Chapter 2
Energy-Efficient Capture of Stochastic Events in Sensor Networks

2.1 Introduction

In this chapter, we consider the *energy-efficient coverage* in a sensor network with an initial random sensor deployment, aiming at capturing stochastic events occurring in the surveillance region [1]. To achieve the goal of energy efficiency, there is a need to duty-cycle the sensors to minimize the redundancy of coverage caused by overlap in their sensing regions. In a *coordinated sleep* approach, we determine when a sensor, is made redundant because its sensing region is completely covered by those of its active neighbors. We can then safely turn off the sensor to conserve energy without hurting the performance. Moreover, it is desirable to rotate the active sensors to achieve energy balance, so that different subsets of the sensors are active at different times. The load balancing prolongs the lifetime of the network before a significant number of the sensors die and cause a severe loss in the coverage [2].

Prior work in coordinated sleep takes a conservative approach. It tries to ensure that every point of the deployment area is covered all the time by at least one active sensor, provided that there is an available sensor in the random placement. The *Role-Alternating*, Coverage-Preserving (RACP) algorithm in [3] is designed to minimize the probability that any given point is not covered by an active sensor, provided that it could be covered. Doing so will also maximize the detection probability of any event and ensure instantaneous detection, if the event is within range of at least one sensor that is alive. In some applications for transient events (e.g., an animal which arrives and then leaves), the goal may be to maximize the detection probability of the events before they disappear, and some delay in the detection is acceptable. Given the relaxed performance objective, leaving an area uncovered some of the time may be acceptable, since an event arriving when there is no active sensor may stay long enough until a sensor becomes active. For simplicity, we can consider periodic scheduling of the individual sensors, in which a sensor is active for only q time every p time ($q \leq p$). It has been shown in [4, 5] that knowledge about the event dynamics—the stochastic processes of the event arrivals/departures—can be used to optimize the periodic schedule of a single sensor for event capture.

S. He et al., *Energy-Efficient Area Coverage for Intruder Detection in Sensor Networks*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-04648-8_2,

Our main contribution in this chapter is to expose the interactions between periodic scheduling and coordinated sleep in a dense static sensor network. We focus on periodic schedules because they can be easily implemented and have been extensively employed with success in diverse settings; in particular, they already allow a wide and systematic spectrum of performance and cost tradeoffs [6]. We will extend the results in [4, 5] to consider energy efficiency and the collective performance of the sensors. The basic observation is that if events may stay for some time, then for a point covered periodically for q time every p time, the fraction of events captured there may be significantly more than q/p. Given the (q, p) schedules of the sensors, the global network itself will achieve the same periodic coverage schedule if the on periods of the sensors are *synchronous* (i.e., they start at the same time). If the on period of each sensor starts at a uniformly random point within p, then the periodic sensor schedules are *asynchronous*. We derive the optimal periodic schedule (q^*, p^*) for both kinds of network. More generally, the size of the sensor synchronization region is specifiable, which gives rise to a spectrum of *regionally synchronous* networks between the synchronous and asynchronous networks. Coordinated sleep by the sensors can then be applied orthogonally to further improve the energy efficiency. We consider the interactions between the periodic scheduling and coordinated sleep, including the cases of (i) synchronous periodic coverage without coordinated sleep (S-nc), (ii) synchronous periodic coverage with coordinated sleep (S-CSP), (iii) asynchronous periodic coverage without coordinated sleep (A-nc), and (iv) asynchronous periodic coverage with coordinated sleep (A-CSP).

Apart from its simplicity, S-nc is mostly interesting as a basis for performance comparison, since its performance is otherwise dominated by that of S-CSP. All the other approaches of S-CSP, A-nc, and A-CSP are of practical interest, and it is instructive to compare their performance with each other and with the RACP protocol in [3]. We have the following findings.

(i) Among the periodic scheduling approaches, S-CSP maximizes the opportunities for coordinated sleep. It can thus achieve the longest network lifetime at the price of some performance loss in event capture. When q is small compared with p, S-CSP is significantly more energy-efficient than RACP, but it is inferior to RACP in terms of capturing events without delay. The performance gap closes significantly, however, when we measure the fraction of events captured with or without delay. S-nc performs better in this measure because it is designed to exploit the possibility of capturing an event before the event leaves.

(ii) A-nc has the least overhead among all the approaches because it requires zero sensor coordination. It has the same network lifetime as S-nc (hence shorter lifetime than S-CSP) but it has better event-capture performance than the synchronous approaches, by spreading out the redundant coverage for a higher global coverage intensity. Under a wide range of q/p, its performance in terms of event capture (with or without delay) is extremely competitive with that of RACP and it has a longer network lifetime than RACP. When the sensor density is high, it is extremely competitive with RACP in terms of *instantaneous* event

capture also, while remaining more energy efficient than RACP. Hence, periodically resting the sensors to conserve energy does *not* necessarily add noticeable delay in the event capture. Moreover, this extremely high performance for *instantaneous* event capture is by nature independent of the event dynamics.

(iii) A-CSP achieves the same event-capture performance (instantaneous or delayed) as A-nc (hence the same competitive performance with RACP), but it can further extend the network lifetime beyond A-nc. When the sensors' periodic schedules are asynchronous, however, the chance for coordinated sleep decreases. A higher sensor density will then help A-CSP realize its potential for energy savings, provided *also* that q is large enough relative to the energy costs of turning off/on the sensor. More generally, as the size of the synchronization region decreases, the event capture performance increases while the opportunities for coordinated sleep decrease. Our performance results show that *if the sensor density is high and both instantaneous and delayed capture performance are important, A-CSP represents the overall best method.*

The rest of the chapter is organized as follows. We formulate the problem of energy-efficient event capture in Sect. 2.2. We present results on event capture by a periodic sensor in Sect. 2.3, based on which the optimization of synchronous and asynchronous periodic schedules is derived in Sects. 2.4 and 2.5, respectively. We propose a general paradigm of regionally synchronous networks in Sect. 2.6. A coordinated sleep protocol under periodic scheduling is presented in Sect. 2.7. We present extensive simulation results in Sect. 2.8 for performance evaluation, and conclude in Sect. 2.9.

