline expires decide to remain active for the considered round, and send a message to all its neighbors informing them about the decision. Such a message is called positive acknowledgment [60]. The process repeats in each round in synchronous fashion.

The details of the protocol given in ref. [60] include how the time-out is decided, and how the area coverage and connectivity tests are performed. First consider the case of equal sensing and transmission radii. One important property of the protocol is that no prior knowledge about neighbor existence and location is required. That is, there is no communication overhead coming from the preprocessing step to collect neighborhood information. The test for connectivity of covering circles must be performed whenever $2S > R$, where $S$ and $R$ are sensing and transmission radii.

The network can reselect covering nodes periodically to spread the sensing cost dynamically over all nodes in a fair manner. This method significantly extends the network's life. If the density is more than 30 nodes per unit area, the area-dominating graph is sparse, with nodes having on average three neighbors (this is valid when sensing and communication ranges are equal). In addition, the distance between its two neighboring nodes is typically two-thirds of the transmission radius. Hence, active nodes form a very simple network with a structure similar to regular hexagonal tiling.

### 11.4.7 Multiple Sensor Area Coverage

In ref. [53], the problem of covering each point in an area with at least $k$ sensors ($k$-coverage) is reduced to the simpler problem of determining the similar coverage of all the intersection points of the sensing circles. A sensor is ineligible for turning active if all the intersection points inside its sensing circle are at least $k$-covered. To find all the intersection points inside its sensing circle, a sensor $v$ needs to consider all the sensors in its sensing neighbor set, $SN(v)$. Set $SN(v)$ includes all the active nodes that are within a distance of twice the sensing range to $v$. The algorithm is then combined with SPAN activity-scheduling protocol [61], which is an inefficient version of Wu's dominating set definitions [25] published long before SPAN (see ref. [25] for details).

Abrams, Goel, and Plotkin [62] studied the problem of partitioning the sensors into covers so that the number of covers that include an area, summed over all $k$ areas, is maximized. Three approximation algorithms, assuming $k$ is fixed, are described. Randomized algorithm assigns to each sensor one of $k$ covers at random. In the distributed greedy algorithm, each sensor sets a time-out and listens to decisions made by neighbors, increasing the counter in the appropriate set for each message announcing the decision by a neighbor (the communication radius is assumed to be twice the sensing radius). When time-out expires, each node selects a set for which the corresponding counter is minimal. The centralized greedy algorithm adds some weight, but otherwise runs a similar procedure. This article [62] does not discuss what the best value is for $k$, that is, how many layers of coverage could be reasonably achieved.

An adaptive localized multiple sensor area-coverage algorithm is proposed in ref. [63]. The algorithm [63] adjusts $k$ dynamically to reflect the sensor density. Each sensor node selects a time-out, which depends on the portion of the area not covered by other sensors, and has some random number or other parameter in the formula to avoid simultaneous transmissions by neighbors. Suppose that node $A$ received a message from a neighbor that informed about $i$, the cover-layer number selected by that neighbor, and its geographic coordinates. Node $A$ adjusts the uncovered portion of layer $i$ at the node, and extends appropriately its deadline. When the time-out expires, there are a few options for making a decision (which is then transmitted):

- Assign the layer $j$, which is a minimal number so that the area in layer $j$ is not yet fully covered;
- Among layers covered partially by some neighbors, and not yet fully covered, choose one that maximizes the uncovered area;
- If all layers covered by some neighbors are fully covered, the sensor chooses a new layer, and informs its neighbors about covering it.

This algorithm can be extended to provide layers for activity scheduling in static ad hoc and sensor networks. Existing algorithms only select sensors for the next round, one round at a time. The difference is that the required area coverage is replaced by neighbor coverage. Each node again sets a time-out. The message received from a neighbor gives selected layer $i$ covered by that neighbor. Time-out is extended, and covered neighbors at layer $i$ are updated. At the end of the time-out, the node may select either minimal $j$, so that its neighbors are not covered at layer $j$, or layer $j$ with a maximal number of uncovered neighbors. If all neighbors are covered for all known layers, the node announces participating in the next layer number.

