7

Sleep-based topology control

7.1 Overview

As we discussed in Chapter 2, the core functionality of a wireless sensor network depends on having a network topology that satisfies two important objectives: connectivity and coverage. Connectivity is essential to be able to move movement data between different points within the network, while coverage of the environment ensures that critical phenomena and events can be detected and monitored.

The problem of constructing an initial topology is of course the problem of node deployment. Beyond this, there are many scenarios in which *topology control* is desired. In its broadest sense, topology control can be defined as the process of configuring or reconfiguring a network's topology through tunable parameters after deployment. There are three major tunable parameters for topology control:

1. **Node mobility:** In WSNs consisting of mobile nodes, such as robotic sensor networks, both coverage and connectivity can be adapted by moving the nodes accordingly. We discussed the work that has been done on mobile node deployments and related topology configuration approaches in Chapter 2.
2. **Transmission power control:** In WSNs with static nodes, if the deployment density is already sufficient to guarantee the required level of coverage, the connectivity properties of the network can be adjusted by tuning the transmission power of the constituent nodes. We discussed these kinds of power control-based topology configuration techniques in Chapter 2 as well.
3. **Sleep scheduling:** Finally, consider large-scale static WSNs deployed at a high density, i.e. *over deployed*. In this case, the appropriate topology control

mechanism that provides energy efficiency and extends network lifetime is to turn off nodes that are redundant. The remaining nodes form the active network over which sensing, computation, and communication take place. The challenge is to periodically determine if any active nodes have malfunctioned or if they have suffered energy depletion, so that additional sleeping nodes can be activated if needed. Sleep-based topology control mechanisms are the subject of this chapter.

Certainly, in applications (e.g. in industrial process monitoring) where a limited number of expensive nodes must be deployed in specific locations and must all be active at all times, it is unlikely that redundant nodes will be deployed. Thus sleep-based topology control schemes are not always relevant. However, in other contexts, where (a) the nodes are relatively inexpensive (so that the cost of redundancy is not unreasonably high), (b) the deployed location is remote (so that adding additional nodes at a later time is either undesirable or infeasible), and (c) the precise positioning of each active sensors is not essential, over deploying the network may provide some significant advantages:

1. **Longer lifetime:** The first is that an over-deployed network, if carefully managed through the topology control mechanisms described in this chapter, can have a longer network lifetime. This is obtained by keeping redundant nodes asleep until they are needed to replace the nodes that have failed or depleted their energy.
2. **Robustness:** While one could argue that longer lifetimes could also be obtained for a lower cost merely by adding longer-lifetime batteries, deploying redundant nodes also provides robustness to device failures for reasons other than energy depletion.
3. **Tunable connectivity/coverage:** Another significant advantage of over deploying a network is that it enables the possibility of adaptive topology control techniques that provide tunable levels of connectivity and coverage.

To save energy and maximize lifetime for an over-deployed network, it is essential to ensure that only the minimal set of nodes needed are awake at any time. The goal of sleep-based topology control is to determine which nodes should be awake at a given time to serve both connectivity and coverage needs. Because of the unattended nature of these networks, it is essential that the topology control process be self-configuring and adaptive. In most cases, distributed, localized techniques are essential, to minimize overhead and enable quick reaction to changes.

The basic approach used in many distributed sleep-based topology control techniques for WSN is the following. A set of sleep-related states is defined for each node, and the nodes then transition between these states depending on explicit messages with their neighbors, overheard messages/beacons, or implicit timers.

Consider the following simple example as an illustration. Each node can be in one of three states: sleep, test, and awake. By default a node switches periodically, using a timer, between sleep and test states. During the test state, a local eligibility rule is applied (e.g., "are there less than K awake neighbors?", "is there data intended for me?") to determine whether the node should wake-up. If the rule is satisfied, then the node transitions to the wake state. When the node is an active state, it may (a) remain there forever, if the goal is to ensure that the active backbone of the network remains up at all times, or (b) return back to sleep after some timer expires; for instance, if the eligibility criterion is no longer satisfied.