2.2 Problem Setup and Performance Metrics

We assume that sensors are used to cover an area for capturing stochastic events. We use the perfect disc sensing model, meaning that an event is captured if its distance from an active sensor is less than distance r, where r is the sensing range. The perfect disc sensing model is widely used in the literature due to its simplicity [7, 8]. While more complex anisotropic models [9] could be used for increased accuracy, doing so will not change our conclusions qualitatively. We assume that sensors can be turned off (made inactive) independently to conserve energy. Strictly speaking, each sensor has three main functional modules—sensing, communication, and computation—which can be turned off independently. For simplicity, we will assume that when the sensor becomes inactive, all three functional modules are inactive. According to specifications of real-world sensors such as Crossbow motes [10], the sensor in the active and inactive states consumes energy at rates of k_1 and k_2, respectively, where k_1 and k_2 are constants. Moreover, a constant amount of energy, given by c, is needed for the sensor to change between the active/inactive states.

We assume that transient random events appear at given *points of interest* (PoIs). The average rate of event arrivals at a PoI is given by λ. After an event arrives at PoI i, it stays for a random *event staying time* drawn from a distribution X. After that, the event disappears and after another random *event absent time* drawn from a distribution Y, the next event arrives at i. We refer to the statistical characteristics of the event staying and absent times as the *event dynamics*. We assume that the staying time and the subsequent event absent time of different events are i.i.d., even though the two quantities for the same event may be dependent. We do not exploit the spatial correlation of PoIs in the network design. Hence, we assume that the event dynamics at different PoIs are independent. This model of event dynamics applies to many application scenarios in event detection. One example is the characterization of event dynamics in mobile sensor networks [4]. Another example is spectrum detection in cognitive sensor networks [11]. A primary user may occupy a channel for some time period, after which the primary user may leave the channel, and so on.

We consider random placement of the sensors according to a Poisson point process of intensity γ. The Poisson point process is widely used in the analysis of random node placements in networks [12]. When γ is large, there is significant redundancy between the sensing regions of sensors. A sensor S is made redundant by its active neighbors if its sensing region is completely covered by those of the neighbors. Moreover, we assume that there is a uniform γ for the whole network for simplicity. If the sensor density varies, our analysis can be applied to each of the varying parts in a straightforward manner.

In quantifying the performance of the sensor network, we restrict our attention *only* to PoIs that are within distance r of at least one sensor. It is because if events happen at PoIs not within range of any sensor, the events cannot be captured irrespective of the network design,[1] and we consider these events out of scope. We use two principal performance metrics: (i) the probability of instantaneous capture, P_{in}, which is the probability that an event arriving at a PoI will be captured immediately upon arrival (without delay); and (ii) the probability of capture, P_c, which is the probability that an event appearing in a PoI will be captured before it leaves the PoI (though the event does not necessarily leave the network). It is clear that $P_c \geq P_{in}$. In many applications, some delay in the detection is acceptable. In a wildlife monitoring network, for example, researchers may be interested in identifying the animals that pass by a given point, but they do not need immediate report of the information. Also, when events are detected with delays, the average delay can be further quantified for performance evaluation.

[1] Since we do not control the sensor placement.

2.3 Event Capture by Periodic Sensor

We now analyze the per-PoI event capture performance of a sensor on a periodic schedule (q, p). The periodic schedule means that before energy runs out, the sensor is alternately active and inactive for q time and $p - q$ time, respectively. The following theorem concerns the probability that an event is captured by the sensor and is a paraphrase of Theorem 2 in [5].

Theorem 2.1. *For a (q, p) periodic sensor whose sensing region covers PoI i, the sensor captures an event at i (before the event disappears) with probability*

$$P_c = \frac{q}{p} + \frac{1}{p}\int_0^{p-q} \Pr(X \geq t)\, dt$$

before it runs out of energy.

Proof. Refer to [5], Theorem 2.

Note that the expression for P_c is a sum of two terms. The first term gives the fraction of events captured during the sensor present period $[0, q]$. It is equivalent to the performance measure P_{in} since these events are captured instantaneously. The second term gives the fraction of captured events that arrive during the sensor absent period $[q, p]$, but stay long enough to be captured during the *next* sensor present period $[p, p+q]$. These events are captured with a delay, and therefore contribute to the performance measure P_c. The main observation is that due to the contribution of the second term, the sensor working for q time every p time may capture a fraction of events that is much higher than q/p. This property holds true as long as the events stay, as they typically do in real applications, and does not depend on the detailed event dynamics. We intend to exploit this property of the periodic sensor to achieve high energy savings with relatively minor loss in the capture performance.

We can analyze the expected delay for those events captured with a delay, as well as for all the captured events, as given by the following theorem.

