The Chinese University of Hong Kong
Department of Computer Science and Engineering

Ph.D. – Term Paper

Title:	Energy-Conserving Coverage Configuration for Dependable Wireless Sensor Networks
Name:	CHEN, Xinyu
Student I.D.:	01409620
Contact Tel. No.:	3163-4253 Email A/C: xychen@cse.cuhk.edu.hk
Supervisor:	Prof. Michael R. Lyu
Markers:	Prof. Jerome Yen (SEEM) & Prof. John C.S. Lui
Mode of Study:	Full-time
Submission Date:	December 12, 2004
Term:	6
Fields:	
Presentation Date:	December 14, 2004(Tuesday)
Time:	2:00–2:45 pm
Venue:	Rm. 1027, Ho Sin-Hang Engineering Building

Energy-Conserving Coverage Configuration for Dependable Wireless Sensor Networks

Abstract

Wireless sensor networks consist of a large number of low-power, short-lived, unreliable sensors. Developing a configuration for sensor sleeping is an effective approach for obtaining a long network lifetime without sacrificing crucial aspects of quality of service (sensing coverage and sensing reliability).

In this report, two sensing models are investigated: boolean sensing model (BSM) and general sensing model (GSM). For the BSM, we present "minimum partial arc-coverage" (MPAC) to exploit sensing arc coverage information provided by the one-hop neighborhood. The MPAC algorithm can deal with the case in which each sensor has different sensing radii while the deployed area preserves its k*-coverage requirement, where* k *is a user defined coverage degree. With the proposed MPAC, three fault-tolerant approaches, adaptive sleeping,* $(k+1)$*-coverage round-based configuration, and configuration with reduced communication radius, are developed. For the GSM, we present "sensibility-based sleeping configuration protocol" (SSCP) by evaluating the proposed neighboring-sensor field sensibility and exploiting the cooperation between neighboring sensors. Pessimistic and optimistic sleeping eligibility conditions are constructed; both are based on the responsible sensing region formed by a Voronoi diagram.*

With the MPAC and SSCP, redundant sensors are optionally identified and scheduled to sleep in order to extend the system lifetime while maintaining adequate sensor redundancy in order to tolerate sensor failures and energy depletions. The proposed MPAC and SSCP are exploited and simulated with ns-2. Simulation results show that there are three effective approaches to build dependable wireless sensor networks: increasing the required degree of coverage or reducing the communication radius, configuring sensor sleeping adaptively, and utilizing the cooperation between neighboring sensors.

Contents

Chapter 1

Introduction

Wireless sensor networks are being increasingly deployed to perform certain tasks, such as sensing, measurement, and surveillance. The sensors, serving as the nodes in this kind of network, are tiny power-constrained devices, which connect together through short-range radio transmission and form an ad-hoc network. The monitoring and surveillance characteristics of a wireless sensor network require that every point in the region of interest should be sensed within given parameters by the cooperation of deployed sensors; otherwise, an event occurring at the under-monitored points will not be detected. This is the coverage issue, one of the fundamental measures for quality of service of a wireless sensor network.

To preserve the coverage requirement, the network should sustain a long lifetime without sacrificing the system's reliability. However, as wireless sensors are microelectronic devices, the energy source provided for them is usually battery power, which has not yet reached the stage for sensors to operate for a long time without recharging or replacement. Furthermore, the unattended nature of sensors and hostile sensing environments make battery recharging or replacement undesirable or impossible [16]. As a result, finding ways to prolong the functional lifetime both of individual nodes and of the network is an important challenge. We know that if a sensor node is frequently alternating between an active and an inactive state, its battery life will be extended [14]. From this observation, sleeping configuration has been proposed as a promising way to extend network lifetime by alternately activating only a subset of sensors and scheduling others to sleep according to some heuristic scheme while providing sufficient coverage in a geographic region.

Besides the coverage problem, sensor nodes may fail or be blocked due to physical damage or environmental interference. The failure of sensor nodes may produce some void areas that do not satisfy the coverage requirement. Therefore, another important design issue is to sustain sensor network functionality without any interruption due to sensor node failure; this is termed the reliability or fault tolerance issue [1, 10]. One way to address this challenging problem is by deploying sensors densely. In a densely distributed sensor network, the system relies on the collective behavior of nodes to function reliably [20], i.e., it is the number, not the capability of each individual node, that really matters. But having too many sensors working at the same time increases the probability of packet collision, thus reducing the network throughput. Therefore, on the one hand, a sleeping configuration protocol should find as many sleeping eligible sensors as possible to prolong network lifetime and to reduce packet collision; on the other hand, it should still retain enough redundancy to ensure dependable sensor networks.

Finally, there is a scalability challenge associated with a high density of nodes when achieving the desired area coverage and robustness. Sensor networks are constructed with multi-hop communications, because generally using several short intermediate hops to send data is more energy efficient than using one longer hop [7, 12]. In addition, communication expends more energy than computation. This implies that each sensor node itself must configure its own operational mode adaptively based on information about its neighborhood, not on the complete information about the deployed region.

In response to all the aforementioned requirements (maintaining coverage, extending system lifetime, fault tolerance, and scalability), we present coverage configuration protocols that are fully decentralized and localized while preserving area coverage and tolerating node failures. Two sensing models, Boolean sensing model (BSM) [15, 17, 19, 20, 21] and general sensing model (GSM) [8, 9, 11] are investigated. The BSM assumes that each sensor has a certain sensing range, and can only detect the occurrences of events within its sensing range; it does not provide any sensibility out of this range. Based on the BSM we propose a sleeping candidate condition called minimum partial arc-coverage (MPAC). The MPAC can deal with sensors with different sensing ranges, and can satisfy k-coverage requirement which indicates that every point in the deployed area is covered by at least k nodes. With the MPAC, we further explore three fault tolerance approaches, adaptive-sleeping, $(k+1)$-coverage round-based node configuration, and configuration with reduce communication radius. Although the BSM model allows a geometric treatment of the coverage problem, it misses the attenuation behavior of signals and ignores the collaboration between adjacent sensors in performing area sensing and monitoring. The GSM captures the fact that signals emitted by a target of interest decay over the distance of propagation. With the GSM, we propose the neighboring-sensor field sensibility (NSFS) to evaluate whether a sensor is eligible to sleep. On the basis of the NSFS, we develop a sensibility-based sleeping configuration protocol (SSCP), which provides dependable configurations to tolerate sensor failures and energy depletions by exploiting the cooperation between neighboring sensors. In order to conserve energy, each node autonomously determines its own status (ON or SLEEPING) by utilizing partial sensor distribution information obtained through communications with its local neighbors. This property enables the MPAC and SSCP to scale to large networks. Simulations with ns-2 [4] show a tradeoff exits between energy conservation, area coverage, and fault tolerance. Three effective approaches to build dependable wireless sensor networks are suggested: increasing the required degree of coverage or reducing the communication radius, configuring sensor sleeping adaptively, and utilizing the cooperation between neighboring sensors.

Chapter 2

Coverage Configuration with Boolean Sensing Model

The Boolean sensing model (BSM) [15, 17, 19, 20, 21] assumes that each sensor has a certain sensing range. A sensor can only detect the occurrences of events within its sensing range and it does not provide any sensibility out of this range. In this chapter, we will develop a sleeping candidate condition based on this sensing model.

2.1 Assumptions and Problem Formulation

When a sensor schedules itself to sleep in order to reduce energy consumption, it should determine whether it satisfies the k-coverage sleeping candidate condition. The k-coverage requires that every point in the deployed area is covered by at least k nodes. Some general assumptions are introduced to help us develop the k-coverage sleeping candidate condition.

Each sensor node N_i knows its own location (x_i, y_i) [15, 19], which can be obtained from the GPS or other localization systems [5]. The nodes discussed in this report are deployed in a two-dimensional Euclidean plane; however, the argument is easily extended to a three-dimensional space. The sensing radius for node N_i is r_i and its Responsible Sensing Region (RSR), denoted by ψ_i, is the area inside its sensing circle, i.e., a point p is in ψ_i if and only if $d(N_i, p) < r_i$, in which $d(N_i, p)$ denotes the Euclidean distance between node N_i and point p. We say a point is covered by a sensor node when this point is in the sensor's RSR. Sensors can communicate directly with their neighboring nodes within transmission radius R.

Definition 1: The one-hop neighbor set of node N_i is defined as

$$N(i) = \{N_j \in \Omega | d(N_i, N_j) \leq R, j \neq i\}$$

where Ω is the sensor node set in the deployment area A.

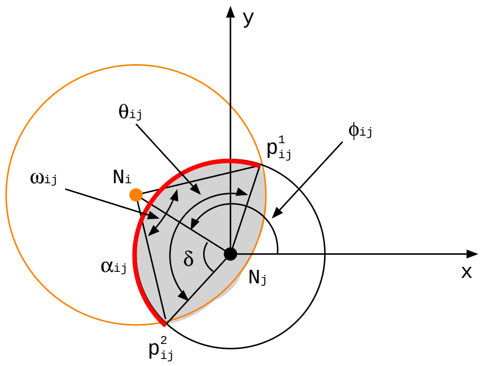

Figure 2.1. Sponsored sensing region, arc and angle and covered sensing angle

We assume that for any $N_j \in N(i)$, $R \geq r_i + r_j$, which simplifies protocol description and avoids routing overhead for any two nodes that sense a common region [19]. This optional assumption ensures that coverage implies connectivity [17, 21].

Definition 2: Suppose nodes N_i and N_j are neighbors, and their RSRs intersect at points p_{ij}^1 and p_{ij}^2 which are arranged in the clockwise order with respect to N_i. As illustrated in Figure 2.1, the part of ψ_i that is also covered by N_j is the shaded region, which is called the *Sponsored Sensing Region* (SSR) by N_j to N_i and equals $\psi_i \cap \psi_j$. The arc $\widehat{p_{ij}^1 p_{ij}^2}$ on the sensing perimeter of node N_j is defined as the *Sponsored Sensing Arc* (SSA), denoted as α_{ij} (shown with a heavy line in Figure 2.1), and its corresponding central angle is called the *Sponsored Sensing Angle* (SSG), denoted as θ_{ij}. Note that the points on α_{ij} are not covered by node N_j, according to the definition of RSR. The direction of node N_i referred to node N_j is denoted as ϕ_{ij}; therefore, $\theta_{ij} = (\phi_{ij} - \delta, \phi_{ij} + \delta)$, in which δ is a half of the central angle θ_{ij}. Note that θ_{ij} expresses the relative position of α_{ij} on the perimeter of ψ_j. In addition, we let $\omega_{ij} = \angle p_{ij}^1 N_i p_{ij}^2$, which is called the *Covered Sensing Angle* (CSG). Actually, $\omega_{ij} = \theta_{ji}$; however, they denote different physical meanings. θ_{ji} is a measure of SSA α_{ji}, whereas ω_{ij} implies which part of the perimeter of ψ_i is covered by node N_j.