### 11.4.8 Coverage Using the Physical-Layer Model of Sensing

Xing, Lu, Pless, and O'Sullivan [64] consider a probabilistic model of sensor coverage. A point $A$ is covered in a sensor network if the probability at which a target located at $A$ is detected by active sensors is above threshold $\beta$ and the system false-alarm rate is below threshold $\alpha$. The probability of correct detection by a sensor depends on the distance of the sensor. The authors describe a centralized algorithm for deciding which sensors should remain active. Using the active sensors' locations and local false-alarm rate, the location with minimal detection probability is found. If that probability is below $\beta$, then the closest sensor to the considered location is selected to become active. The authors also describe a distributed algorithm that divides the network into grids, selects one sensor in each grid to be coordinator, and then each coordinator follows the centralized algorithm to decide which sensors from its grid need to be active. Neighboring coordinators collaborate to improve the decisions in border areas.

A localized algorithm along these lines can be described as follows [65]. First, we need an approximate function for sensing probability with respect to distance. Then sensors select random time-outs and wait to hear from nearby sensors about their active status. For each received message, the sensor adjusts (normally prolongs) its time-out based on the measured coverage in its local area (e.g., the percentage of its local area having "satisfactory" coverage) and the measured benefit if that sensor is to become active. At the end of the time-out, if the sensor sees that its local area already has satisfactory coverage by other active sensors, it decides to sleep. Otherwise it decides to become active and informs its neighbors about it. The local area to be considered may be a small circle around the sensor that has high values for sensing. Some particular sample of points from the area can be taken to reduce computation time.

### 11.4.9 Variations of the Sensor Area-Coverage Problem

Gui and Mohapatra [66] observe that it is not necessary to achieve a perfect sensing coverage of a moving object. They found that the expected length of a straight-line path and object should move before hitting the boundary of any covered area, for a random sensor placement in the area, can be approximated by $|X|/(4nr)$, where $|X|$ is the area of a given field, $n$ is number of sensors, and $r$ is their sensing radius.

Cardei and Du [67] considered the point coverage problem. A certain number of points needs to be covered by sensors within sensing range of them. Each target point needs to be monitored by at least one sensor. The authors divide the sensors into disjoint sets, each covering target points, with sets being activated in turn. They prove that the problem is NP-complete and propose a centralized solution based on the heuristics of the disjoint set cover.

Shakkottai, Srikant, and Shroff [68] showed that the necessary and sufficient conditions for the random grid network of $n$ nodes, arranged in a grid over a square region of unit area, to cover the unit square region as well as ensure that the active nodes that are connected are of the form $pr^2 = \theta(\log(n)/n)$, where $r$ is the transmission radius of each node, and $p$ is the probability that a node is active.

### 11.4.10 Mobile Sensors for Improved Area Coverage

Zou and Chakrabarty [69] proposed a virtual force algorithm as a sensor deployment strategy to enhance the coverage after an initial random placement of sensors. It is assumed that sensors can move by "virtual force" with the force's strength determined by node distance.

Cao, Wang, La Porta, and Zhang [70] considered the problem of moving some sensors from their initial random placement in order to cover some areas that were not covered by either the nature of randomness or some other effects such as wind. It is also assumed that sensors can move after gathering some information from neighbors. The algorithm proceeds in rounds. In each round, sensors communicate to local neighbors in order to construct Voronoi diagrams. Each sensor then subtracts its sensing area from its Voronoi polygon, and moves in the direction of the largest uncovered piece of area. The process repeats until no further improvement is possible. The approach appears suitable when robots, equipped with sensors, are monitoring an area, which can also be monitored by some static sensors. Voronoi diagram construction, however, may not always be locally constructed, and it may be better to use localized versions such as the partial Delauny triangulation [71]. The Gabriel graph can also be used. An alternative approach may be to use face routing [72] to estimate the size of a hole, find its centroid, estimate the number of sensors that should move toward the centroid, and provide the best possible information to sensors for their move.