Theorem 2.2. *For the events captured at PoI i that arrive during a sensor absent period, the expected delay until their capture at i is given by*

$$D_{del} = \frac{\int_0^{p-q} \Pr(X \geq t) \times t\, dt}{\int_0^{p-q} \Pr(X \geq t)\, dt}.$$

For all the events captured at the PoI i, the expected capture delay is given by

$$D_{tot} = \frac{\int_0^{p-q} \Pr(X \geq t) \times t\, dt}{q + \int_0^{p-q} \Pr(X \geq t)\, dt}.$$

Proof. An event arriving at t time before the next sensor present period will be captured with delay t provided that it stays long enough. Since the event must arrive during the current sensor absent period, the value of t ranges from 0 to $p-q$, and $\int_0^{p-q} \Pr(X \geq t)\, dt$ is the probability that the event stays long enough. Therefore, $\int_0^{p-q} \Pr(X \geq t) \times t\, dt$ is the average delay of all the captured events. $\int_0^{p-q} \Pr(X \geq t)\, dt$ expresses the probability that an event occurs during the absent period. Hence, $\frac{\int_0^{p-q} \Pr(X \geq t) \times t\, dt}{\int_0^{p-q} \Pr(X \geq t)\, dt}$ gives the expected capture delay of the events that arrive during the absent period. A similar computation yields the value of D_{tot}.

Theorem 2.1 applies to general distributions of the event staying times. We can illustrate the result using the Exponential distribution with rate parameter α, i.e., the pdf of X, denoted by $f(x)$, is given by:

$$f(x) = \alpha e^{-\alpha x}, \quad x > 0, \quad \text{mean} = \frac{1}{\alpha}.$$

Then we have

$$P_c[X \in \text{Exponential}(\alpha)] = \frac{q}{p} + \frac{1 - e^{-\alpha(p-q)}}{p\alpha}. \tag{2.1}$$

Similarly, the delay values in Theorem 2.2 specialize to

$$D_{del}[X \in \text{Exponential}(\alpha)] = \frac{1 - e^{-\alpha s}(\alpha s + 1)}{\alpha(1 - e^{-\alpha s})}, \tag{2.2}$$

$$D_{tot}[X \in \text{Exponential}(\alpha)] = \frac{1 - e^{-\alpha s}(\alpha s + 1)}{\alpha(\alpha q + 1 - e^{-\alpha s})}, \tag{2.3}$$

where $s = p - q$.

2.4 Energy-Aware Optimization of Synchronous Periodic Schedule

We discuss optimization of the periodic schedule (q, p) for the *synchronous* network. In such a network, all the sensors employ the same (q, p) schedule, and they start their on periods at the same time so that the on/off periods are synchronized. Thus, the global network as a whole behaves like one big sensor on the (q, p)-periodic schedule. In practice, lightweight and accurate time synchronization protocols are available [13] to support the implementation of the synchronous network.

It can be shown that, for general distributions of X, the P_c value in Theorem 2.1 has two properties: (i) for the same p, P_c is monotonically increasing in q/p; (ii) for the same q/p, P_c is monotonically decreasing in p. Hence, to optimize the event

capture, we can either make $\frac{q}{p} = 1$, or if we decide to use a smaller q/p, we can make p as small as possible. The first option will not help us save energy, while the second option is limited physically by the delay and energy expense in switching the sensor between the on/off states frequently.

To optimize the periodic schedule, we need to explicitly account for energy use by the energy model given in Sect. 2.2.[2] Specifically, we know that for a (q, p)-periodic sensor, it is able to capture a fraction P_c of events according to Theorem 2.1. Hence, its per-PoI rate of capturing events is given by $Q' = \lambda \times P_c$. On the other hand, the rate of energy use of such a sensor is given by

$$E' = \frac{k_1 \cdot q + k_2(p-q) + 2c}{p}. \tag{2.4}$$

We allow the user to specify the minimum P_{in}, denoted by $P_{in}^{\min}$, that events are detected without delay, where $P_{in}^{\min} > 0$. By Theorem 2.1, P_{in} is equal to q/p. Hence, we set $\frac{q^*}{p^*} = P_{in}^{\min}$ and optimize P_c by solving the following optimization problem for the *per-PoI number of events captured per unit of energy*, denoted by Q_E:

$$\text{Find } p^* = \arg\max_p \ Q_E = \arg\max_p \ Q'/E'.$$

(Note that the above formulation is equivalent to optimizing the expected number of events captured before an energy budget given by B is depleted.) For example, if $X \in \text{Exponential}(\alpha)$ and $Y \in \text{Exponential}(\beta)$, then $\lambda = \alpha\beta/(\alpha+\beta)$. By (2.1), the rate of capturing events at a PoI is

$$Q' = \frac{\beta}{p(\alpha+\beta)}(1 + q\alpha - e^{-\alpha(p-q)}). \tag{2.5}$$

The optimization is then to find p^* that maximizes

$$Q_E = \frac{\beta}{\alpha+\beta}\frac{1 + q\alpha - e^{-\alpha(p-q)}}{k_1 \cdot q + k_2(p-q) + 2c}, \tag{2.6}$$

where $q = P_{in}^{\min} \times p$. Note that we have a one-dimensional optimization problem, and the solution can be computed numerically by comparing the points of p where

$$\frac{dQ_E}{dp} = 0 \text{ and } \frac{d^2Q_E}{dp^2} < 0.$$

[2]For simplicity, we do not consider the latency constraints of turning on/off the sensor, but note that such constraints can be easily incorporated.

Remark. Note that in the network, more than one PoI may be within range of a sensor, and the same PoI may be within range of more than one sensor. Hence, the sensors may not capture events that are distinct. Assume that there are m distinct PoIs within range of n sensors in the network. The number of distinct events captured per unit energy for the whole network is $\frac{m \times Q'}{n \times E'}$. This value is the Q_E value derived above scaled by a constant factor of $\frac{m}{n}$. The scaling by a constant factor will *not* affect the solutions to the optimization problem.

2.5 Optimization of Asynchronous Periodic Schedule

We now analyze the *asynchronous* network. In this kind of network, each sensor employs the same (q, p)-periodic schedule, but they start their on periods independently at a uniformly random point in time within the period p. Because the on periods of the sensors are spread out in the asynchronous network, the event capture performance must consider the joint operation of the sensors. We have the following main results (Theorems 2.3 and 2.4).