From geometry calculations, we know

$$\phi_{ij} = \begin{cases} \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) \\ \qquad : \quad if \;\; x_j - x_i < 0 \;\; \wedge \;\; y_j - y_i < 0 \\ \pi + \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) \\ \qquad : \quad if \;\; x_j - x_i > 0 \\ 2\pi + \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) \\ \qquad : \quad if \;\; x_j - x_i < 0 \;\; \wedge \;\; y_j - y_i > 0 \end{cases}$$

and

$$\delta = \arccos\left(\frac{r_j^2 + d^2(N_i, N_j) - r_i^2}{2 r_j \cdot d(N_i, N_j)}\right).$$

The above definitions apply to the case in which the sensing perimeters of two neighboring nodes have intersection points. To describe the cases in which the sensing perimeters have no intersection points, we introduce some additional concepts as follows.

Definition 3: The number of neighboring nodes whose RSRs completely contain the RSR of node N_i is called the *Degree of Complete Coverage* (DCC) sponsored by neighboring nodes, denoted as γ_i. These neighboring nodes are called *Complete-Coverage Sponsors* (CCSs) of N_i, denoted as $CCS(i)$, and other neighboring nodes are called non-CCSs of N_i.

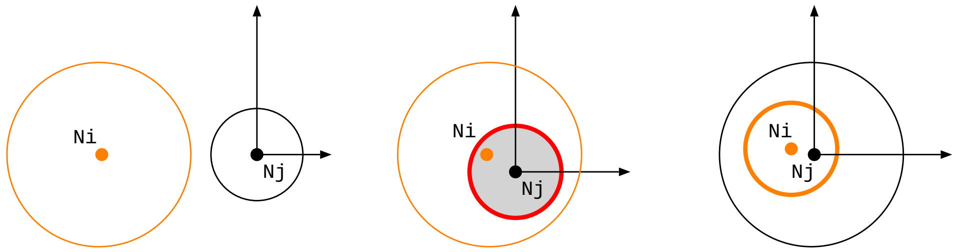

Figure 2.2. Special cases of sponsored sensing region and arc

Now we consider three special cases, which all lack intersection points between two nodes' sensing perimeters, as shown in Figure 2.2. The first one is the case when $d(N_i, N_j) \geq r_i + r_j$. In this case, there is no RSR-overlap between two neighboring sensors and N_j does not provide any coverage to ψ_i. For the second case when $d(N_i, N_j) \leq r_i - r_j$, ψ_j is completely covered by node N_i; therefore, the SSR is all the RSR of N_j, SSA α_{ij} is the perimeter of N_j, and SSG $\theta_{ij} = 2\pi$. However, the CSG ω_{ij} is not defined. This kind of node is called a *Completely Covered Node* (CCN) of N_i, and $CCN(i) = \{N_j \in N(i) | d(N_i, N_j) \leq r_i - r_j\}$ denotes the set of CCNs of N_i. The last case is when ψ_i is completely contained in ψ_j, which happens whenever $d(N_i, N_j) \leq r_j - r_i$ holds. In this case, the SSR, SSA and SSG are not defined; however, the CSG is defined as $\omega_{ij} = 2\pi$. We know that N_j is one of N_i's CCSs; therefore, the set of CCSs of N_i is denoted as $CCS(i) = \{N_j \in N(i) | d(N_i, N_j) \leq r_j - r_i\}$.

Definition 4: The *Minimum Partial Arc-Coverage* (MPAC) sponsored by node N_j to node N_i, denoted as ξ_{ij}, is defined as the number of N_i's non-CCSs covering the point on the SSA α_{ij} that has the fewest nodes covering it.

Figure 2.3 illustrates a calculation of MPAC ξ_{ij}, which is borrowed from [6]. The calculation steps are as follows:

1. Draw a line segment representing [0, 2π];

2. Calculate the SSG θ_{ij}, and mark this angle accordingly on the line;

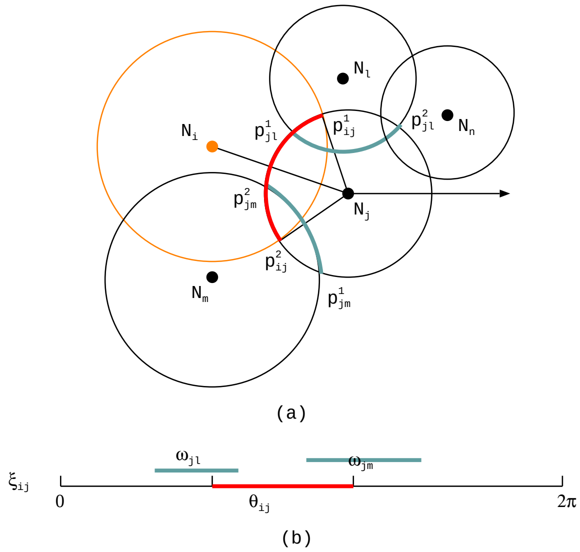

Figure 2.3. Derivation of the MPAC ξ_{ij} sponsored by node N_j to node N_i

3. For each non-CCS of N_i, calculate its CSG related to N_j and lay out the derived CSG proportionally on the line;

4. Scan the line segment θ_{ij} and write down the minimum number that the points on this line segment are covered by the SSG and CSGs.

The derived minimum number is MPAC ξ_{ij}. For the case illustrated in Figure 2.3, we get $\xi_{ij} = 1$. The critical part of SSA α_{ij} is $\widehat{p_{jl}^1 p_{jm}^2}$ as this arc is covered by the minimum number of nodes and determines the MPAC. From the definition and calculation steps, we know $\forall\ N_i \in \Omega$, $\xi_{ij} \geq 1$, because all points on SSA α_{ij} are at least covered by N_i itself.

2.2 MPAC and DCC Based k-Coverage Sleeping Candidate Condition

The statement "a region is k-covered" means every point inside this region is covered by at least k nodes. If the coverage degree of the RSR of a node is at least $(k+1)$, then this node could fall asleep to save energy without sacrificing the network's overall k-coverage requirement. For node N_i, we define its neighboring index set $I_i = \{m | N_m \in N(i)\}$, $m = 1, 2, 3, \ldots$. Then the general k-coverage sleeping candidate condition can be expressed as $\psi_i \subseteq \bigcup_{j \in I_i^n} \psi_j$, in which $n = 1, \ldots, k$, $I_i^n \subseteq I_i$, and $\forall\ 1 \leq p, q \leq k$, $I_i^p \cap I_i^q = \emptyset$ when $p \neq q$. This general sleeping candidate condition is theoretically accurate but cannot be directly employed to select sleeping candidates. We utilize the proposed MPAC and DCC to decide which nodes are sleeping candidates while preserving k-coverage.

Lemma 1: If node N_j is a non-CCS of node N_i, the MPAC sponsored by N_j to N_i is ξ_{ij} iff all subregions, formed by SSA α_{ij} and outside RSR ψ_j, are covered by at least ξ_{ij} non-CCSs.

Proof: The statement "the MPAC sponsored by N_j to N_i is ξ_{ij}" implies that all points on SSA α_{ij} are covered by at least ξ_{ij} non-CCSs. We know that each subregion in ψ_i is formed by SSAs or the perimeter of N_i. All subregions formed by SSA α_{ij} and outside RSR ψ_j contain all points on this SSA due to the definition of the RSR. By continuity of subregion, all these subregions are covered by at most ξ_{ij} non-CCSs.

The subregions formed by SSA α_{ij} and outside RSR ψ_j contain all points on this SSA. If all these subregions are covered by at least ξ_{ij} non-CCSs, all points on this SSA are also covered by at least ξ_{ij} non-CCSs. Therefore, the MPAC sponsored by N_j to N_i is ξ_{ij}. ■

Theorem 1: A sensor node N_i is a sleeping candidate while preserving k-coverage, iff its DCC is greater than or equal to k or for all its non-CCS neighboring sensor nodes N_j, $N_j \in N(i) - CCS(i)$, the MPAC sponsored by node N_j to node N_i is greater than k minus N_i's DCC. Formally, N_i is sleeping-eligible iff $\gamma_i \geq k$ or $\forall$ $N_j \in N(i) - CCS(i)$, $\xi_{ij} > k - \gamma_i$.

Proof: In order for sensor N_i to be a sleeping candidate, all points in its RSR should be covered by at least $(k+1)$ nodes including N_i. If its DCC is γ_i, then all points in its RSR are already covered by at least (γ_i+1) nodes. If $\gamma_i \geq k$, then obviously N_i is sleeping-eligible. If $\gamma_i < k$, we should ensure that the non-CCS neighboring sensor nodes of N_i provide residual $(k - \gamma_i)$-coverage. This is equivalent to proving that if N_i is not sleeping-eligible, there exists a node N_j, $N_j \in N(i) - CCS(i)$, $\xi_{ij} \leq k - \gamma_i$. We know that the boundary of a subregion containing at least one SSA is the necessary condition to form a subregion. Without loss of generality, we assume this SSA is sponsored by non-CCS N_j. Therefore, its MPAC is less than or equal to $(k - \gamma_i)$ by Lemma 1, i.e., $\xi_{ij} \leq k - \gamma_i$.

For the "only if" part, we should prove if N_i is sleeping-eligible, then $\gamma_i \geq k$ or $\xi_{ij} > k - \gamma_i$ for all nodes N_j, $N_j \in N(i) - CCS(i)$. This is equivalent to proving that, when $\gamma_i < k$, there exists a node N_j whose MPAC to N_i is less than or equal to $(k - \gamma_i)$, i.e., if $\xi_{ij} \leq k - \gamma_i$, node N_i is not sleeping-eligible. If $\gamma_i < k$ and $\xi_{ij} < k - \gamma_i$, obviously N_i is not sleeping-eligible. If $\gamma_i < k$ and $\xi_{ij} = k - \gamma_i$, some points on α_{ij} are covered by only k nodes. Note that these points are in RSR ψ_i; however, they are not in ψ_j. If N_i goes to sleep, these points will be covered by at most $(k - 1)$ nodes; then k-coverage will not be preserved. Therefore, when $\gamma_i < k$ and $\xi_{ij} = k - \gamma_i$, N_i is not sleeping-eligible. ■

Theorem 1 can be extended to deal with irregular and/or non-uniform RSRs as long as the RSR of each sensor node can be precisely defined.