Wang, Cao, and La Porta [73] propose a proxy-based sensor deployment protocol. Instead of moving iteratively, sensors calculate their target locations based on a distributive iterative protocol. Current proxy sensors advertise the service of mobile sensors to their neighborhoods (up to certain parameter distance), searching for a better coverage location. They collect bidding messages and choose the highest bid. Then they delegate the bidder as the new proxy. The iterative moves are logical, not physical. Actual movement only occurs when sensors determine their final locations. If the bidding process is local, the sensor movement and the area-coverage gains may be restricted. If the bidding process includes neighbors at several hops distance, the communication overhead for bidding becomes significant. Bidding decisions are based on price (number of logical movements made so far) and distance that the moving sensors are physically supposed to move altogether. A procedure to prevent multiple healing is described, which includes some message overhead. The bidding criterion does not include lost area coverage for moving out of the current position. It is not certain whether the described procedure is always loop-free and always converging. The difference between sensing and transmission radii (the ratio is not discussed in ref. [73]) has a direct impact on message complexity.

The iterative nature of logical moves in ref. [73] may still easily lead each mobile sensor to a local minimum. It may be better to apply an expanding ring strategy [74] in search of the best proxy, by using the increasing sizes of distances from each mobile sensor, and asking sensors within the ring to respond with their biddings (these responses may be suspended on the way to the mobile sensor by intermediate nodes that learn about better bids). The bidding price also includes the traveling distance for the mobile sensor, which chooses the best bid. Some mechanisms for avoiding multiple healing of the same hole need to be added in the protocol, such as reporting to only one mobile sensor (note that this is not sufficient, since few sensors can be located around the same hole). Instead of moving, mobile sensors send the best bid so far to the next ring, asking sensors from that ring to report their bids only if they have a better bid to offer. This protocol should reduce the number of reports, because substantially more free area needs to be made available to justify longer movement.

Wu and Yang [75] propose a scan-based movement-assisted sensor deployment method (SMART) that uses scan and dimension exchange to achieve a balanced state. In SMART, a given rectangular sensor field is first partitioned into a 2-D mesh through clustering. Each cluster corresponds to a square region and has a clusterhead which is in charge of bookkeeping and communication with adjacent clusterheads. Clustering is a widely used approach in sensor networks for its support for design simplification. In fact, it is shown in ref. [76] that clustering is the most efficient for sensor network where data are continuously transmitted. A hybrid approach is used for load balancing, where the 2-D mesh is partitioned into 1-D arrays by row and by column. Two scans are used in sequence: one for all rows, followed by the other for all columns. Within each row and column, the scan operation is used to calculate the average load and then to determine the amount of overload and underload in clusters. The load is shifted from overloaded clusters to underloaded clusters in an optimal way to achieve a balanced state. By optimal, we mean the minimum number of moves and minimum total moving distance and minimum number of moves. By a balanced state, we refer to a state where the maximum cluster size (the number of sensors in a cluster) and the minimum cluster size are different by at most 1. Using this 2-D scan without global information, each sensor moves at most twice, although it may not be globally optimal in terms of total moving distance in 2-D meshes. SMART addresses a unique problem called *empty cells* in sensor networks and provides a local solution to it.

Mobile and static sensors can use the perimeter created by a Gabriel graph to make moving decisions after only one iteration, as elaborated in ref. [72]. First, static sensors will locally communicate to ensure that their biddings are made for nonintersecting coverage areas. They then send their bidding by the GFG routing protocol [72] (which guarantees delivery in connected unit-disk graphs) in an arbitrary direction. Such routing will end up creating a loop along the perimeter. The node that detects the loop will store the bid. Mobile sensors also will search for the best bid by routing in arbitrary directions, ending on a perimeter. A similar idea has been described in ref. [77] for the purpose of providing location service. If the network of static sensors is disconnected, then mobile sensors will send one message to each connected component and search several perimeters. Mobile sensors will set a criterion for selecting the bid, which will include the cost for moving to a new location, and gain made for changing the coverage area (the difference between current and new coverage). After making a full traversal along the perimeter, the message sent by the mobile sensor will select the best bid and return it to the node responsible for the bid, which in turn will eliminate the bid to prevent other mobile sensors from taking it. The message is then routed back to the mobile sensor, which performs the indicated move. Note that the proposed protocol has only one iteration, flooding type of message circulation is avoided, and the message cost is made quite uniform.