Theorem 2.3. *For a random placement of sensors by the Poisson point process of intensity γ, the probability that an event appearing at a PoI is captured instantaneously is given by $P_{in} = \frac{1-e^{-\gamma\pi r^2 \frac{q}{p}}}{1-e^{-\gamma\pi r^2}}$, where r is the sensing range.*

Proof. A sensor can potentially detect an event if the event happens within distance r of the sensor. Hence, an event may be potentially detected by any sensor within the circular region centered at the event and of radius r. Note that at least one such sensor exists, by the definitions of our performance metrics P_{in} and P_c. By the property of the Poisson point process, the probability p_k that there are k sensors in the circular region is given by

$$p_k = \frac{(\gamma\pi r^2)^k e^{-\gamma\pi r^2}}{k!}. \tag{2.7}$$

The probability that any sensor is inactive when the event happens is $1 - \frac{q}{p}$. The event is undetected on arrival only if all the k sensors are inactive, which happens with probability $(1 - \frac{q}{p})^k$. Hence, summing over the range of k, we can compute the probability that the event is undetected on arrival as

$$\frac{1}{1-p_0} \sum_{k=1}^{\infty} (1 - \frac{q}{p})^k \, p_k = \frac{e^{-\gamma\pi r^2 \frac{q}{p}} - p_0}{1-p_0}.$$

The result follows as the complement of the above probability.

Remark. Whereas $1 - P_{in}$ decreases linearly with $\frac{q}{p}$ in the synchronous network, the decrease is exponential in the asynchronous case. This implies that for the

asynchronous network, a small increase in $\frac{q}{p}$ may result in a large increase in P_{in}. We will show in Sect. 2.8.2 that the asynchronous network can indeed achieve high energy efficiency with an extremely small loss in event capture performance. Also, when the network is configured to achieve an extremely high instantaneous capture probability, its excellent performance is in fact *not* dependent on the event dynamics.

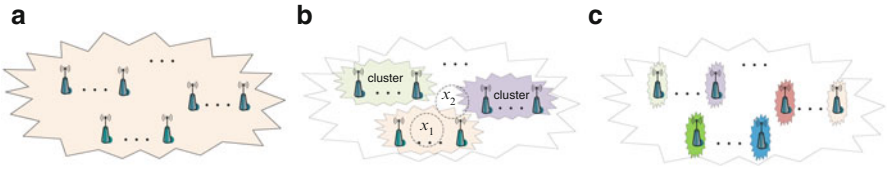

Fig. 2.1 Illustration of the synchronous, regionally synchronous, and asynchronous networks. (**a**) Synchronous network. (**b**) Regionally synchronous network. (**c**) Asynchronous network

Theorem 2.4 ($X \in$ **Exponential**(α)). *For a random placement of sensors by the Poisson point process of intensity γ, the probability that an event appearing at a PoI is captured before it leaves the PoI is given by $P_c = \frac{1-e^{\gamma\pi r^2(\rho-1)}}{1-e^{-\gamma\pi r^2}}$, where r is the sensing range and $\rho = \frac{\alpha(p-q)-1+e^{-\alpha(p-q)}}{p\times\alpha}$.*

Proof. Let $s = p - q$. Consider a sensor in the circular region of radius r centered at the event. The sensor starts its most recent off period at time 0. Consider an event arriving at time t and staying for time x, where $x \sim X \in$ Exponential(α). Without loss of generality, we assume that the start time t of the event is within the period $[0, p]$. The sensor does not detect the event if it is inactive in $[t, t + x]$, which occurs with probability $\Pr(x < s - t)$ provided that $t < s$. Hence, summing over the range of t and denoting by ρ the probability that the sensor does not detect the event before the event leaves, we have

$$\rho = \frac{1}{p}\int_0^s 1 - e^{-\alpha(s-t)}\,dt = \frac{\alpha s - 1 + e^{-\alpha s}}{p \times \alpha}.$$

For the Poisson point process, (2.7) gives the probability that there are p_k sensors within range of the event. The probability that the event is undetected by any in-range sensor before it leaves is therefore

$$\frac{1}{1-p_0}\sum_{k=1}^{\infty} p_k\,\rho^k = \frac{e^{\gamma\pi r^2(\rho-1)} - p_0}{1-p_0}.$$

The result follows as the complement of the above probability.

According to Theorem 2.3, P_{in} does not depend on the event dynamics X and Y. From Theorem 2.4, P_c only depends on the event staying time X, which is only used to calculate ρ. Similar results can be obtained for other distributions of X. For ease of exposition, we will present results for Exponential event staying times only.

As with the synchronous network, we allow the user to specify the minimum probability $P_{in}^{\min}$ of instantaneous event detection. Notice that P_{in} is monotonically increasing in both the sensor density γ and the ratio q/p. When $q = p$, $P_{in} = 1$. Hence, any P_{in} is satisfiable. We can compute the smallest q/p needed to satisfy $P_{in}^{\min}$ and denote this value by z. Once z is determined, we can find p^* (hence $q^* = z\,p^*$) as the optimal p that maximizes $Q_E = \frac{P_c \times \lambda}{E'}$. As in the synchronous network case, we have a one-dimensional optimization problem, although the expression for P_c is more complex.

Note also that if we considered the capture probabilities of *all* events (i.e., whether they fall within range of at least one sensor or not), the above Q_E measure would be scaled by $1 - p_0$, which is independent of p and q and hence will not affect the optimization problem.