2.3 Extended Sleeping Candidate Conditions

When we apply the MPAC and DCC based sleeping candidate condition to schedule nodes' sleeping for a restricted two-dimensional area, more sensors will be identified as sleeping candidates if we take the boundary case into consideration, illustrated in Figure 2.4. We denote the original sleeping candidate condition as *Mpac*, and the sleeping candidate condition which takes the boundary case into consideration as *MpacB*. To determine whether node N_i is a sleeping candidate, we need only consider its RSR inside the sensing area. Therefore, some parts of the original SSA α_{ij} are removed and we must test only arcs $\widehat{p^2_{ij}p^1_j}$, $\widehat{p^2_j p^3_j}$, and $\widehat{p^4_j p^1_{ij}}$. As a result, N_i is not sleeping eligible for 1-coverage with *Mpac*; however, it is a sleeping candidate with *MpacB*.

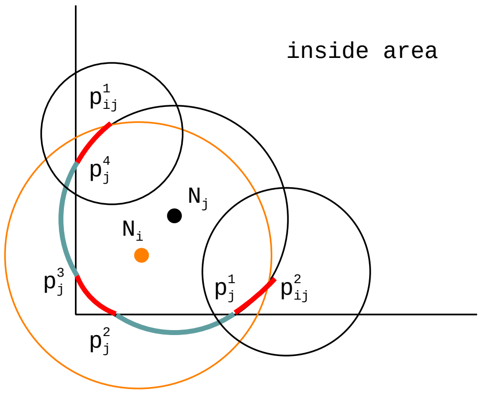

Figure 2.4. Sponsored sensing arcs in boundary case

When each node evaluates its sleeping eligibility with MPAC, it identifies which parts of SSAs in its RSR are critical arcs. A critical arc is the part of an SSA and covered by the minimum number of nodes. These critical arcs form some subregions which do not satisfy the required coverage of degree when the node is set to sleeping and we call them as critical regions. If we could estimate the size to be smaller than a predefined threshold, we may opt to omit this region, as the required coverage degree is still largely satisfied. This may lead to the node becoming sleeping eligible. With this critical region extension added to the model, more nodes can sleep at any given time; however, the cost we must pay is the loss of area coverage. Let *MpacBCa* denote the critical region extension to *MpacB*. We utilize the length of critical arc to estimate the size of its formed critical region. If the length of a critical arc is less than a threshold, we treat this critical arc as a normal arc; when all critical arcs bounding a critical region are treated as normal arcs, the region is no longer considered critical. A special case is when the threshold is *0*, *MpacBCa* is reduced to *MpacB* because no critical arcs are treated as normal.

2.4 Node Scheduling Protocols

Until now, we have introduced three sleeping eligibility conditions, *Mpac*, *MpacB*, and *MpacBCa*. We can apply these conditions for a sensor to evaluate whether it is sleeping eligible or not. However, if we simply schedule all sleeping eligible sensors to turn off their sensing services, void areas will be produced that do not satisfy the required degree of coverage when two compensative sensors go to sleep together. Two sensors are defined as compensative sensors when, if either of them goes into sleeping status, their corresponding RSRs are still covered; however, if both of them are sleeping, a void area must be formed. Therefore, sensors should be carefully coordinated to negotiate which sleeping eligible sensors go to sleep while its compensative sensors keep working. We develop here two decentralized and localized coordination protocols: round-based and adaptive sleeping, to schedule sensors' on and off time properly in order to conserve energy and reduce packet collision whilst also preserving area coverage.

2.4.1 Round-Based Node Scheduling Protocol

The round-based node scheduling protocol divides time into rounds. In each round, every live sensor is given a chance to be sleeping eligible to balance energy depletion between sensors. It requires that sensors should be approximately synchronized [13]. For a sensor to evaluate its sleeping eligibility by applying the MPAC and DCC based k-coverage sleeping candidate condition, it is required to obtain one-hop neighbors information. Letting each node broadcast its own information is a straightforward approach.

Initially, every sensor is in the ON status and sets a round timer T_{round}. When its round timer expires, the node sets its status to UNCERTAIN and enters the on-sleeping decision phase, which contains five steps:

1. Setting a backoff timer T_{hello}, a window timer T_{win}, a wait timer T_{wait}, and the next round timer T_{round}; then starting to collect HELLO messages from neighbors and creating a neighbor list.

2. After T_{hello} times out, broadcasting a HELLO message to all neighbors. The HELLO message conveys node ID, location and sensing radius.

3. After T_{win} expires, evaluating the sleeping eligibility according to sleeping candidate conditions discussed above, using the collected neighborhood information. If eligible, setting status to READY-TO-SLEEPING and broadcasting a STATUS message; otherwise, setting status to ON. The STATUS message contains node ID and current status.

4. When receiving a STATUS message, changing the corresponding sender's status in the neighbor list. If the node's own status is READY-TO-SLEEPING and the received status is READY-TO-ON or the node's own status is READY-TO-ON and the received status is READY-TO-SLEEPING, reevaluating its own sleeping eligibility. If sleeping eligible, setting the status to READY-TO-SLEEPING; if not, setting the status to READY-TO-ON. If this results in a change in the node's status, sending out a STATUS message.

5. After T_{wait} times out, if the status is READY-TO-SLEEPING, setting the status to SLEEPING and entering power saving mode; if the status is READY-TO-ON, setting the status to ON and keeping working.

The corresponding status transition diagram is shown in Figure 2.5. Initially all nodes are in the ON status. The READY-TO-SLEEPING status and timer T_{wait} are intended to avoid compensative neighboring sensors entering

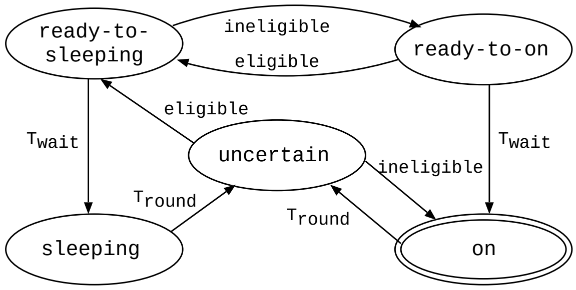

Figure 2.5. Node status transition in round-based scheduling protocol

SLEEPING status at the same time. A sensor in READY-TO-ON may become sleeping eligible if it receives a STATUS message from another sensor that is providing the residual coverage. After the first eligibility evaluation in each round, if a sensor is not a sleeping candidate, we set its status directly to ON. The reason is that even though it may receive subsequent STATUS messages, it cannot be sleeping eligible any more.

The k-coverage sleeping eligibility evaluation procedure for node N_i is shown as follows:

1. for each node N_j, $N_j \in N(i)$, and its status is not READY-TO-SLEEPING
   (a) if ($r_j - r_i \leq d(N_i, N_j)$) then
      i. $k--$;
      ii. if ($k \leq 0$) then break;
   (b) else
      i. for each node N_l whose status is not READY-TO-SLEEPING, $N_l \in N(i) \wedge N_l \in N(j) \wedge l \neq j$, calculate θ_{jl};
      ii. calculate ξ_{ij};
      iii. if ($\xi_{ij} \leq k$) return false;
2. return true;

In this evaluation procedure, each node need only collect its one-hop neighboring nodes information; this is proved in Theorem 2.

Theorem 2: If a sensor node N_i has two neighboring nodes N_j and N_l and their RSRs overlap, i.e., $\psi_i \cap \psi_j \cap \psi_l \neq \emptyset$, then node N_l is one of the neighboring nodes of N_j.

Proof: From $\psi_i \cap \psi_j \cap \psi_l \neq \emptyset$, we get $\psi_j \cap \psi_l \neq \emptyset$, which indicates that $d(N_j, N_l) < r_j + r_l$. With the assumption that for any $N_m \in N(j)$, $R \geq r_j + r_m$, $d(N_j, N_l) < R$, i.e., the distance between N_j and N_l is less than the communication radius. As a result, node N_l is one of the neighboring nodes of N_j. ■

Theorem 2 ensures that collecting one-hop neighboring information is enough to evaluate a node's sleeping eligibility for algorithms which are built on SSG and CSG.

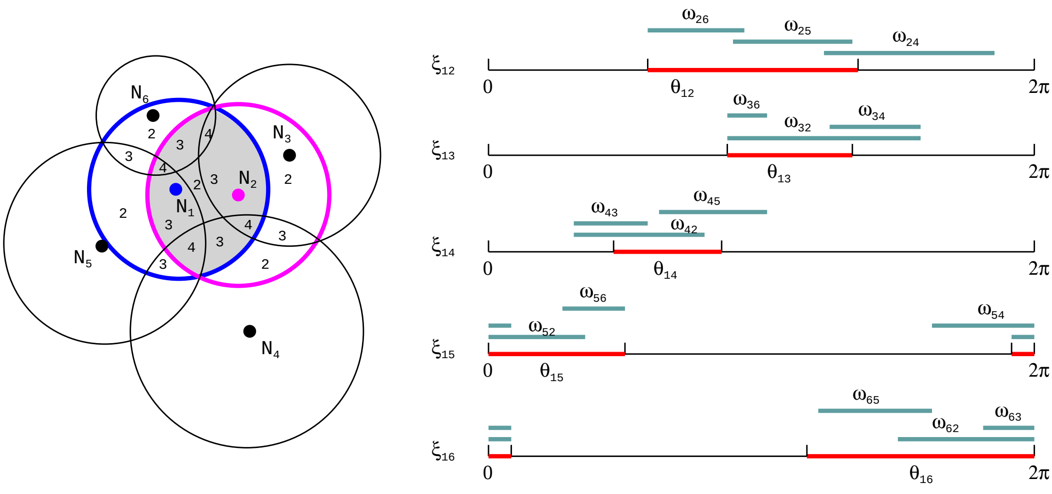

Figure 2.6. An example of sleeping eligibility evaluation

Figure 2.6 illustrates a sleeping eligibility evaluation for node N_1. For all neighbors of N_1, their MPACs are greater than or equal to *2*. Therefore N_1 is 1-coverage sleeping eligible. Additionally, we identify all critical regions for 2-coverage in ψ_1, which are the areas whose coverage degree is *2* in this figure. N_2 is also a sleeping candidate; however, if both nodes are scheduled into sleeping, the part of the shaded area whose coverage degree is *2* will not be covered by any sensors, thus creating a sensing void. This is the reason why a node does not set its status to ON or SLEEPING directly after evaluating its sleeping eligibility. If N_2 learns that N_1 will be scheduled into sleeping for a STATUS message received from N_1, it will be not sleeping eligible anymore, thus preserving area coverage.

When detecting an event, sensors report this event to data sinks. Therefore, the network should be connected to successfully perform its sensing and monitoring task. Considering only the coverage issue when evaluating a sensor's sleeping eligibility may produce disconnected subnetworks, and as a result, even though an event is successfully detected by sensors, this information may not be delivered to the data sinks. To construct an effective sensor network, we must take the communication connectivity into consideration. For any $N_i \in \Omega$, for any $N_j \in N(i)$, $R \geq r_i + r_j$, preserving coverage implies network connectivity, which has been proved in [17, 21]. In this case, no further work need to be done; otherwise we ensure a connected network by evaluating whether its one-hop neighbors will remain connected through each other when N_i is removed.