7.2 Constructing topologies for connectivity

We first describe several sleep-based topology control techniques that have been proposed to address connectivity alone.

7.2.1 Basic/adaptive fidelity energy conserving algorithms (BECA/AFECA)

The basic energy-conserving algorithm (BECA) [229] has three states: sleep, listen, and active. It is illustrated in Figure 7.1(a). The eligibility criterion used to transition from listen to active state is the presence of routing traffic in which the node should participate. Different timers T_s and T_l are used to go from sleep to listen state and back. The node will also transition to sleep from active state if there is no relevant traffic for T_a units. The node can also transition directly from sleep to active state if it has outgoing data to send.

The closely related adaptive fidelity energy-conserving algorithm (AFECA) extends BECA by adapting the sleep time depending on the neighborhood density for increased sleep efficiency. Specifically, the sleep time is chosen as the basic sleep time multiplied by a random number between 1 and N, where N is the number of neighbors estimated in the listening period.

7.2.2 Geographic adaptive fidelity (GAF)

The geographic adaptive fidelity (GAF) technique [230], illustrated in Figure 7.1(b) uses sleep, discovery, and active states. The node moves from the

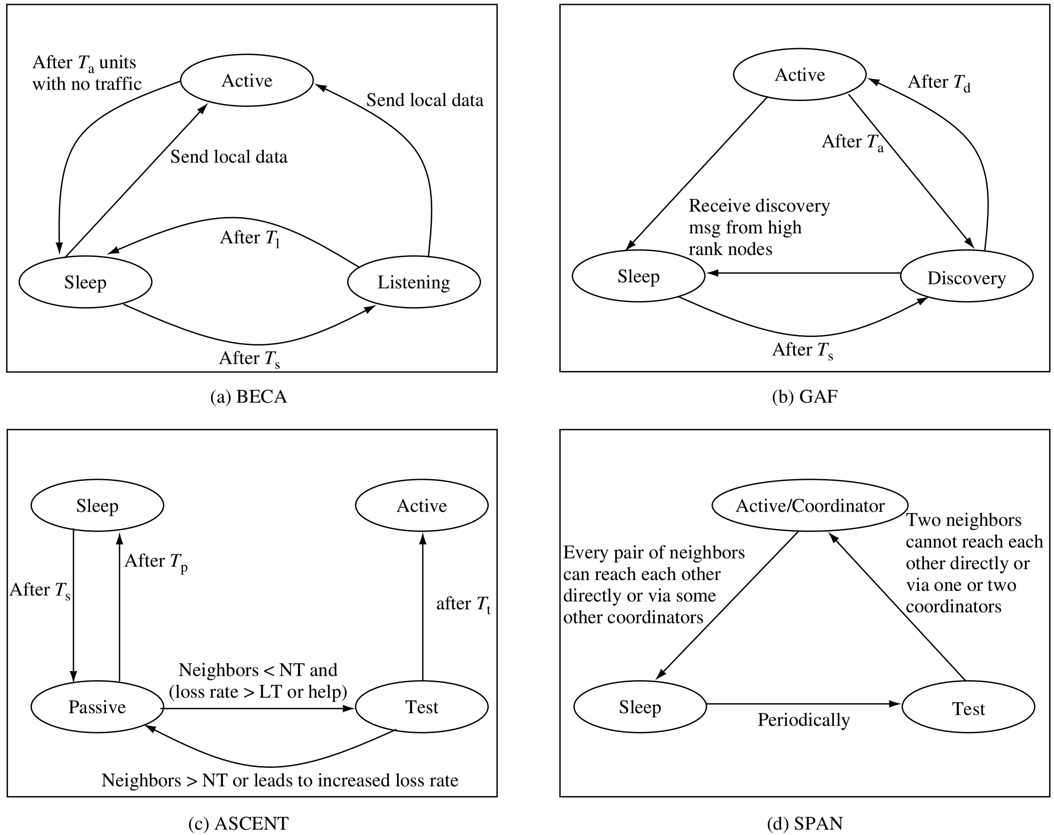

Figure 7.1 State transition diagrams for various sleep-based topology control techniques aimed at connectivity alone

discovery state to the active state and back with timers T_d and T_a respectively; and from the discovery state or the active state to the sleep state if it detects any other higher-priority nodes active within its geographical virtual grid (explained below) through the reception of a discovery message within its neighborhood. It moves from the sleep state to the discovery state after a time T_s. The nodes in discovery/active state transmit discovery messages containing their own ID, the grid ID, and estimated residual lifetime. Nodes with longer residual lifetimes have greater priority.