## 11.5 RELATED SURVEY ARTICLES

Because of space limitations, this chapter did not cover all relevant aspects of the considered problems. For a more complete coverage, the reader is referred to several complementary book chapters [11,78–81]. In particular, ref. [78] contains comprehensive coverage of the topology aspects of these problems, ref. [11] discusses broadcasting with directional antennas and reliable broadcasting, refs. [79] and [81] give comprehensive coverage for broadcasting with adjustable transmission powers with omnidirectional and directional antennas (that is, the minimum energy broadcasting problem). Further, ref. [79] contains comprehensive coverage of broadcasting reliability issues, deciding transmission radii, and resource-aware broadcasting. Probabilistic broadcasting protocols are covered in refs. [11] and [79] (see also the recent article, ref. [82]). Finally, ref. [80] describes routing and broadcasting schemes for hybrid ad hoc and sensor networks.

## ACKNOWLEDGMENTS

This research was supported by grants from NSERC, INRIA, and NSF. The authors are grateful to Roger Wattenhoffer for a fruitful e-mail discussion that contributed to clarification of major aspects of dominating sets and subsequently to an improved presentation of this chapter. We are also grateful to Stefano Basagni for a careful reading of the manuscript and several useful suggestions.

## EXERCISES

**11.1.** Prove that intergateway and gateway nodes in Wu's concept [14,25] create dominating sets [26].

**11.2.** Write a procedure for deciding whether or not a node is an intermediate, intergateway or gateway node in Wu's concept [14,25,26].

**11.3.** Give a formal definition of an enhanced dominating set, generalizing the case of coverage by one neighbor presented in this chapter. Describe the appropriate efficient algorithm, and prove that the new set is indeed a CDS [30].

**11.4.** To increase reliability, double (and in general $t$-coverage) dominating sets can be considered. In this approach, every neighbor needs to be covered by two (in, general, $t$) neighbors, instead of only one. Describe some backbone construction methods based on double domination, and some broadcasting schemes that would require each node to receive the message at least twice [30].

**11.5.** Assume that each node knows its geographic location, but has no knowledge about the existence or position of its neighbors. Describe a beaconless broadcasting scheme that will work with such assumptions and will minimize the number of retransmissions [81,83].

**11.6.** Suppose that broadcast messages need to be acknowledged. Describe a protocol that will minimize the number of acknowledgment packets for reliable broadcasting [81].

**11.7.** Generalize the sensor area-coverage scheme [60], described in this chapter, for the case of unequal sensing radii at sensor nodes.

**11.8.** Give an example showing that GAF [21] can disconnect the network [17]. (*Hint*: Consider scenarios with nodes near corners of grids and near some empty grids.)

**11.9.** Suppose that sensor nodes are placed at vertices of a regular hexagonal tiling with side length $r$ corresponding to the transmission radius. Prove that the side length that minimizes the total transmission power used when all nodes retransmit the packet is $r = (2c/(\alpha - 2))^{1/\alpha}$ [43].

## REFERENCES


1. F. Kuhn and R. Wattenhoffer. Constant-time distributed dominating set approximation. In *Proceedings of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, Boston, Massachusetts, July 2003.
2. L. Barriere, P Fraigniaud, L. Narajanan, and J. Opatrny. Robust position-based routing in wireless ad hoc networks with irregular transmission ranges. *Wireless Communications and Mobile Computing*, **3**(2):141–153, 2003.
3. I. Stojmenovic, A. Nayak, J. Kuruvila, F. Ovalle-Martinez, and E. Villanueva-Pena. Physical layer impact on the design and performance of routing and broadcasting protocols in ad hoc and sensor networks. *Computer Communications*, forthcoming.
4. M. Agarwal, J. H. Cho, L. X. Gao, and J. Wu Energy Efficient Broadcast in Wireless Ad Hoc Networks with Hitch-hiking, to appear in *ACM/Kluwer MONET*.
5. H. Chan and A. Perrig. ACE: An emergent algorithm for highly uniform cluster formation. In *Proceedings of the European Workshop on Wireless Sensor Networks (EWSN 2004)*, pages 154–171, Berlin, Germany, January 2004.
6. D. A. Fisher and H. F. Lipson. Emergent algorithms: A new method for enhancing survivability in unbounded systems. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences (HICSS-32)*, Maui, Hawaii, January 1999.