2.6 General Regionally Synchronous Networks

We have discussed the design of synchronous and asynchronous networks. In a synchronous network, the sensors are network-wide synchronized, i.e., they begin their on periods at the same time (Fig. 2.1a). In an asynchronous network, the sensors do not synchronize but start their on periods independently (see Fig. 2.1c). The asynchronous network outperforms the synchronous network in terms of P_{in} and P_c. However, when neighboring sensors may also coordinate to eliminate their coverage redundancy (the design of the supporting coordinated sleep protocol is the subject of Sect. 2.7), the synchronous network may maximize the coordinated sleep opportunities because the synchronized on periods of neighboring sensors will better allow these sensors to cover for each other. Hence, sensor synchronization may have an advantage in energy efficiency.

In view of the energy saving potential of synchronous networks, we now generalize the synchronous network into a class of *regionally synchronized networks* in which the network is partitioned into a number of disjoint synchronization regions and the size of each region is specifiable. Specifically, we divide the network into a specifiable number of K equal size *clusters*. Each cluster defines a synchronization region in that all the sensors within the same cluster synchronize their periodic on/off schedules. However, each cluster will choose the beginning of its on period randomly and independently of the other clusters. Hence, although sensors within the same cluster are synchronous, different clusters are asynchronous with respect to each other. Figure 2.1b illustrates the regionally synchronized network. We assume that each sensor will belong to one and only one cluster, and it is clear that the larger the value of K, the smaller is the size of the cluster or synchronization region.

Regionally synchronous networks are interesting because they may reduce the synchronization overhead of the (global) synchronous network in that synchronization messages from a sensor will not need to be disseminated throughout the network but only within a local region. This is important because sensors are generally small devices with constrained computation and communication capabilities. Moreover, given that the (global) asynchronous network is expected to be more effective in event capture but less effective in coordinated sleep than the (global) synchronous network, the regionally synchronous network paradigm provides the intermediate design points so that a broad spectrum of tradeoffs between the event capture and coordinated sleep performance can be enabled.

We proceed to analyze the performance of the regionally synchronous network. We assume that the network is divided into K disjoint equal size clusters. For an arbitrary PoI x in the network, the circular region centered at x of radius r is then divided into several sub-regions depending on how many clusters intersect the circular region. Assume that cluster i has an intersection of size s_i with the circular region; clearly, s_i is a function of x and ranges within $[0, \pi r^2]$, and $\sum_{i=1}^{K} s_i(x) = \pi r^2$. If cluster i does not intersect the circular region, $s_i = 0$. Conversely, if the circular region is fully contained in cluster i, $s_i = \pi r^2$. For an event occurring at x, we can analyze the probabilities $P_{in}(x)$ and $P_c(x)$ as follows.

Lemma 2.1 ($X \in$ **Exponential**(α))**.** *Assume the network has K clusters, for a specific PoI x, the probabilities $P_{in}^K(x)$ and $P_c^K(x)$ of capturing an event occurring at x instantaneously and before the event disappears, respectively, are given by*

$$P_{in}^K(x) = \frac{1 - \prod_{i=1}^{K} (1 - \frac{q}{p}(1 - e^{-\gamma s_i(x)}))}{1 - e^{-\gamma \pi r^2}}, \tag{2.8}$$

$$P_c^K(x) = \frac{1 - \prod_{i=1}^{K} (1 - (1-\rho)(1 - e^{-\gamma s_i(x)}))}{1 - e^{-\gamma \pi r^2}}. \tag{2.9}$$

Proof. Recall $\frac{q}{p}$ and ρ in Theorem 2.4, which are denoted as the probabilities of capturing an event instantaneously and before it disappears, respectively. In each cluster, all the sensors behave like one big periodic sensor. For cluster i and PoI x, i has an area of size $s_i(x)$ intersecting with the circle centered at x of radius r. The probabilities that it can capture the event instantaneously or before it disappears are equal to $\frac{q}{p}(1-e^{-\gamma s_i(x)})$ and $(1-\rho)(1-e^{-\gamma s_i(x)})$, respectively. The event is captured if at least one of the clusters captures it, and the result follows.

As K changes, $P_{in}^K(x)$ and $P_c^K(x)$ will change. Even for the same K, the clustering can be performed in many different ways leading to different values of $s_i(x)$. For example, for the three clusters shown in Fig. 2.1b, $s_i(x_1)$ and $s_i(x_2)$ are different. $P_{in}^3(x_1)$ and $P_c^3(x_1)$ are the same as the values for the (global) synchronous network because x_1 is fully contained in one of the clusters. $P_{in}^3(x_2)$

and $P_c^3(x_2)$ are larger than the corresponding values for x_1, since x_2 lies near the boundary of the clusters so that it can be covered by more than one clusters. Hence, the values of $P_{in}^K(x)$ and $P_c^K(x)$ are dependent on the location of x. In general, the values of P_{in} and P_c for the whole regionally synchronous network will be higher than the corresponding values for the (global) synchronous network (i.e., $K = 1$) but smaller than those of the asynchronous network. When $K \to \infty$ and $\max A_i \to 0$, where A_i is the area of the cluster i, the regionally synchronous network should become like the asynchronous network. The following theorem formalizes this intuition.