2.4.2 Adaptive Sleeping Node Scheduling Protocol

A node may suffer failure or deplete its energy. If either of these events occurs when a node in the ON status, some parts of its RSR may no longer satisfy the required coverage degree. Therefore, we should provide a mechanism to detect the loss of area coverage and recover it as soon as possible. The heartbeat message monitoring approach [19] is not suitable in a wireless sensor architecture due to its low power and its intrinsic distributed characteristic. The round-based node scheduling protocol will attempt to recover the area loss in the next round of sleeping scheduling [15]; however, timer T_{round} is a global parameter and not adaptive to recover a local area loss. Instead, we can employ an adaptive node sleeping scheduling protocol by letting each node calculate its sleeping time locally and adaptively to provide fault tolerance in sensor networks.

In the initial stage, all nodes are scheduled in the same way as the round-based protocol. After that, each sleeping node calculates its next wake-up time independently and sets a timer $T_{sleeping}$. When $T_{sleeping}$ times out, the sleeping node N_i wakes up and broadcasts a PROBE message to its neighbors. Each neighbor receiving the PROBE message will return a STATUS message to the sender. The STATUS message contains not only node ID and status, but also residual energy. Then N_i evaluates its sleeping eligibility according to the aforementioned sleeping candidate conditions, such as *Mpac*, *MpacB*, or *MpacBCa*. If N_i is not sleeping eligible, it will set its status to ON and keep working. If it is sleeping eligible, it will calculate its sleeping time. To do this, it generates a random sleeping time, compares this with the minimum remaining working time estimated from residual energy information collected from its neighbors, and sets the smaller of the two as its $T_{sleeping}$. With this adaptive sleeping protocol, each node will wake-up randomly and scan its RSR. If its RSR is not covered due to its neighbors failing or running out of energy, it will switch to the ON status and recover the void area partially or completely. The recovery procedure is gradual, because nodes wake up one by one. Note that, if we wish to detect an area coverage loss early, we must schedule nodes to scan their RSRs more frequently, thus more energy will be consumed.

Chapter 3

Coverage Configuration with General Sensing Model

The BSM model allows a geometric treatment of the coverage problem, it misses the attenuation behavior of signals and ignores the collaboration between adjacent sensors in performing area sensing and monitoring. The general sensing model (GSM) [8, 9, 11] captures the fact that signals emitted by a target of interest decay over the distance of propagation. By engaging the GSM, we exploit the cooperation between adjacent sensors to provide intrinsic fault tolerance to node failures and energy depletion and avoid to produce a single point of failures. To configure a sensor to sleep while preserving area coverage in a decentralized network environment, we should answer three fundamental questions: when we can assert that a region is covered by a set of sensors; what each sensor's responsibility is in providing area coverage; and whether its sleeping will produce any void areas that do not satisfy the coverage requirement.

3.1 Problem Formulation

In general, the ability of a sensor to detect the occurrence of an event of interest at a ceratin point degrades as the distance between the sensor and the point increases. Different sensing models can be constructed to capture the sensing characteristics, which depend on the specific sensor device and the deployment environment. In this report, we employ the following general sensing model, which has been adopted by [8, 9, 11].

Definition 5: In the general sensing model (GSM), the sensibility of a sensor N_i for an event occurring at an arbitrary measuring point p is defined by

$$s(N_i, p) = \frac{\alpha}{[d(N_i, p)]^{\beta}}, \tag{3.1}$$

in which $d(N_i, p)$ is the Euclidean distance between sensor N_i and point p, α is the energy emitted by events occurring at point p, and β is the decaying factor of the sensing signal. For radio signal sensing, β typically ranges from *2* to *5*.

To describe the event sensibility in a region with cooperation of deployed sensors, it is convenient to introduce the concept of the sensing field, a corresponding concept to that of an electric field with a distribution of charges.

Definition 6: Suppose we have a "background" distribution of n sensors, denoted by N_1, N_2, $\ldots$, N_n, in a deployment region, and measure the sensibility on an event occurring at a point p. The sensing field associated with this sensor distribution is defined through the relation

$$S_a(p) = \sum_{i=1}^{n} s(N_i, p), \tag{3.2}$$

in which $S_a(p)$ is called the *All-Sensor Field Sensibility* (ASFS) at point p.
With a sensibility threshold γ, if $S_a(p) \geq \gamma$, we say that the point p is covered. If for every point in the deployment region, its ASFS is greater than or equal to γ, we say that the deployment region is covered.

If we determine a point's coverage based on the ASFS, we need a sink working as a data fusion center, which collects the signal intensities perceived by all sensors. Therefore, directly utilizing the ASFS to evaluate whether a point is covered will produce a heavy network load in multi-hop sensor networks and pose a single point of failures. Applying the fact that radio transmissions are non-directional in wireless sensor networks, we treat each sensor as a sensing fusion center and introduce the following concept.

Definition 7: The *Neighboring-Sensor Field Sensibility* (NSFS) of sensor N_i at point p is defined by

$$S_n^i(p) = s(N_i, p) + \sum_{j=1}^{|N(i)|} s(N_j, p), \quad N_j \in N(i). \tag{3.3}$$

When an event occurs at a certain point, the sensors that receive the signal will broadcast their perceived field sensibility. Each sensor calculates its NSFS after receiving the broadcast messages from its neighbors. If there is at least one sensor whose integrated field sensibility is greater than or equal to the threshold γ, then we say this point is covered. Thus we transform the originally global coverage decision problem into a local decision problem, and avoid producing a single point of failures.

Definition 8: A point p in a deployed region A is covered by a sensor set Ω when there is at least one sensor whose NSFS at point p is greater than or equal to a predefined sensibility threshold γ. Formally, point p is covered if $\exists N_i \in \Omega$, $S_n^i(p) \geq \gamma$. Thus, a deployed region A is covered if all measured points in this region are covered.

The NSFS of node N_i considers only the sensibility provided by its neighboring nodes and itself. Obviously, it is not greater than the ASFS, i.e., $S_n^i(p) \leq S_a(p)$. Note that the coverage problem is intrinsically global in the sense that lack of knowledge of the location of any single sensor implies that the problem may not be solved correctly [12]. But allowing for the possibility of missing some information in the coverage decision provides some redundancy, which is beneficial in building dependable sensor networks. We will identify this property later.

3.2 NSFS-Based Sleeping Candidate Condition

To develop a sleeping candidate condition, we need to define a sensor's responsibility in providing sensing coverage. Here, we employ the Voronoi diagram [2]. The Voronoi diagram, composed of a set of sensor nodes, partitions the deployed region into a set of convex polygons such that all points inside a polygon are closet to only one particular node. Therefore, for sensor N_i, its *Responsible Sensing Region* (RSR) is the polygon in which it resides, denoted by Ψ_i. Because of this property of the Voronoi diagram, if an event occurs in Ψ_i, sensor N_i will receive the strongest signal, thus its NSFS will most likely satisfy the sensibility threshold. In a restricted two-dimensional monitoring region, if a RSR is partly bounded by the boundary of the region, we call it an open RSR; otherwise, we call it a closed RSR.

When a sensor enters the sleeping mode, it does not send and receive messages and does not monitor its environment. As a result, the network topology will be changed and the field sensibility of some regions will be reduced. If we can ensure that there is no void area produced, the sensor can be scheduled to sleep. After it falls asleep, it becomes invisible to all its neighbors, thus the Voronoi diagram will be changed and the RSRs of its neighbors will be enlarged to cover the sleeping node's original RSR. When introducing the concept of NSFS, we have stated that if an event occurs at a certain point, the sensors that receive the signal will broadcast their perceived field sensibility. Therefore, a sensor contributes its sensibility to and only to its one-hop neighbors. No other sensors will receive its broadcasting message. Accordingly, we need only assess its one-hop neighbors' enlarged RSRs when evaluating its sleeping eligibility. If all these RSRs are still covered, then we can say that it is safe to allow a sensor to sleep.

For a sensor to learn its neighbors' RSRs, either it must know the locations of all deployed sensors, or there needs to be a centralized service that calculates every sensor's RSR and dispatches this information to all sensors. Both of these scenarios are undesirable. We need to solve this problem based on local information only; however, computing the neighbors' RSRs exactly with local information only is impossible. Moreover, some redundancy should be kept to tolerate node failures and energy depletion. Therefore, two-hop neighbor information is required, and is sufficient to produce a sensor's local view of the Voronoi diagram composed by its neighbors. In this way, we pessimistically enlarge the RSRs of some neighbors and augment their responsibilities. Figure 3.1(a) illustrates a deployment of sensors and its corresponding Voronoi diagram. From the viewpoint of node N_1, the original Voronoi diagram is changed to Figure 3.1(b), calculated using only its one- and two-hop neighbors' information, after it goes to sleep. The circle represents the one-hop communication area. As nodes N_7, N_8, and N_9 are out of its two-hop reach, N_1 does not know of their existence. We observe that the RSR of N_2 is now partly bounded by the boundary of the monitoring region, thus it is an open RSR. If we directly scan this region, the area near the boundary most likely will not satisfy the sensibility threshold due to long signal transmission distances, which results in N_1's ineligibility to sleep. We reduce the open RSR by confining the scan region within N_1's outermost

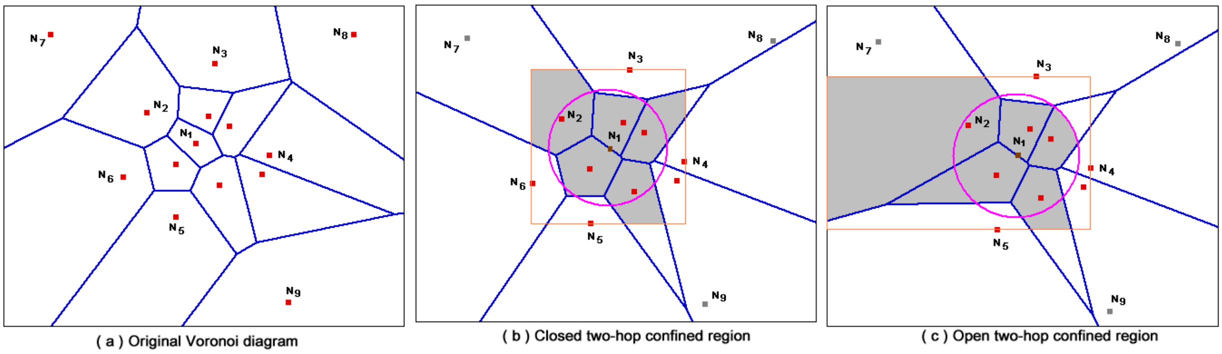

Figure 3.1. Pessimistic scan region for node N_1

two-hop neighbors, shown as the rectangle in Figure 3.1(b). The two-hop confined scan region thus constructed is denoted as a pessimistic scan region because N_1 ignores the sensibilities contributed by out-of-reach sensors and so underestimates the sensibility of this region by using partial sensor deployment information only. Now, some parts of the derived RSRs of N_1's one-hop neighbors may lie outside the two-hop confined region. But at this time sensor N_1 could expect that there are some nodes, which are out of its reach, that are able to provide sufficient coverage for these areas. As a result, the gray area is the scan region that we must check in evaluating N_1's sleeping eligibility. This is a closed two-hop confined region for sensor N_1. If sensor N_6 is not deployed, then the two-hop confined region will be open, as shown in Figure 3.1(c). When N_1 learns that there is no two-hop neighbor beyond its leftmost one-hop neighbor N_2, it must assure that there may be a large gap between its leftmost one-hop neighbor and other sensors or, even worse, that no other sensors exist. In view of this, N_1 pessimistically assumes that N_2 is responsible for all of its left region, and so an open two-hop confined region is constructed.