The goal of GAF is therefore to ensure that a single node with the highest remaining lifetime is awake in each virtual grid. The virtual square grids are created to ensure that all nodes in one grid can communicate with all others in the adjacent grid. This is accomplished by setting the side length of each grid cell r to be sufficiently small so that $r < \sqrt{R/5}$, where R is the radio communication range.

The related cluster-based energy conservation algorithm (CEC) [230] eliminates the need for geographic information necessary to set up the virtual grids. Instead a set of clusters is created such that each has an elected cluster-head (a node is made cluster-head if it has the greatest residual lifetime among its neighbors). To ensure connectivity, additional gateway nodes are elected – primary gateway nodes directly connect two distinct cluster-heads, while secondary gateway nodes connect to other cluster-heads through other secondary/primary gateway nodes. All nodes that are not cluster-heads or gateway nodes are eligible to sleep. The sleep timers are set so that nodes wake-up to run a re-election before the cluster-head's energy is depleted. Nodes that are asleep may wake-up to send their own information, and any data intended for them must be buffered for pick-up after they are awake.

7.2.3 Adaptive self-configuring sensor network topology control (ASCENT)

ASCENT [19] is intended to adapt to highly dynamic environments; nodes wake-up to assist in routing depending on the number of active neighbors and the measured data loss rates in their vicinity.

In ASCENT, illustrated in Figure 7.1(c), nodes are in one of four states: sleep, passive, test, and active. A node in a test state goes to the active state after a timer T_t unless it finds that the number of active neighbors is greater than the threshold NT or that its participation would increase the loss rate further (e.g. due to congestion), in which case it transitions back to the passive state. From the passive state the node transitions to the sleep state after a timer T_p

unless the number of active neighbors is below the desired threshold and either the loss rate is higher than a threshold LT or a help message is received, in which case it transitions to the test state. Help messages are sent by a node that needs the assistance of neighboring nodes to help relay messages to a given destination that it currently is unable to because of high loss. Finally, a node in the sleep state transitions to the passive state after a timer T_s. A key point to note in the basic ASCENT scheme is that nodes do not transition back to sleep from the full active state – this is suitable for applications where nodes in the constructed topology are needed/expected to be active until energy depletion. Message losses are assumed to be detected based on missing packet sequence numbers.

There are a number of tunable parameters in ASCENT that must be adjusted depending on the application needs. The average degree and distance between active nodes needed for connectivity (or even coverage) influence NT and LT. The timers T_t and T_p effect a tradeoff between decision quality and energy efficiency, while T_s provides a tradeoff between energy efficiency and latency of response to failures in the network.

7.2.4 Neighborhood coordinators (SPAN)

SPAN [23] is a related randomized approach to topology control that aims to provide connectivity with redundancy. Only a subset of nodes called coordinator nodes are in active state at any time. Its state transitions are illustrated in Figure 7.1(d). Nodes that are not coordinators go to sleep, waking up periodically to go to the test state to send HELLO messages and check for eligibility to become coordinators. The coordinator eligibility rule is the following: if two neighbors cannot reach each other directly or via one or two coordinators, then the node in test state should become an active coordinator. The decision regarding the eligibility rule is made based on the content of HELLO packets sent by all nodes in which they announce their current coordinators and current neighbors. This algorithm provides for a connected backbone, but not with a minimal set of nodes. It provides sufficient redundancy to ensure that there are coordinators in every radio broadcast range in the network, to minimize congestion bottlenecks.