7. D. Niculescu. Positioning in ad hoc sensor networks. *IEEE Networks*, **18**:24–29, July 4, 2004.
8. L. M. Feeney. Energy efficient communication in ad hoc wireless networks. In *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, and S. Giordano, and I. Stojmenovic (eds.), pages 301–327, IEEE/Wiley, 2004.
9. V. Raghunathan, C. Schurger, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, **19**:40–50, 2002.
10. M. L. Sichitiu. Cross-layer scheduling for power efficiency in wireless sensor networks. In *Proceedings of IEEE INFOCOM 2004*, Hong Kong, China, March 2004.
11. I. Stojmenovic and J. Wu. Broadcasting and activity scheduling in ad hoc networks. In *Mobile Ad Hoc Networking*, pages 205–229, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic (eds.), IEEE Press, 2004.
12. S. Basagni, M. Mastrogiovanni, and C. Petrioli. A performance comparison of protocols for clustering and backbone formation in large scale ad hoc networks. *In Proceedings of them 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2004)*, Fort Landerdale, Florida, October 2004.
13. C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, **15**(7):1265–1275, 1997.
14. I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, **13**(1): 14–25, January 2002.
15. S. Basagni. Distributed clustering for ad hoc networks. In *Proceedings of the 1999 International Symposium on Parallel Architectures Algorithms, and Networks (ISPAN 99)*, pages 310–315, Freemantle, Australia, June 1999.
16. P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating sets in wireless ad hoc networks. *Mobile Networks and Applications (MONET)*, **9**(2):141–149, April 2004.
17. S. Basagni, A. Carosi, and C. Petrioli. Sensor-DMAC: Dynamic topology control for wireless sensor networks. *In Proceedings of the IEEE VTC*, Los Angeles, California, September 2004.
18. J. Wu and F. Dai. A distributed formation of a virtual backbone in MANETS using adjustable transmission ranges. In *Proceedings of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 372–379, Tokyo, Japan, March 2004.
19. F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing newly deployed ad hoc and sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 260–274, Philadelphia, Pennsylvania, September 2004.
20. J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete mobile centers. In *Proceedings of the 17th Annual Symposium on Computational Geometry (SCG)*, pages 188–196, Boston, Massachusetts, June 2001.
21. Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 70–84, Rome, Italy, July 2001.
22. L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, **13**:383–390, 1975.

23. C. Adjih, P. Jacquet, and L. Viennot. Computing Connected Dominating Sets with Multipoint Relays. Research Report #4597, INRIA, October 2002. To appear in Ad Hoc & Sensor Wireless Networks. Vol. 1, No. 1–2, 27–40.
24. J. Wu. An enhanced approach to determine a small forward node set based on multipoint relay. In *Proceedings of the IEEE VTC*, Orlando, Florida, October 2003.
25. J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, pages 7–14, Seattle, Washington, August 1999.
26. I. Stojmenovic. Comments and corrections to "Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks." *IEEE Transactions on Parallel and Distributed Systems*, forthcoming.
27. F. Dai, and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, **15**(10), 2004.
28. J. Carle and D. Simplot-Ryl. Energy efficient area monitoring for sensor networks. *IEEE Computer*, **37**(2):40–46, February 2004.
29. I. Stojmenovic. Enhanced Dominating Sets. In preparation.
30. I. Stojmenovic. Data gathering and activity scheduling in ad hoc and sensor networks, Proc. International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks, Chicago, USA, June 11–12, 2004.
31. J. Shaikh, J. Solano, I. Stojmenovic, and J. Wu. New metrics for dominating set based energy efficient activity scheduling in ad hoc networks. In *Proceedings of the IEEE Conference on Local Computer Networks*, pages 726–735, Bonn, Germany, October 2003.
32. S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 152–162, Seattle, Washington, August 1999.
33. H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '00)*, pages 61–68, Boston, Massachusetts, August 2000; see also *Computer Communications*, **24**:3–4, 353–363, February 2001.
34. W. Peng and X.-C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the 1st ACM International Symposium on Mobile and Ad Hoc Networking and Computing*, pages 129–130 Boston, Massachusetts, August 2000.
35. I. Stojmenovic and M. Seddigh. Broadcasting algorithms in wireless networks. In *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet SSGRR*, L'Aquila, Italy, July–August 2000.
36. J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. In *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 460–467, May 2003.
37. G. Calinescu, I. Mandoiu, P. J. Wan, and A. Zelikovsky. Selecting forwarding neighbors in wireless ad hoc networks. In *Proceedings of the 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication (DIAL M 2001)*, pages 34–43, Rome, Italy, 2001.