As the same as the BSM, we also must take the communication connectivity into consideration. However, as tow-hop neighbors' information are collected in the GSM, we relax the connectivity evaluation from one-hop neighbors to one- and two-hop neighbors thus discovering more sleeping eligible sensors. Then, we say that sensor N_i is sleeping eligible if all the RSRs, that are inside its two-hop confined region, of its one-hop neighbors are covered, and its one-hop neighbors will remain connected through each other or through its two-hop neighbors when N_i is removed. We denote this type of path between N_j and N_k as $P_{jk} = N_j N_1 N_2 \ldots N_l N_k$, in which $N_1 \in N(j) - N_i$, $N_2 \in N(1) - N_i$, …, $N_l \in N(l-1) - N_i$, $N_k \in N(l) - N_i$, $l \geq 0$. Obviously, if all N_i's one-hop neighbors are connected, its two-hop neighbors also are connected because these two-hop neighbors are connected with at least one of its one-hop neighbors. If we denote the two-hop confined region of sensor N_i by Υ_i, the sleeping eligibility condition for node N_i can be expressed by

$$\forall N_j \in N(i), \{(\forall p \in \Psi_j \cap \Upsilon_i)\ S_n^j(p) - s(N_i, p) \geq \gamma\} \ \wedge \ \{(\forall N_k \in N(i) - N_j)\ \exists P_{jk}\}. \tag{3.4}$$

Equation (3.4) is called the pessimistic sleeping eligibility condition because it ignores the sensibility contributed by sensors out of N_i's two-hop reach to the RSRs of its one-hop neighbors. As a result, when N_i goes to sleep, the field sensibility of the scan region of N_i will not degrade to less than the sensibility threshold if those more distant sensors fail or run out of energy. Therefore, we say that the pessimistic sleeping eligibility condition provides the ability to tolerate sensor failure and out-of-energy intrinsically. We may relax this sleeping eligibility condition by further confining the scan region within N_i's outermost one-hop neighbors, illustrated in Figure 3.2. In this way, N_i optimistically expects that its sleeping does not degrade the field sensibilities of all

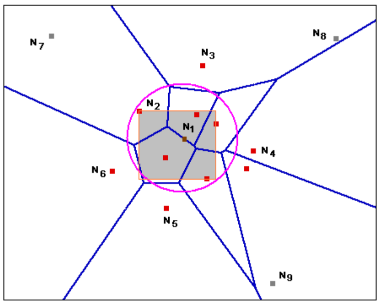

Figure 3.2. Optimistic scan region with closed one-hop confined region for node N_1

those RSRs which are of N_i's one-hop neighbors and outside its one-hop confined region to be less than the sensibility threshold. This introduces a so-called optimistic sleeping eligibility condition. This condition makes more sensors eligible to sleep, thus prolonging the network lifetime; however, it reduces the node redundancy.

3.3 Sensibility-Based Sleeping Configuration Protocol

Until now, we have introduced two, pessimistic and optimistic, sleeping eligibility conditions. We can apply these conditions for a sensor to evaluate whether itself to be sleeping eligible or not. Here we utilize the round-based protocol as described in Chapter 2. The difference is that when applying NSFS-based sleeping candidate condition to evaluate a sensor's sleeping eligibility, not only one-hop neighbors information but also two-hop neighbors information are required. Thus leads to a different acquisition procedure to learn the statuses of other sensors.

Initially, every sensor is in the ON status and sets a round timer T_{round}. When its round timer expires, the sensor sets its status to UNCERTAIN-I and enters the on-sleeping decision phase, which contains eight steps:

1. Setting a backoff timer $T_{hello-i}$, a window timer T_{win-i}, and the next round timer T_{round}; then starting to collect HELLO-I messages from neighbors and creating a neighbor list. The HELLO-I message conveys a node's ID and location.

2. After $T_{hello-i}$ times out, broadcasting a HELLO-I message to all neighbors.

3. After T_{win-i} expires, the HELLO-I messages sent by its one-hop neighbors have been collected, thus a list of its one-hop neighbors has been created. At this time, it prepares to send out this information to its one-hop neighbors. Therefore, its status is changed to UNCERTAIN-II. It sets another backoff timer $T_{hello-ii}$, a window time T_{win-ii}, and a wait timer T_{wait}; then it starts to collect HELLO-II messages, which conveys the information of a node's one-hop neighbors.

4. After $T_{hello-ii}$ times out, broadcasting a HELLO-II message to all neighbors.

5. After T_{win-ii} expires, it learns all its one- and two-hop neighbors. Now it is ready to evaluate its sleeping eligibility according to the sleeping candidate conditions discussed above. If eligible, its status is set to READY-TO-SLEEPING and it broadcasts a STATUS-I message; otherwise, its status is set to ON. The STATUS-I message contains a node's ID, current status and its one-hop neighbors information. The purpose of this message is to alert other compensative sensors to reevaluate their sleeping eligibilities as the sender has changed its status.

6. When receiving a STATUS-I message, updating the corresponding information in the neighbor list, then

   (a) If the node's own status is ON, other sensors' statuses do not affect its own status; however, as two-hop information is needed in our sleeping eligibility condition, the node should forward the received STATUS-I message to its one-hop neighbors. Therefore, it constructs a STATUS-II message, which contains the same information as a STATUS-I message, and broadcasts it.

   (b) If the status is READY-TO-SLEEPING or READY-TO-ON, its status may be affected by the statuses of other sensors; therefore, it reevaluates its sleeping eligibility. If it is sleeping eligible, it sets its status to READY-TO-SLEEPING; if not, it sets its status to READY-TO-ON. If its status is not changed, it sends out a STATUS-II message; otherwise, it sends out a STATUS-I message. Because a STATUS-I message contains the same information as a STATUS-II message does, no further STATUS-II message is required when a STATUS-I message will be sent out.

7. When receiving a STATUS-II message, also updating the corresponding information in the neighbor list, then

   (a) If the node's own status is ON, nothing should be done as other sensors' statuses do not affect its own status.

   (b) If the status is READY-TO-SLEEPING or READY-TO-ON, its status may be affected by other sensors' statuses; therefore, it reevaluates its sleeping eligibility and sets its own status accordingly. If its status is not changed, no further actions are required; otherwise, it sends out a STATUS-I message.

8. After T_{wait} times out, if the status is READY-TO-SLEEPING, setting the status to SLEEPING and entering the power saving mode; if the status is READY-TO-ON, setting the status to ON and keeping working.

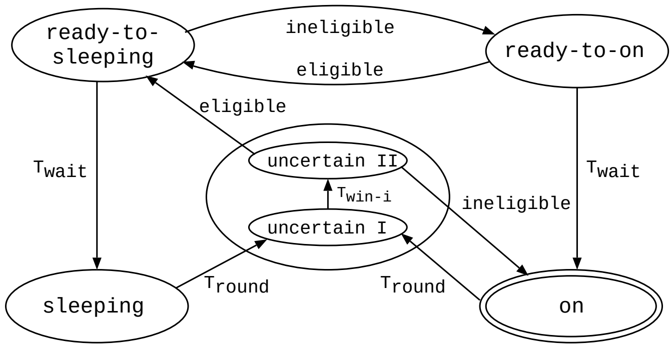

Figure 3.3. Sensor status transition in SSCP

The corresponding status transition diagram is shown in Figure 3.3. A major difference between this figure and Figure 2.5 is that it adds the UNCERTAIN-II status to collect two-hop neighbors information. The UNCERTAIN-I status is the same as the UNCERTAIN status in Figure 2.5 for collecting one-hop neighbors information. A sensor in READY-TO-ON may become sleeping eligible if it receives a STATUS-I or STATUS-II message from a sensor who or whose one-hop neighbors will provide the residual field sensibility. After the first eligibility evaluation in each round, if a sensor is not a sleeping candidate, we set its status directly to ON. The reason is that even though it may receive subsequent STATUS-I and STATUS-II messages, it cannot be sleeping eligible any more.

In this protocol, six timers are employed. Timers $T_{hello-i}$ and $T_{hello-ii}$ are backoff timers for reducing the probability of packet collision. Timers T_{win-i} and T_{win-ii} are intended for collecting HELLO-I and HELLO-II messages, respectively. Timer T_{wait} is a window for compensative sensors to negotiate their cooperative statuses. Timer T_{round} divides the sensors' working time into cycles. At the start of each cycle, all live sensors compete to enter the sleeping mode. Because of this timer, a working sensor may get a chance to be sleeping eligible, thus balancing the sensor's energy consumption and prolonging the lifetime of the network. Moreover, round-based reconfiguration engages a certain inherent immunity to node failures as it will attempt to recover area coverage and network connectivity in the next round's sleeping configuration.

Chapter 4

Simulations and Performance Evaluation

To evaluate and validate the capabilities of our proposed different sleeping candidate conditions and sleeping configuration protocols for the BSM and the GSM, we have implemented them in ns-2 [4] and conducted a simulation study. There is a bridge between the BSM and the GSM. With Equation (3.1) and the sensibility threshold γ, we can define an *ensured-sensibility radius* r_i for sensor N_i as $r_i = \left(\frac{\alpha}{\gamma}\right)^{\frac{1}{\beta}}$, which is equivalent to the sensing radius defined in the BSM. In addition, we evaluate as a baseline the performance of the sponsored sector (SS) eligibility rule proposed by Tian *et al.* [15]. The SS rule considers only the nodes inside the RSR of the evaluated node. These authors also discussed the case of nodes having different sensing ranges, which we call the extended sponsored sector (ESS) rule. We implemented *ESS*, *Mpac*, *MpacB*, and *MpacBCa* in a round-based node scheduling protocol and built an adaptive node scheduling protocol on *MpacB,* denoted *MpacBAs*. Furthermore, we denote the SSCP with the pessimistic sleeping eligibility condition as SscpP and the SSCP with the optimistic one as SscpO.