A randomized prioritized backoff is used to ensure that multiple nodes do not elect to become coordinators simultaneously. Nodes that are likely to benefit a greater set of neighbors are more likely to win contention to become coordinators.

Nodes withdraw as coordinators (i.e. go back from active to sleep state) if they no longer satisfy the eligibility rule, or after some timer period, to ensure load balancing.

7.3 Constructing topologies for coverage

The above-described topology control techniques have focused on waking up the radios of sufficient nodes to maintain good network connectivity. We now turn to the problem of ensuring good coverage in addition to connectivity. Most of these approaches are similar in spirit to those discussed above, except they involve coverage-based eligibility rules for activation of nodes.

7.3.1 Probing environment and adaptive sleep (PEAS)

The probing environment and adaptive sleep (PEAS) technique [235] aims to provide topology control for highly dynamic environments. There are three states in PEAS, as shown in Figure 7.2(a): sleep, probe, and active. From the sleep state the node uses a randomized timer with exponential distribution of rate λ to transition to the probe state. Randomized wake-up times are used to spread the probes from nodes so as to minimize the likelihood that any portion of the network is left without an active node for too long. The rate λ is adapted depending on the environment to ensure that the sleeping rate in every part of the network is about the same desired value λ_d, regardless of spatio-temporal inhomogeneities in node density. In the probing state, a node detects if there is any active node within a probing range R_p, by sending a PROBE message at the appropriate transmit power and listening for REPLY messages. If there are no responses, the node transitions to the active state and stays there until its energy is depleted. If there are REPLY messages, the node computes an updated λ, as described below, and transitions to the sleep state.

Active nodes measure the rate at which PROBEs are heard from nodes in their neighborhood to estimate the current aggregate neighborhood probing rate $\hat{\lambda}$. The measured probe rate $\hat{\lambda}$ and the desired probe rate λ_d are sent in the REPLY message in response to each PROBE message. The probing node updates its rate λ using the following rule to ensure that the aggregate rate stays around the desired probe rate.