Theorem 2.5 ($X \in$ **Exponential**(α))**.** *For general K, the network performance of the regionally synchronous network is less good than the asynchronous network but better than the synchronized network, i.e., $\forall K$, we have $q/p \le P_{in}^K(x) \le P_{in}^{async}$ and $1 - \rho \le P_c^K(x) \le P_c^{async}$, where P_{in}^{async} and P_c^{async} are the P_{in} and P_c of the asynchronous network. Furthermore, when the number of clusters $K \to \infty$ and $\max A_i \to 0$, the regionally synchronous network performs the same as the asynchronous network in terms of P_{in} and P_c,[3] i.e.,*

$$\lim_{K\to\infty} P_{in}^K(x) = \frac{1 - e^{-\frac{q}{p}\gamma\pi r^2}}{1 - e^{-\gamma\pi r^2}} = P_{in}; \tag{2.10}$$

$$\lim_{K\to\infty} P_c^K(x) = \frac{1 - e^{-(1-\rho)\gamma\pi r^2}}{1 - e^{-\gamma\pi r^2}} = P_c. \tag{2.11}$$

Proof. To obtain $P_{in}^K(x) \le P_{in}$, $P_c^K(x) \le P_c$, note that $f(x) = \ln(1 - \frac{q}{p}(1 - e^{-x})) + \frac{q}{p}x$ is an increasing function. In fact,

$$f'(x) = \frac{\frac{q}{p}(1 - \frac{q}{p})(1 - e^{-x})}{1 - \frac{q}{p}(1 - e^{-x})} \ge 0.$$

Then $f(x) \ge f(0) = 0$. Hence, we get that

$$\ln(1 - \frac{q}{p}(1 - e^{-\gamma s_i(x)})) \ge -\frac{q}{p}\gamma s_i(x).$$

So $\forall K \ge 1$,

$$\ln h^K \ge \sum_{i=1}^{K} -\frac{q}{p}\gamma s_i(x) = -\frac{q}{p}\gamma\pi r^2,$$

and it follows that $P_{in}^K(x) \le P_{in}$. A similar derivation shows that $P_c^K(x) \le P_c$.

[3] The proof goes through even if the clusters have significant different sizes.

We now show that $q/p \leq P_{in}^K(x)$, $1-\rho \leq P_c^K(x)$. We state the result: For $a \geq 0, b \geq 0, a+b=1, 0 \leq x_i \leq 1$, and for any positive integer K, the following inequality holds:

$$\prod_{i=1}^{K}(a+bx_i) \leq a + b\prod_{i=1}^{K} x_i. \tag{2.12}$$

We prove the above result by mathematical induction. When $K=1$, the equality obviously holds. Assume that (2.12) holds for $K=n$. When $K=n+1$, we have

$$\begin{aligned}\prod_{i=1}^{n+1}(a+bx_i) &\leq (a+b\prod_{i=1}^{n} x_i)(a+bx_{n+1}) \\ &= a^2 + abx_{n+1} + ab\prod_{i=1}^{n} x_i + b^2\prod_{i=1}^{n+1} x_i \\ &\leq a^2 + ab(1+\prod_{i=1}^{n+1} x_i) + b^2\prod_{i=1}^{n+1} x_i \\ &= a + b\prod_{i=1}^{n+1} x_i.\end{aligned}$$

Then for $K=1,2,\cdots$, (2.12) holds. Using (2.12), we have that

$$\prod_{i=1}^{K}(\frac{s}{p}+\frac{q}{p}e^{-\gamma s_i(x)}) \leq \frac{s}{p}+\frac{q}{p}\prod_{i=1}^{K} e^{-\gamma s_i(x)} = \frac{s}{p}+\frac{q}{p}e^{-\gamma\pi r^2},$$

which shows that $q/p \leq P_{in}^K(x)$. A similar computation yields the result that $1-\rho \leq P_c^K(x)$.

To prove Eqs. (2.10) and (2.11), let $h^K = \prod_{i=1}^{K}(1-\frac{q}{p}(1-e^{-\gamma s_i(x)}))$. By taking the log of h^K, we obtain $\ln h^K = \sum_{i=1}^{K}\ln(1-\frac{q}{p}(1-e^{-\gamma s_i(x)}))$. We have

$$\begin{aligned}&\lim_{s_i(x)\to 0}\frac{\ln(1-\frac{q}{p}(1-e^{-\gamma s_i(x)}))}{-\frac{q}{p}\gamma s_i(x)} \\ &= \lim_{s_i(x)\to 0}\frac{e^{-\gamma s_i(x)}}{1-\frac{q}{p}(1-e^{-\gamma s_i(x)})} = 1\end{aligned}$$

where the first equation holds by l'Hôpital's rule. Then

$$\ln(1-\frac{q}{p}(1-e^{-\gamma s_i(x)})) = -\frac{q}{p}\gamma s_i(x) + o(s_i(x)).$$

When $K \to \infty$ and $\max A_i \to 0$, then $\max s_i \to 0$. Let C denote the circular region centered at x. We have

$$\lim_{K\to\infty} \ln h^K = \lim_{K\to\infty} \sum_{i=1}^{K} (-\frac{q}{p}\gamma s_i(x) + o(s_i(x))).$$

By the definition of integrals, we can get

$$\lim_{K\to\infty} \ln h^K = \iint_C -\frac{q}{p}\gamma ds = -\frac{q}{p}\gamma\pi r^2.$$

Let $\ln h = \lim_{K\to\infty} \ln h^K$, then $h = e^{-\frac{q}{p}\gamma\pi r^2}$. The result for (2.10) directly follows. A similar derivation yields the result for (2.11).

Remark. Other distributions of sensor placement, e.g., a uniform grid placement, could be used, and the results in Theorems 2.3–2.5 can be readily adapted based on the density of sensors in a unit area.

2.7 Coordinated Sleep Under Periodic Scheduling

In a dense network, there is significant spatial overlap in the sensing regions of sensors. Temporally, the on periods of the periodic sensors may also overlap. The latter problem is the most severe in the case of the synchronous network, but it is also a concern in the asynchronous network particularly when q/p is large. Such overlap leads to coverage redundancy and waste of energy. When a significant fraction of the sensors die, the coverage of the network degrades severely.