4.1 Parameters Setting

The deployed sensing area is *50*m$\times$*50*m. Sensors are scattered in this area with a uniform distribution. $\alpha = 1$ and $\beta = 3$. The timer parameters are set as follows: $T_{round} = 100$s; T_{hello}, $T_{hello-i}$ and $T_{hello-ii}$ are uniformly distributed between *0* and *0.5*s; T_{win}, T_{win-i} and T_{win-ii} are set to *1*s; $T_{wait} = 3$s. All the results quoted were obtained from an average of *20* simulation runs. The power consumption of Tx (transmit), Rx (receive), Idle, and Sleeping modes are *1.4*W, *1*W, *0.83*W, and *0.13*W, respectively [3]. The initial energy reserved for each node is uniformly distributed between *200*J and *240*J. The communication radius R is *12*m, the number of deployed sensors is *120,* the coverage requirement for the BSM is the degree of coverage set to be *1* , and the coverage requirement for the GSM is the sensibility threshold γ set to be *0.001* unless specified. To evaluate a sensor's sleeping eligibility with the GSM, we cover its scan region with a virtual square grid [18, 19]. The gird size is *2*m. We select the center of each grid as the sampling point. If every sampling point in sensor N_i's scan region satisfies the sensibility threshold γ, then we say N_i is a sleeping candidate.

4.2 Experimental Results and Discussions

In this section, we present the experimental results from different aspects: communication radius, network topology, deployed sensor number, sensibility threshold, coverage loss, degree of coverage, sensitivity to node failures, and network lifetime.

4.2.1 Sleeping Sensor vs. Communication Radius

Since the neighboring information is shared by broadcasting messages, the communication radius should affect the number of neighbors, and thus impact the percentage of sleeping sensors. Figure 4.1 shows the variation of the percentage of sleeping sensors with the communication radius. As the SSCPs requires two-hop neighboring information, it will be time consuming when the communication radius R is relatively large. We only simulate them when $R \leq 16$m.

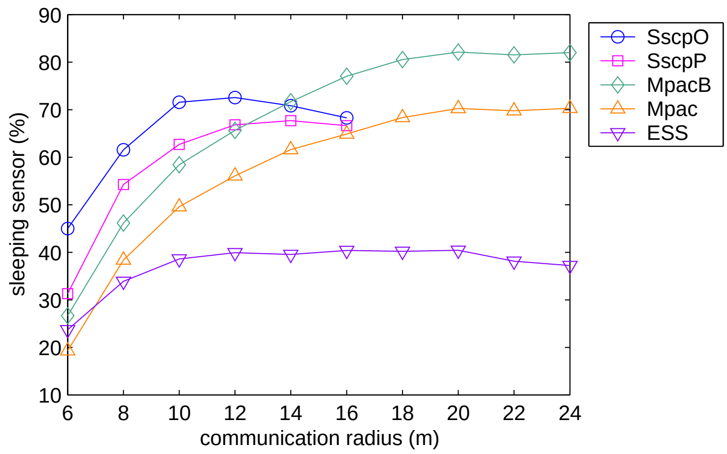

Figure 4.1. Percentage of sleeping sensor vs. communication radius R

When we increase the communication radius, a sensor will identify more adjacent sensors. If a sensor has more neighbors, its RSR is more likely to be covered by its neighbors. As a result, more sensors will be sleeping eligible. However, if we increase the communication radius further, the performance of all protocols tend to degrade. From the construction procedure of the pessimistic and optimistic scan regions, we know that these scan regions are directly related to the communication radius. The larger the communication radius is, the larger the scan regions are. A larger scan region implies that a sensor will get a smaller opportunity to be sleeping eligible. When this negative factor exceeds the advantage brought by the increase in the number of adjacent sensors, the percentage of sleeping sensors will start to decrease. For the *ESS*, we may intuitively expect that the percentage should reach saturation when all the potential for sensors' responsibilities to be covered by their neighbors has been realized. But we should notice that the probability of packet collision is also increased when there is a long communication radius, due to the sharing characteristic of the wireless transmission medium. Consequently, the number of neighbors discovered through the broadcasting approach decreases. Linking this with the fact that a long communication range consumes more energy, we should choose an appropriate communication radius to enable acquisition of adequate neighboring information with as little energy consumption as possible.

When the performances of SSCPs start to decrease or the ESS reaches saturation, the MPACs still identify more sleeping eligible sensors. Because the MPACs investigate not only neighbors in the RSR of the considered node, but also neighbors whose RSRs intersect with the RSR of the considered node. The chose sensibility threshold $\gamma = 0.001$, so the ensured-sensibility radius r is *10*m. Obviously, only when *R=2r=20*m, the performances of MPACs reach the maximum, which is demonstrated by Figure 4.1. Afterwards their performances also start to decrease due to packet collision. As expected, for a restricted deployed area, *MpacB* provides more sleeping eligible sensors than *Mpac* does.

4.2.2 Network Topology

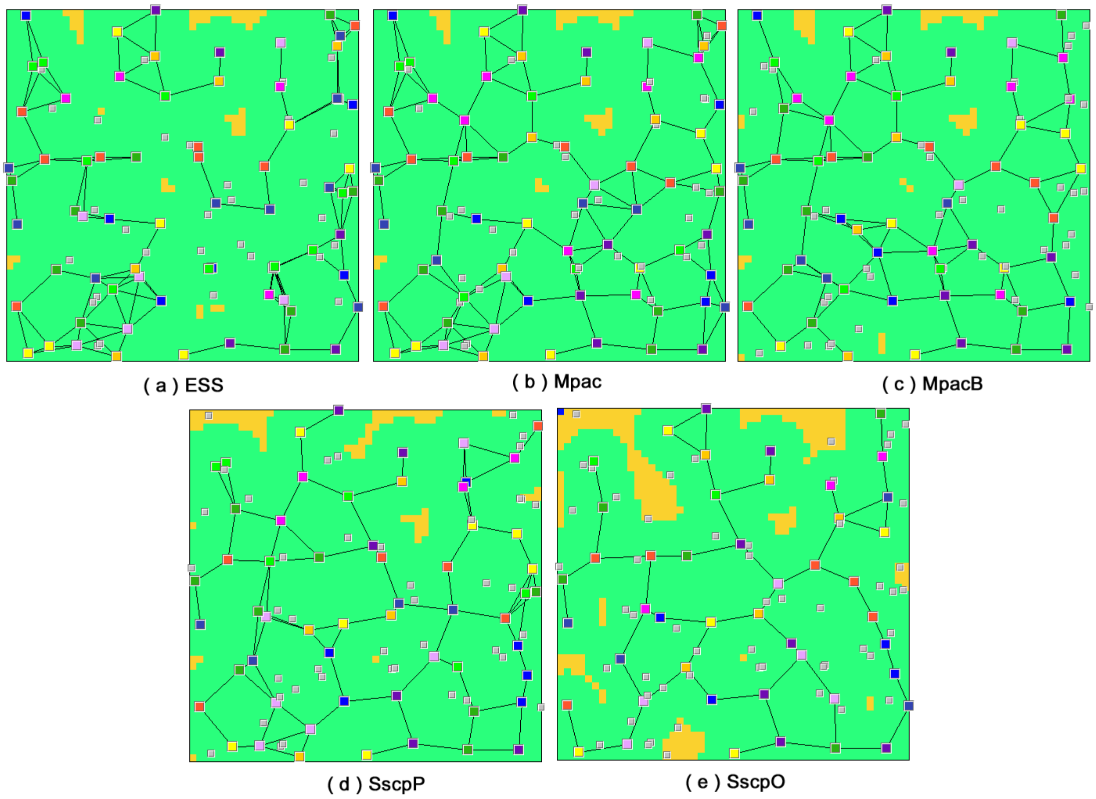

Figure 4.2. Network topologies with R=8m; and the number of deployed sensors is *100*

Figure 4.2 illustrates the resulting network topologies with *100* nodes after applying *ESS*, *Mpac*, *MpacB*, *SscpP*, and *SscpO* to configure sleeping eligible sensors. The small gray squares represent sleeping sensors, and the big colored squares represent working sensors. A line between sensors indicates they are one-hop communication neighbors. The *ESS* produces disconnected subnetworks, while MPACs and SSCPs form a connected network. Another observation is that the marginal sensors are always in working status with the *ESS* and *Mpac*; however, the *MpacB* effectively removes the boundary effect. The SSCPs also allow a lot of marginal sensors to sleep.

4.2.3 Sleeping Sensor vs. Sensor Number

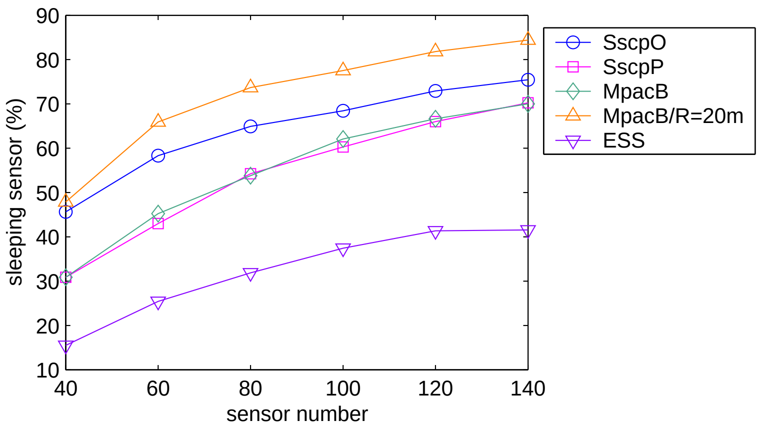

Figure 4.3. Percentage of sleeping sensor vs. sensor number with R=*12*m

Figure 4.3 shows a plot of the percentage of sensors that are asleep as a function of the number of sensors. Obviously, increasing the number of deployed sensors will result in more nodes being sleeping eligible. Our MPACs and SSCPs allow more sensors to sleep than the *ESS* does. While the *ESS* ignores too much sensibility provided by neighboring nodes, the MPACs exploit as much this information as possible, especially when $R \geq 2r$. A sensor itself cannot provide sufficient sensibility for events occurred out of its ensured-sensibility radius; however, through the information exchanged between its adjacent sensors, it can achieve a certain degree of event sensibility. The SSCPs exploit this cooperation property in wireless sensor networks, and thus discover more sleeping eligible sensors than the *ESS* does. As expected, the *SscpO* provides more sleeping eligible sensors than the *SscpP* does, due to the smaller responsibility scan region.