$$\lambda_{new} = \lambda_{current} \frac{\lambda_d}{\hat{\lambda}} \tag{7.1}$$

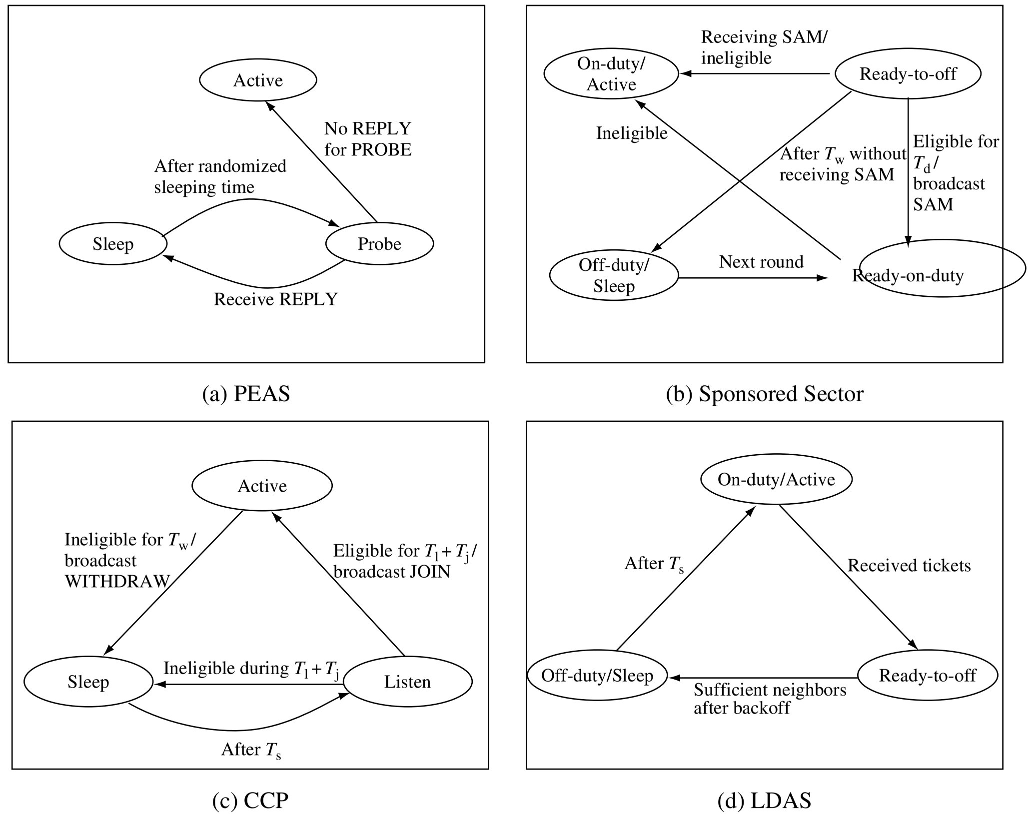

Figure 7.2 State transition diagrams for various sleep-based topology control techniques that address coverage

The choice of λ_d depends on the application tolerance for delays, with the tradeoff being energy efficiency.

Besides connectivity, the PEAS technique also provides tunable coverage. This is done through the eligibility rule in PEAS, where nodes try to ensure that there is an active neighbor within a configurable probing range R_p. R_p could be arbitrarily smaller than the communication range, and allows the density of active nodes to be varied. This does not necessarily guarantee, though, that the original coverage is preserved, which is the goal of the next scheme.

7.3.2 Sponsored sector

Tian and Georganas [210] present an approach that aims to turn off redundant nodes while preserving the original coverage of the network. Each node checks to see if its coverage area is already covered by active neighbors before deciding whether to be on or off. Four states are used: off-duty, ready-on-duty, on-duty, ready-to-off, as seen in Figure 7.2(b).

Nodes that are in the ready-on-duty state investigate whether they are eligible to turn off their sensor and radio, by examining their neighbors' coverage. If they are eligible, they first wait a random backoff time T_d, and broadcast a status advertisement message (SAM), before transitioning to the ready-to-off state. Any neighboring nodes with a longer backoff will not consider these nodes that issued a SAM before them in their neighbor coverage determination. This prevents multiple nodes from shutting off simultaneously. If nodes determine they are ineligible to turn off, they transition to the on-duty state. From the ready-to-off state, after a timer T_w, nodes transition to off-duty unless they receive a SAM from a neighbor and find that they are ineligible to turn off. If they are ineligible to turn off, they transition to on-duty.

The process is conducted via multiple sequential rounds. At the beginning of each round, each node sends a position advertisement message (PAM) to all neighbors within sensing range (by adjusting the transmission power accordingly), containing its sensing range as well as position. The coverage eligibility rule is determined by each node independently in the ready-on-duty state as follows. Based on the PAM messages, each node determines which angular sector of its own sensing range is covered by each of its neighbors (that have not already issued SAMs). If the union of all these *sponsored sectors* is equal to the node's own coverage area, it determines itself to be eligible to turn off. At the end of each round, eligible nodes turn off, while other nodes continue to sense.

7.3.3 Integrated coverage and connectivity protocol (CCP)

Wang *et al.* [220] present an integrated coverage and connectivity protocol (CCP) for sleep-based topology control. This protocol is based on two analytical results we discussed in Chapter 2:

- A convex region is K-covered by a set of sensors if all intersection points between sensing circles of different nodes and between sensing circles and the boundary are at least K-covered.
- A set of nodes that K-cover a given convex region form a K-connected communication graph if $R_c \geq 2R_s$, where R_c, R_s are the communication and sensing ranges respectively.