We propose a solution in which sensors exchange information about their locations, residual energies, etc., so that a sensor, say i, whose sensing region is completely covered by those of its active neighbors can go to sleep to conserve energy. Before i can go to sleep, it needs permission from its neighbors because they have to agree to remain active and cover i's responsibilities. Sensors renegotiate the permission to sleep from time to time. The decision to grant permission is partly based on the residual energies of the competing nodes, which allows sensors to rotate their roles and achieve an energy balance for maximum network lifetime.

We now present a *coordinated sleep protocol* (CSP) for sensors to achieve the goals above. The protocol can be applied to the synchronous, regionally synchronous, and asynchronous networks. CSP is similar to the RACP protocol in [3], but our main purpose is to study its use in a periodic scheduling context. In Sect. 2.8, we will evaluate how the two techniques interact in several design points. We will also present performance comparisons with RACP where periodic scheduling can achieve significant energy savings with competitive performance.

In CSP, an active sensor can assume one of three roles: *regular*, *supporting*, and *redundant*. Sensors periodically exchange hello messages indicating their ids, locations, period start times, roles, and residual energies.[4] The locations of the sensors can be obtained by GPS or a localization protocol [14, 15]. A supporting sensor has agreed to be active for a stated amount of time to support the sleep of a neighbor. While active in supporting role, the sensor maintains a timer that signals the end of this role. When the timer expires, the sensor changes role to regular.

For a regular sensor, say i, each subset of its active neighbors whose sensing regions completely cover that of i is called a *support* of i. Sensor i periodically checks if it has a non-empty support set. If so, i can choose a support set by (C1) the minimum residual energy of its members, denoted by E^{res}, or (C2) the overlap, denoted by L, between the intersection of the members' on periods and i's own on period. The first criterion aims to maximize the energy balance of the network, while the second criterion aims to maximize the productivity of the sleep being negotiated. There is generally a tradeoff between the two criteria, which can thus be integrated into a hybrid measure, denoted by ω, as a function of E^{res} and L, e.g., a weighted sum of the two measures. Ideally, sensor i should choose a minimal support set that scores highest in the chosen criterion (i.e., E^{res}, L, or ω). However, a large number of candidate support sets may exist. If sensor resources are limited, a more efficient algorithm that approximates the ideal selection can be used by i for the actual support set selection, as follows. (i) Via the hello messages, i learns from each neighbor j E_j^{res} (the remaining energy of j) and L_j (the overlap between i's and j's on periods). If needed, i can then compute ω_j as the function of E_j^{res} and L_j that integrates the criteria C1 and C2 above. (ii) i sorts the neighbors in decreasing order of their scores in the support set selection criterion. (iii) i includes into the support set the smallest number of the sorted list's top neighbors such that their combined sensing ranges subsume that of i.

Once i has chosen its support set, it broadcasts a request-to-sleep (RTS) packet to the neighbors in the support. Each neighbor who does not desire to go to sleep immediately replies to i with a clear-to-sleep (CTS) packet. Once i receives a CTS from all the support's members, it broadcasts a confirm (CNF) packet to the support, changes to redundant role, and sleeps for L time, after which it changes back to regular. While still waiting for at least one more CTS, i may receive an RTS from a neighbor, say j. In this case, i sets a (per-neighbor) random delay timer T_j for j. If, by the time T_j expires, i is still waiting for a CTS, i sends j a CTS. Once i receives a CNF from j, it assumes supporting role for j for the specified time. An illustration of the protocol is shown in Fig. 2.2.

Note that if L is less than $\frac{2c}{k_1-k_2}$, the sleep is too short to be productive for energy savings. In this case, i will simply decide not to send an RTS. This situation can be common if q is small (particularly for asynchronous networks), so that the temporal

[4] If the on periods of two sensors do not overlap, they do not hear each other's hello messages. This does not affect performance since there is no temporal overlap of coverage between the two sensors.

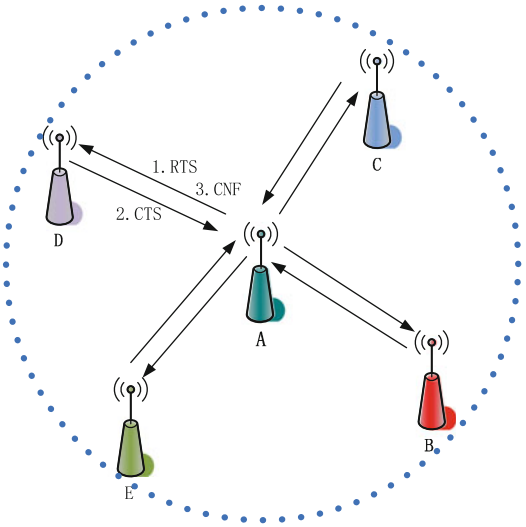

Fig. 2.2 Illustration of CSP protocol. The regular sensor A desires to go to sleep, and obtains permission by the following protocol message exchanges. (i) A sends a request-to-sleep (RTS) message to its supporting neighbors; (ii) The neighbors B–E send back a clear-to-sleep (CTS) to A if they can assume the supporting role; and (iii) A sends a confirm message (CNF) to B–E to indicate that it has obtained the required support

redundancy is not clear enough for coordinated sleep to help even if the sensor density is high. Another issue is i's setting of the random delay T_j. The design principle is that if i has a larger residual energy than that of j, then i should be more likely to send j the CTS before j sends i the CTS. Hence, it should be likely that i's delay timer for j is smaller than j's timer for i. We choose to pick T_j uniformly at random from $[0, 2^{E_j^{res}/E_i^{res}}]$, but alternative methods are possible.