4.2.4 Sleeping Sensor vs. Sensibility Threshold

Figure 4.4 shows how the percentage of sleeping sensors changes with the sensibility threshold γ. If we enhance the sensibility threshold in the GSM (equivalent to decreasing the sensing radius in the BSM), the sleeping

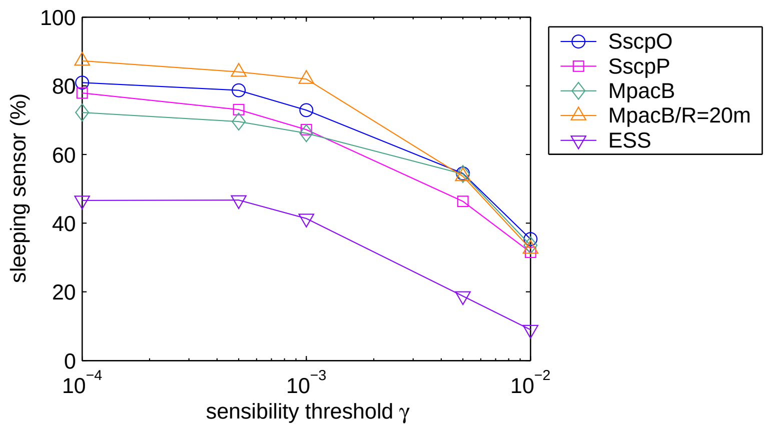

Figure 4.4. Percentage of sleeping sensor vs. sensibility threshold γ

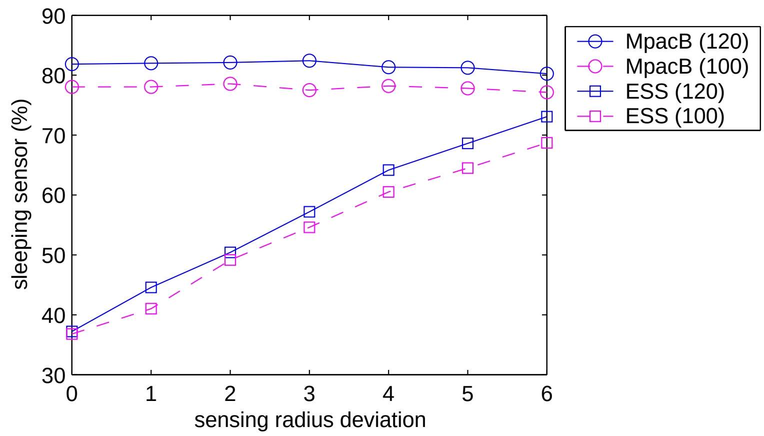

Figure 4.5. Standard deviation of sensing radius

percentage is decreased. The result confirms that, when we demand higher event sensibility, more nodes must be working and fewer nodes are granted the opportunity to sleep.

The sensing radius of a node may change during its lifetime due to the depletion of power, or changing environmental conditions. Figure 4.5 shows how the percentage of sleeping sensors changes with different sensing radii in different sensors. We set the mean of the sensing radius at *12*m, and simulated networks with *100* and *120* sensors. The result reveals that the *ESS* increases the percentage of sleeping sensors significantly as the deviation in sensing radius increases; however, the performance of the *MpacB* varies little. When the deviation of sensing radius becomes large, the sensing area of a node with a small sensing radius will be more likely to be completely or largely covered by other nodes; therefore, more nodes will be sleeping eligible. As the *ESS* initially allows too few nodes to sleep, it benefits most from this phenomenon. In contrast, since the *MpacB* allows more nodes to sleep initially, there is less scope for others to sleep when the sensing radii start to vary.

4.2.5 Coverage Loss of *MpacBCa*

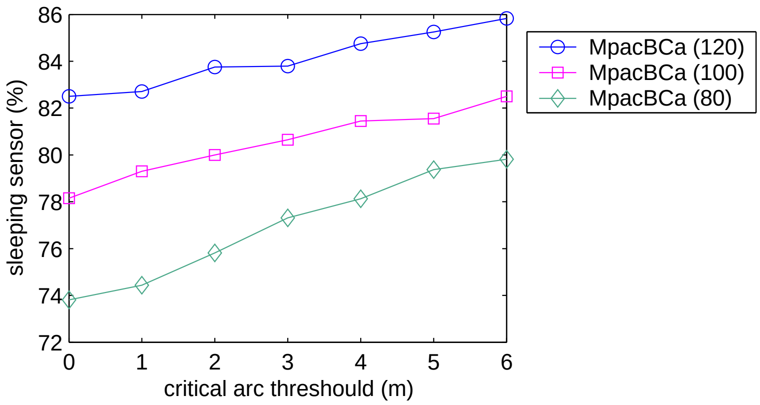

Figure 4.6. MpacBCa sleeping sensor

Table 4.1. Percentage of area coverage loss (%)

	0	1	2	3	4	5	6
MpacBCa (80)	0	0	-0.1	-0.2	-0.5	-1.1	-1.6
MpacBCa (100)	0	0	-0.1	-0.2	-0.6	-0.8	-1.4
MpacBCa (120)	0	0	-0.1	-0.2	-0.5	-1.1	-1.3

The *MpacBCa* is a tradeoff between loss of area coverage and an increase of sleeping sensors. Figure 4.6 shows how the percentage of sleeping sensor vary with the critical arc threshold when the sensing radius is *10*m; the corresponding coverage area losses are given in Table 4.1. These results confirm that our simplified area loss estimation algorithm keeps the area loss at a low level while increasing the percentage of sleeping sensors.

4.2.6 Sleeping Sensor vs. Degree of Coverage

Different applications may demand different coverage degrees for the BSM. Figure 4.7 shows curves of sleeping sensor percentage with different coverage degrees when the sensing radius is *10*m. The *ESS* can only provide *1*-coverage. The percentage of sleeping sensors decreases as coverage degree increases. If a point in the deployed area has to be covered by more sensors, fewer sensors can afford to be in the SLEEPING status.

Figure 4.8 shows the distribution of coverage degree when the required coverage degree is *1*. Without any node scheduling protocols, most areas are covered by *9-15* nodes. The *ESS* takes the mean of coverage degree down to *6*. However, there is still a high degree of sensor redundancy. Our MPAC-Based protocols, *MpacB* and *MpacBCa*, move the average coverage successively closer to *1*. Ideally the average coverage should be as close as possible to k to maximize network lifetime.

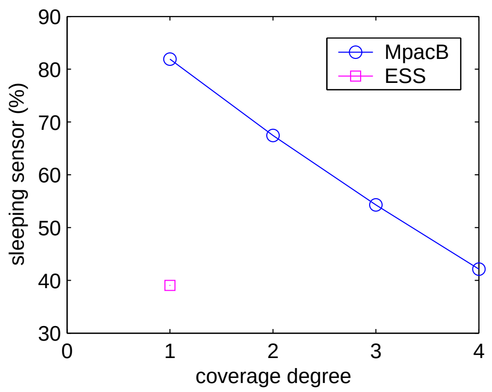

Figure 4.7. Required coverage degree

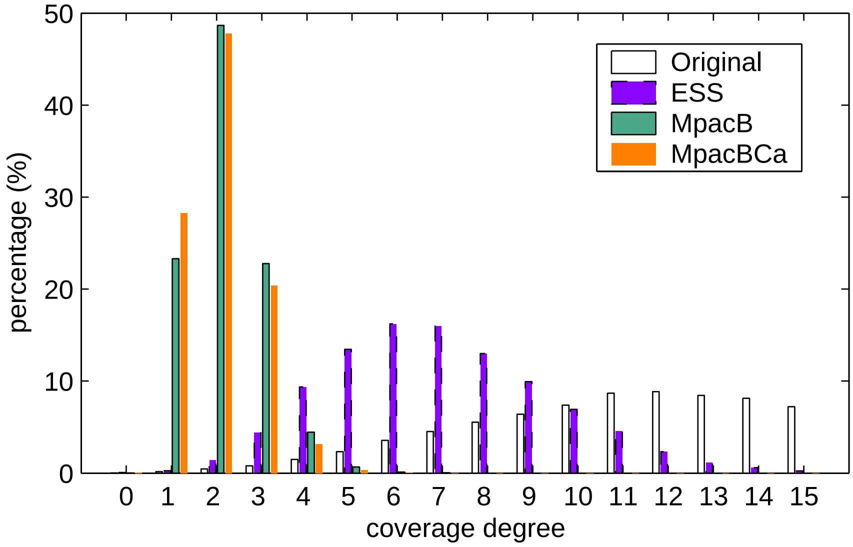

Figure 4.8. Distribution of coverage degree

4.2.7 Network Lifetime and Sensitivity to Node Failures

To simulate failures due to causes other than energy depletion, we assume failures strike sensors according to an exponentially distributed random variable. The mean time between failures (MTBF) in different runs is set between *200*s and *1000*s. The randomly generated sleeping time for the *MpacBAs* follows an exponential distribution with a mean equal to T_{round} for comparison with the round-based protocol. But all generated variables which are greater than $2 \times T_{round}$ will be rounded down to $2 \times T_{round}$. To reduce the k-coverage loss time with the BSM when nodes experience failures and energy depletion, one approach is the aforementioned adaptive sleeping protocol; another approach is that we initially specify $(k+1)$-coverage, which provides one more coverage degree than the design requirement. We set the communication radius as the value at which a coverage configuration protocol achieves its maximum in the percentage of sleeping sensors. According Figure 4.1, we set $R = 12$m for the *ESS*, *SscpP*, and *SscpO*, and $R = 20$m for the *MpacBAs* and *MpacB/2*. As a short communication radius consumes less energy than a long one and provides some node redundancy, we also simulate the *MpacB* with $R = 12$m to observe the relationship between the communication radius and the sensitivity to node failures.

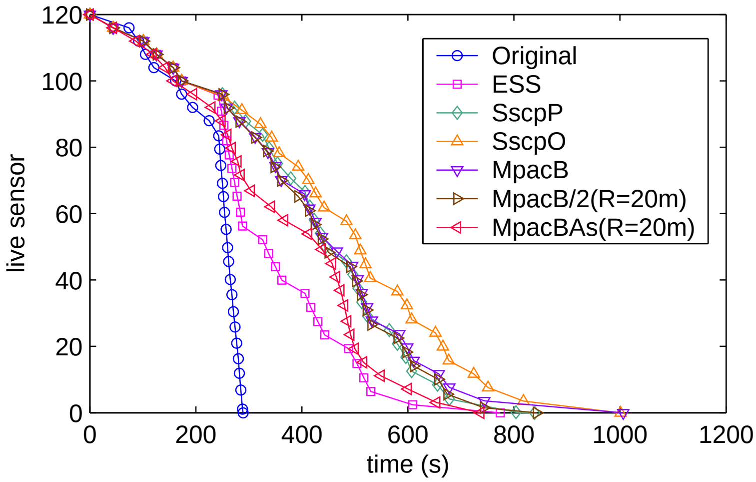

Figure 4.9. Number of live sensor vs. time when the MTBF is *800*s

Figure 4.9 shows the network lifetimes with different protocols when the MTBF is *800*s. The original deployment without any node sleeping configuration performs the worst as it keeps all sensors in working. Although *MpacBAs(R=20m)* keeps the fewest sensors operating at any given time, but adaptively querying the environment consumes more energy thus reducing the network lifetime. The *SscpO* extends the network lifetime most effectively because it obtains the best tradeoff between the energy consumption for sensor coordination and the number of sleeping sensors. As the *SscpP*, *MpacB*, and *MpacB/2(R=20m)* achieve a similar percentage of sleeping sensors, they display similar curves. The *ESS* displays a quicker decrease in the number of live sensors with time.