CCP, shown in Figure 7.2(c), also uses three states: sleep, listen, and active. Sensor nodes transition from the sleep to the listen state after a timer T_s. Nodes in the listen state start a timer with duration T_l. They evaluate their eligibility if they receive a HELLO, WITHDRAW, or JOIN message. A node is eligible if all intersection points of its own circle with those of other sensors or the region boundary are at least K-covered. If the node is eligible, it starts a join timer T_j, otherwise it returns to the sleep state after the timer T_l expires. If a node hears a JOIN beacon from a neighbor after the join timer is started it becomes ineligible and cancels the join timer T_j and goes to sleep. If the join timer is not cancelled, when it expires the node broadcasts a JOIN beacon and enters the active state. In active state, a node periodically sends HELLO messages. An active node that receives a HELLO message checks its own eligibility; if ineligible, it starts a withdraw timer T_w after which it broadcasts a WITHDRAW message and goes to sleep. If it becomes eligible before the withdraw timer expires (due to the reception of a WITHDRAW or HELLO message), it cancels the withdraw timer and remains in active state.

If $R_c \geq 2R_s$, since K-coverage guarantees K-connectivity, the protocol inherently provides both coverage and connectivity guarantees. However, for the case where $R_c < 2R_s$, the authors create a hybrid technique merging CCP with SPAN. This is done by simply combining the eligibility rules for CCP as well as SPAN together, so that both coverage and connectivity concerns are integrated in determining which nodes should form the active topology.

7.3.4 Lightweight deployment-aware scheduling (LDAS)

The LDAS protocol [228] uses a random-voting technique for topology control. The key objective of this technique is to provide probabilistic coverage guarantees without requiring exact location information from neighboring nodes.

There are three states in LDAS as shown in Figure 7.2(d): on-duty (active), ready-to-off, and off-duty. During the on-duty state, the sensor node checks the number of its working neighbors n. If this is larger than the required threshold r (described below), then it sends out penalty tickets to $n - r$ randomly selected active neighbors. A node that receives greater than a threshold number of tickets b goes to the ready-to-off state. In this state a random backoff time between 0 and $W_{\max}$ is used as a timer. If it still has sufficient neighbors at the end of this state, it goes to the off-duty state, erases all tickets, and stays asleep for a timer of duration T_s.

It is assumed that all nodes are placed uniformly in an area with average node density of n. Each node sends a ticket to each neighbor with probability $\frac{n-r}{n}$ if $r \leq n$. The ticket threshold b needs to be chosen so that the remaining average number of nodes after the removal of all nodes with b or more tickets is r. It is hard to estimate b in general, but in low-density settings, it is shown that

$$b \approx (n - r) - \sqrt{2(n - r) \ln \frac{n + 1}{r}} \qquad (7.2)$$

The goal of this algorithm is to ensure that r neighbors are kept awake. Under the assumption of uniform random deployment, bounds are derived in [228] that relate the probability of complete redundancy as well as the average partial redundancy to the number of neighbors. In particular, it is shown that only five neighbors are necessary for each sensor's 90% sensing area to be covered by its neighbors, while having 11 neighbors guarantees, with greater than 90% probability, that its range can be completely covered by its neighbors. These constant numbers of neighbors for coverage that are independent of the total number of nodes in the network N, of course, have to be balanced with the requirement that $O(\log N)$ neighbors are required to ensure network-wide connectivity with high probability, as per the results of Xue and Kumar [232].

7.4 Set *K*-cover algorithms

A completely different approach to the problem of sleep-based topology control for sensor coverage is to formulate it as the following SET K-COVER problem [202]:

Problem 3

Given a collection $C = \{S_j\}$ of subsets of a set S, and a positive integer K. Does C contain K disjoint covers of the set S, i.e. covers $C_1, C_2, \ldots, C_K$, where $C_i \subseteq C$ such that every element of S belongs to at least one member of each of C_i?

7.6 Summary

In WSN applications involving remote surveillance with inexpensive nodes it may be reasonable to over-deploy the network with higher density than needed for basic operation. To extend the lifetime of operation in an over-deployed network, it is necessary to have some mechanism to keep redundant nodes inactive until the active nodes in their neighborhood fail or deplete their energy. This is the key functionality of the sleep-based topology control mechanisms examined in this chapter.