38. A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference System Sciences (HICSS '02)*, Big Island, Hawaii, January 2002.

39. M. T. Sun and T. H. Lai. Location aided broadcast in wireless ad hoc network systems. In *Proceedings of the IEEE Symposium on Ad Hoc Wireless Networks, at GLOBECOM*, November 2001.

40. W. Peng and X. Lu. AHBP. An efficient broadcast protocol for mobile ad hoc networks. *Journal of Science and Technology* (Bejing, China), 2002.

41. W. Peng and X. Lu. Efficient broadcast in mobile ad hoc networks using connected dominating sets. In *Proceedings of the 7th International Conference on Parallel and Distributed Systems (ICPADS 2000)*, Iwate, Japan, July 2000.

42. J. Wu and W. Lou. Extended multipoint relays to determine connected dominating sets in MANETs. In *Proceedings of the 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004)*, Santa Clara, California, October 2004.

43. F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Target transmission radius over LMST for energy-efficient broadcast protocol in ad hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, Paris, June 2004.

44. M. Cardei and J. Wu. Energy-efficient coverage problems in wireless ad hoc sensor networks. *Computer Communications*, forthcoming.

45. T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja. Sensor deployment strategy for target detection. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 42–48, Atlanta, Georgia, September 2002.

46. D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. *Wireless Communications and Mobile Computing*, **3**:271–290, 2003.

47. J. Wu and S. Yang. Coverage and connectivity in sensor networks with adjustable ranges. In *Proceedings of the 2004 International Workshop on Mobile and Wireless Networks (MWN)*, Montreal, Canada, August 2004.

48. H. Zhang and J. C. Hou. *Maintaining sensing coverage and connectivity in large sensor networks*. Technical Report UIUCDCS-R-2003-2351, Department of Computer Science, University of Illinois at Urbana-Champaign, June 2003. (Alsotoappear in *Ad Hoc & Sensor Wireless Networks*).

49. G. Xing, C. Lu, R. Pless, and Q. Huang. Greedy geographic routing is good enough in sensing covered networks. INFOCOM 2004.

50. B. Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *Proceedings of the 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2004)*, pages 475–483, Fort Lauderdale, Florida, October 2004.

51. H. Zhang and J. Hou. On deriving the upper bound of alpha-lifetime for large sensor networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 121–132, Tokyo, Japan, May 2004.