Figure 4.10 shows simulation results with different MTBFs. The data shown are μ-coverage accumulated time, defined as the total time during which μ or more percentage of the deployed area satisfies the coverage requirement (the degree of coverage for the BSM and the sensibility threshold for the GSM). With the measure of μ-coverage accumulated time, a coverage configuration should maintain as adequate redundancy as possible with the least energy consumption. All μ-coverage accumulated times increase with the MTBF; however, different node scheduling schemes dominate at different μ-coverage accumulated time metrics.

If the deployed area should supply *100*%-coverage as long as possible, the *SscpP* performs the best, especially when the MTBF is relatively large, shown in Figure 4.10(a). The original deployment without sleeping configuration and the *ESS* perform well. Although their network lifetimes are shorter than out proposed protocols, they keep enough redundancy to node failures.

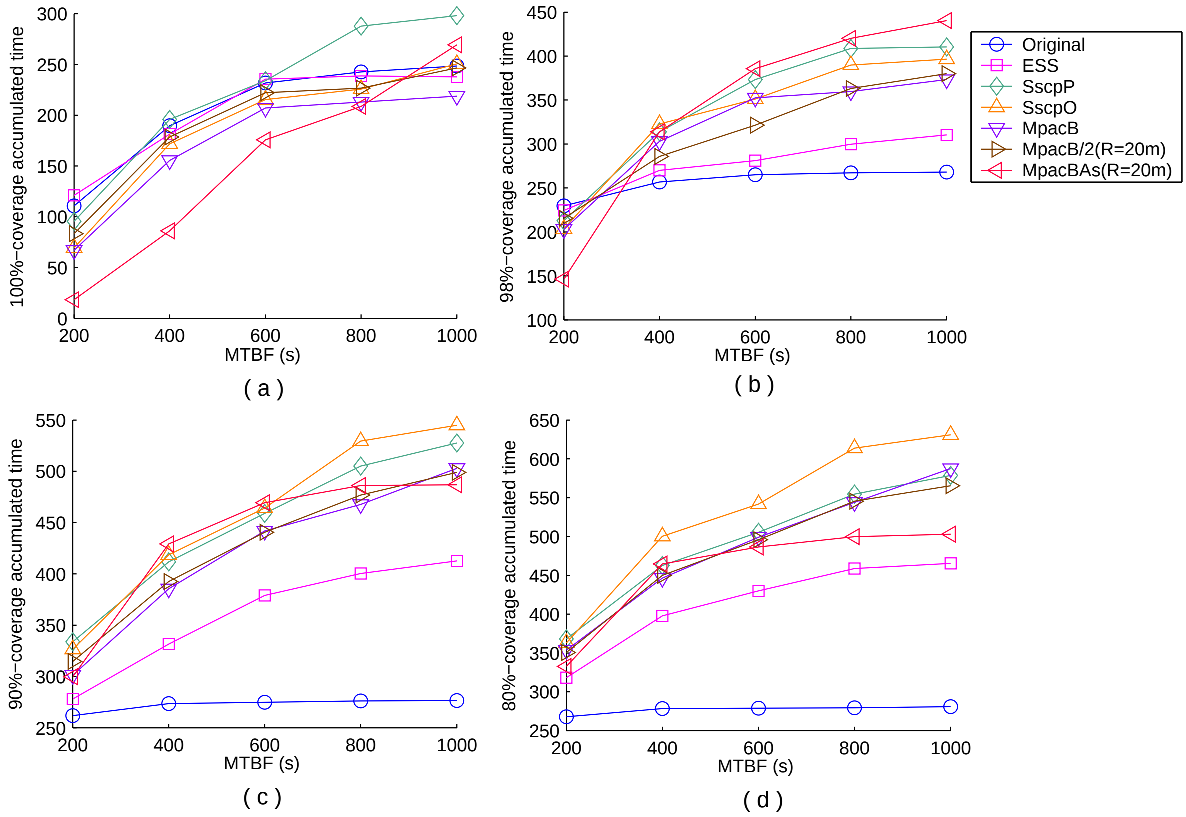

Figure 4.10. μ-coverage accumulated time vs. mean time between failures

When we decrease μ to 98%, the advantages of our MPACs and SSCPs are revealed clearly. The original deployment and the *ESS* are less effective than ours, because too many sensors run out of energy. Figure 4.10(b) confirms that the adaptive sleeping configuration *MpacBAs* recovers area loss faster than round-based protocols. It also shows that the GSM is more reliable than the BSM as the GSM exploits the cooperation between neighboring sensors. If μ keeps decreasing, the protocol which extends the network lifetime most effectively will defeat all others, here it is the *SscpO*.

Taking the results in Figures 4.9 and 4.10 together, we observe that their are three effective approaches to build a dependable sensor network. The *MpacB* and *MpacB/2(R=20m)* display similar performance; therefore, decreasing the communication radius or increasing the coverage degree is equivalent in providing fault tolerance, which is the first approach. Detecting sensor failures and recovering the area loss as quick as possible is the second one, such as *MpacBAs*. Exploiting the cooperation between neighboring sensors is the third one, such as the GSM. We also observe that the cost we must pay for fault tolerance with the BSM is more energy consumption, thus shorter area monitoring time.

Chapter 5

Conclusions and Future Work

This report exploits problems of energy conservation and fault tolerance while maintaining a desired coverage level in wireless sensor networks. We investigate two sensing models: boolean sensing model (BSM) and general sensing model (GSM).

For the BSM, we develop MPAC-based coverage configuration protocols that achieve k-coverage degree and can be applied with different sensing radii. Moreover, three fault tolerance approaches, adaptive sleeping, $(k+1)$-coverage round-based configuration, and configuration with reduced communication radius, are evaluated. For the GSM, we develop a SSCP with two sleeping eligibility conditions: one is pessimistic (*SscpP*) and the other is optimistic (*SscpO*). The proposed protocols effectively identify redundant sensors and coordinate them to sleep to save energy by exploiting the cooperation between adjacent sensors. Moreover, an adequate node redundancy is kept to tolerate node failures and energy depletions. Furthermore, we integrate the sensing coverage requirement with the network connectivity, which results in the network still being connected after sleeping eligible sensors turn off their communication devices.

Our results show that there exists a tradeoff among network lifetime, sensing coverage, and fault tolerance; this varies between different configuration protocols. The communication radius should be appropriately adjusted so that each node can find sufficient neighbors while keeping packet collision at a low level. Three effective approaches to build dependable wireless sensor networks are suggested: increasing the required degree of coverage or reducing the communication radius, configuring sensor sleeping adaptively, and utilizing the cooperation between neighboring sensors.

In the future, we would like to evaluate out proposed coverage protocols under real network traffic. We also will exploit algorithms to identify node redundancy without requiring location information. Additional exploration of network behavior with node failures is important. When does the network partition? How does adjusting the communication radius with different power affect the results? Therefore, we need to build dependable wireless sensor networks both on area coverage and network connectivity.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, pages 102–114, Aug. 2002.

[2] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, Sept. 1991.

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8:481–494, 2002.

[4] K. Fall and K. Varadhan. *The ns manual*, Dec. 2003. http://www.isi.edu/nsnam/ns.

[5] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Transactions on Computers*, 3(8):57–66, Aug. 2001.

[6] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. In *Proc. of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications*, pages 115–121, San Diego, California, Sept. 2003.

[7] G. Khanna, S. Bagchi, and Y.-S. Wu. Fault tolerant energy aware data dissemination protocol in sensor networks. In *Proc. of the 2004 International Conference on Dependable Systems and Networks*, pages 739–748, Florence, Italy, June 2004.

[8] X.-Y. Li, P.-J. Wan, and O. Frieder. Coverage in wireless ad hoc sensor networks. *IEEE Transactions on Computers*, 52(6):753–762, June 2003.

[9] B. Liu and D. Towsley. A study on the coverage of large-scale sensor networks. In *Proc. of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Fort Lauderdale, Florida, Oct. 2004.

[10] M. R. Lyu, editor. *Handbook of Software Reliability Engineering*. IEEE Computer Society Press and McGraw-Hill Book Company, 1996.

[11] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak. Exposure in wireless sensor networks: Theory and practical solutions. *Wireless Networks*, 8:443–454, 2002.

[12] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proc. of IEEE Infocom 2001*, pages 1380–1387, Anchorage, Alaska, Apr. 2001.

[13] J.-P. Sheu, C.-M. Chao, and C.-W. Sun. A clock synchronization algorithm for multi-hop wireless ad hoc networks. In *Proc. of the 24th International Conference on Distributed Computing Systems*, pages 574–581, Tokyo, Japan, Mar. 2004.

[14] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Proc. of 2001 IEEE International Conference on Communications*, pages 472–476, Helsinki, Finland, June 2001.

[15] D. Tian and N. D. Georganas. A node scheduling scheme for energy conservation in large wireless sensor networks. *Wireless Communications and Mobile Computing*, 3:271–290, May 2003.

[16] D. Tian and N. D. Georganas. Location and calculation-free node-scheduling schemes in large wireless sensor networks. *Ad Hoc Networks*, 2(1):65–85, Jan. 2004.

[17] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proc. of the 1st ACM International Conference on Embedded Networked Sensor Systems*, pages 28–39, Los Angeles, California, Nov. 2003.

[18] G. Xing, C. Lu, R. Pless, and J. A. O'Sullivan. Co-Grid: an efficent coverage maintenance protocol for distributed sensor networks. In *Proc. of the 3rd International Symposium on Information Processing in Sensor Networks*, pages 414 – 423, Berkeley, California, Apr. 2004.

[19] T. Yan, T. He, and J. A. Stankovic. Differentiated surveillance for sensor networks. In *Proc. of the 1st ACM International Conference on Embedded Networked Sensor Systems*, pages 51–62, Los Angeles, California, Nov. 2003.

[20] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *Proc. of the 23rd International Conference on Distributed Computing Systems*, pages 28–37, Providence, Rhode Island, May 2003.

[21] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. Technical Report UIUCDCS-R-2003-2351, Univ. of Illinois at Urbana Champaign, June 2003.