The topology control mechanism must ensure that the network provides adequate connectivity and coverage, despite node failures, so long as sufficient redundancy is available. For unattended operations, it is essential that the topology control mechanism be distributed in nature. Most of the techniques we described in this chapter essentially operate in a similar manner: each node uses local rules and observations to transition between sleep, test, and active states (see Figure 7.1 and Figure 7.2). Some of these techniques focus exclusively on connectivity, such as BECA/AFECA, GAF, CEC, ASCENT, and SPAN, while others, such as PEAS, the sponsored-sector technique, CCP, and LDAS, address coverage issues as well (either in isolation or integrated with connectivity). The set K-cover approaches offer another alternative.

A subtle design issue is that sleep-based topology control protocols must co-exist with any sleep-based MAC protocols that may be implemented on the same network. MAC-level sleep–wake control should be activated only when the node is an active participant of the topology.

Exercises

7.1 *State timeline:* Consider a node in a network running the BECA topology control mechanism. The three timer parameters are set as follows: $T_s = 10$ s, $T_l = 2$ s, $T_a = 3$ s. Assume the node just starts on its sleep mode at time 0 s. There is routing traffic in the network that it can potentially participate in at times 5 s, 11 s, 13 s, and the node has some local data of its own to send at time 19 s. Given this information, draw a time line showing periods when the node is asleep, when it is in listen mode, and when it is in active mode.

7.2 *GAF virtual grid setting:* Prove why setting the side of the virtual grid in GAF to be $r < \sqrt{R/5}$ suffices to ensure that all nodes in adjacent grids (in the four directions) can communicate with each other. What is the bound

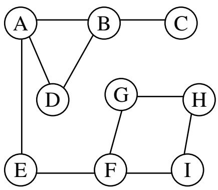

Figure 7.3 A sample communication network graph (for exercise 7.3.)

on r if it is required that nodes should also be able to communicate with neighbors in diagonally adjacent grids (i.e. with the nearest eight grids)?

7.3 *SPAN:* For the communication network graph shown in Figure 7.3, identify a set of nodes such that the nodes could be coordinators when using the SPAN topology control mechanism.

7.4 *PEAS:* Consider a 4×4 grid deployment of 16 sensor nodes with coordinates (0, 0), (0, 1), …, (3, 3). Let us assume that 25% of the deployed nodes are required to wake-up within the first minute for the network to function properly.

(a) Based on the fact that each node sleeps for an exponential distributed duration with probability density function $f(t_s) = \lambda e^{-\lambda t_s}$, what is the desired value of λ_d to ensure that 4 nodes wake-up at least once within the first minute?

(b) Assume that the initial λ of all nodes is set to 0.012 and the probing range $R_p = 1$, what is the expected new value of λ after one round of adaptation based on equation (7.1)? If $R_p = 2$, what is the expected new value of λ?

(c) Simulate the protocol by initializing λ of all nodes as a random variable between 0 to 0.1. Draw the curve of the aggregated λ within 5 minutes with $R_p = 1$ and $R_p = 2$, respectively.

7.5 *Sponsored sector:* Simulate the deployment of nodes one by one randomly in a unit area, with sensing range circles of $R_s = 0.2$, determining whether to activate each one based on the sponsored-sector technique. Plot the number of active nodes with respect to the total number of nodes. At what point does this curve become saturated?

7.6 *LDAS:* Essentially, the purpose of the LDAS topology control mechanism is to keep the average number of active nodes in a neighborhood at r. Consider a simple heuristic where any sensor node goes to sleep with probability $\frac{n-r}{r}$, where n is the number of its working neighbors. Simulate this heuristic in a network with initial average number of neighbors varing from 10 to 30, and compare with a simulation of LDAS based on equation (7.2).

7.7 *Set K-cover:* Consider the simplest randomized Set K-cover algorithm, in which each node allocates itself at random to one of the K-covers. For evaluation, the following coverage metric is to be used: the maximum distance between any active node in the network and its nearest active neighbor, averaged over each cover (note that it is desirable to minimize this metric for superior coverage). Perform simulations in which n nodes are deployed randomly with uniform distribution in a unit area, varying K, showing the performance of the simple randomized algorithm in terms of this maximum distance metric.