52. F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, page 28, Providence, Rhode Island, May 2003.

53. Di Tian and N. D. Georganas. Location and calculation-free node scheduling schemes in large wireless sensor networks. *Ad Hoc Networks*, forthcoming.
54. X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. D. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys '03)*, Los Angeles, California, November 2003.
55. Di Tian. Node Activity Scheduling Schemes in Large-Scale Wireless Sensor Networks. Ph.D. thesis, SITE, University of Ottawa, 2004.
56. T. Yan, T. He, and J. A. Stankovic. Differentiated surveillance for sensor networks. In *Proceeding of the 1st International Conference on Embeded Networked Sensor Systems (SenSys '03)*, pages 51–62, Los Angeles, California, November 2003.
57. C. F. Hsin and M. Liu. Network coverage using low duty-cycled sensors: Random and coordinated sleep algorithms. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, California, April, 2004.
58. H. Gupta, S. R. Das, and Q. Gu. Connected sensor cover: Self-organization of sensor networks for efficient query execution. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Annapolis, Maryland, June 2003.
59. J. Jiang and W. Dou. A coverage preserving density control algorithm for wireless sensor networks. In *Proceedings of the 3rd International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW '04)*, (LNCS 3158), pages 42–45, Vancouver, July 2004.
60. J. Carle, A. Gallais, D. Simplot-Ryl, and I. Stojmenovic. Localized Sensor Area Coverage with Small Communication Overhead. In preparation.
61. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *In Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, July 2001.
62. Z. Abrams, A. Goel, and S. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (ISPN)*, pages 424–432, Berkeley, California, April 2004.
63. I. Stojmenovic. An Adaptive Localized Algorithm for Multiple Sensor Area Coverage. In preparation.
64. G. Xing, C. Lu, R. Pless, and J. O'Sullivan. Co-Grid: An efficient coverage maintenance protocol for distributed sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Network (IPSN)*, Berkeley, California, April 2004.
65. I. Stojmenovic. A Localized Algorithm for Sensor Area Coverage Based on a Realistic Physical Layer. In preparation.
66. C. Gui and P. Mohapatra. Target Tracking and Surveillance Using Sensor Networks. Paper presented at the International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks, Fort Lauderdale, Florida, February 2004.
67. M. Cardei and D. Z. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, forthcoming.
68. S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: Coverage, connectivity, and diameter. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, California, June 2003; See also *Ad Hoc Networks*, forthcoming.

69. Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *Proceedings of IEEE INFOCOM 2003*, pages 1293–1303, San Francisco, California, March 2003.

70. G. Cao, G. Wang, T. La Porta, and W. Zhang. Distributed Algorithms for Deploying Mobile Sensors. Paper presented at the International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks, Fort Lauderdale, Florida, February 2004.

71. Xiang-Yang Li, Ivan Stojmenovic, and Yu Wang. PartialDelaunay triangulation and degree limited localized Bluetooth multihop scatternet formation. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 4, April 2004, 350–361.

72. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, **7**(6):609–616, November 2001.

73. G. Wang, G. Cao, and T. La Porta. Proxy-based sensor deployment for mobile sensor networks. In *Proceedings of the 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2004)*, Fort Lauderdale, Florida, October 2004.

74. I. Stojmenovic. Deploying Mobile Sensors for Improved Area Coverage. In preparation.

75. J. Wu and S. Yang. SMART: A Scan-based Movement-Assisted sensoR deploymenT Method in Wireless Sensor Networks. Paper presented at INFOCOM 2005, Miami, Florida, March 2005.

76. W. Heinzelman. Application-Specific Protocol Architectures for Wireless Networks. Ph.D. thesis, Massachusetts Institute of Technology, 2000.

77. I. Stojmenovic. *A Scalable Quorum Based Location Update Scheme for Routing In Ad Hoc Wireless Networks*. Technical Report TR-99-09, SITE, University of Ottawa, September 1999.

78. X.Y. Li and I. Stojmenovic. Broadcasting and topology control in wireless ad hoc networks. In *Handbook of Algorithms for Mobile and Wireless Networking and Computing*, A. Boukerche and I. Chlamtac (eds.), CRC Press, forthcoming.

79. F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic. Energy-efficient broadcasting in wireless mobile ad hoc networks. In *Resource Management in Wireless Networking*, Mihaela Cardei, Ionut Cardei, and Ding-Zhu Du (eds.), Kluwer, forthcoming.

80. François Ingelrest, David Simplot-Ryl, and Ivan Stojmenovic. Routing and broadcasting in hybrid ad hoc and sensor networks. In *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Jie Wu (ed.), CRC Press.

81. Justin Lipman, Johnson Kuruvila, and Ivan Stojmenovic. Localized broadcasting in ad hoc networks. In *Wireless Ad Hoc and Sensor Networks*, Ahmed Safwat (ed.), Kluwer, forthcoming.

82. L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti. Localized techniques for broadcasting in wireless sensor networks. In *Proceedings of the DIALM-PODC Joint Workshop, on Foundations of Mobile Computing*, pages 41–51, Philadelphia, Pennsylvania, October 2004.

83. I. Stojmenovic. Beaconless Area Based Broadcasting. In preparation.

HL1. Z. Hu and B. Li. On the fundamental capacity and lifetime limits of energy-constrained wireless sensor networks. In *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '04)*, pages 38–47, Toronto, Canada, 2004.