Apart from support for periodic scheduling, CSP differs from RACP in its use of the delay timer. In RACP, a countdown delay is mandatory for every RTS. In CSP, i sends an RTS without delay, but a neighbor j who receives the RTS performs the random countdown if (and only if) j also desires to go to sleep. Hence, a countdown delay is avoided in CSP if there is no competition between neighbors in their sleep requests. Also, RACP uses a random sleep duration after a sensor has obtained permission to sleep, whereas CSP uses L as determined by the overlapping on periods of the sensors. Our experiments show that RACP's performance is more random in terms of variable network lifetimes achieved, while the lifetime of CSP is more predictable.

2.8 Numerical Results

We present Matlab results to illustrate and verify the analysis. In addition, we present Matlab simulations to evaluate the performance of the synchronous and asynchronous networks with and without coordinated sleep, including performance comparisons with RACP. Lastly, we will evaluate the regionally synchronous network under different sizes of the synchronization region, and compare its performance with the synchronous and asynchronous networks under coordinated sleep. We do not consider the energy costs of running either the CSP or RACP protocol. Hence, our comparison between CSP and RACP is fair. Further, the performance gains for both protocols are optimistic and can be regarded as upper bounds. Our results will show that in our synchronous, asynchronous, and regionally synchronous networks, the simplified best-case performance of CSP generally leads to fractional gains in system performance only, although these gains are indeed affected by different coordinated sleep opportunities in the different types of networks. Thus our simplification in the CSP evaluation will not affect the qualitative conclusion that the periodic scheduling, considered both locally for individual sensors and globally among the sensors, has a generally higher impact on system performance than CSP.

Unless otherwise stated, we use the following: (i) $X \in$ Exponential(α), $Y \in$ Exponential(β), $\alpha = 1$, and $\beta = 2$, where the distributions are numbers in time units of 0.1 h; (ii) for the energy model, $k_1 = 2.369$ J/h, $k_2 = 0.17$ J/h, and $c = 0.05$ J; (iii) each sensor of sensing range $r = 1$ m has an energy capacity of 9.26 mAh(equivalent to 100 J assuming a voltage of 3 V), and the sensors are deployed in a 20×20 m region according to a Poisson point process of intensity $\gamma = 4$, where γ is the average number of sensors per m^2; and (iv) for the distribution of PoIs, the deployment region is discretized into cells of dimensions 0.2×0.2 m and the center of each cell is a PoI. Each simulation in Sect. 2.8.1 is repeated until the standard deviation of the measurements is negligible compared with the average, and we report the average in the presentation. Representative traces are reported in Sect. 2.8.2.

2.8.1 Illustration of Analytical Results

2.8.1.1 Delay of Capture

Our approach allows a $P_c - P_{in}$ fraction of events to be captured with positive delay. Theorem 2.2 and Eqs. (2.2) and (2.3) quantify the expected delays D_{del} and D_{tot}. Figure 2.3a plots the analytical results for D_{del} against $p - q$ for different α. The measured averages in simulations are also shown as the indicated data points. Notice that the delay increases as the mean event staying time $\frac{1}{\alpha}$ increases. Also, when $p-q$

is large enough, most events that arrive early in an absent period do not stay long enough to be captured. Hence, the expected delay does not further increase with $p - q$ after some point.

Numerical results for the expected delay of all captured events, D_{tot}, are shown in Fig. 2.3b. Different from Fig. 2.3a, D_{tot} first increases and then decreases in Fig. 2.3b. To see how including D_{tot} in the network optimization may impact the results, we also plot the Q_E in Theorem 2.1 in Fig. 2.3b. Notice that for $p \in [0, p_1]$, both Q_E and D_{tot} increase. Hence, there is a tradeoff between Q_E and D_{tot} as the optimization goals. For $p \in (p_1, p_2]$, Q_E decreases while D_{tot} increases. Hence, this range of p is not optimal. For $p \in [p_2, \infty]$, both Q_E and D_{tot} decrease, again showing a tradeoff between the two optimization goals.

2.8.1.2 P_{in} of Asynchronous Network

Figure 2.3c verifies the correctness of Theorem 2.3 by plotting P_{in} against q/p for different γ. Notice the close agreement between the analytical and simulation results. More importantly, notice that at high sensor density (e.g., $\gamma = 4$), the achieved P_{in} is extremely close to 1 even if there is a significant fraction of off time in the periodic schedule (e.g., $\frac{q}{p} = 0.4$). *This shows that periodically resting the sensors does* not *necessarily cause noticeable delay in the event capture.*

2.8.1.3 Q_E of Synchronous Network

Figure 2.3d plots P_c (Theorem 2.1) against p for different values of q/p. Data points from simulations are also shown, which are in strong agreement with the analysis. For a fixed q/p, the function is monotonically decreasing in p, showing that we should turn on the sensor as briefly as possible at the start of each period. This is because once the sensor detects an event, the event is considered captured and the sensor does not gain by observing the event any longer. When energy is also considered, however, the Q_E plots given by (2.6) (Fig. 2.3e) initially increase with p and then decrease after reaching a global maximum. (The validation of Q_E follows directly from the validation of P_c.) Hence, the method outlined in Sect. 2.4 can be used to find the optimal p that maximizes Q_E.

2.8.1.4 Q_E of Asynchronous Network

We now illustrate Q_E, computed via the expression for P_c in Theorem 2.4, for the asynchronous network. We plot Q_E against p in Fig. 2.3f for different q/p. Notice that the plots are similar in trend to the Q_E plots of the synchronous network (Fig. 2.3e), although the peak performance is much less pronounced in this case. In particular, when $\frac{q}{p}$ is large (e.g., $\frac{q}{p} \geq \frac{1}{3}$), the Q_E value remains close to the optimal as p increases further.

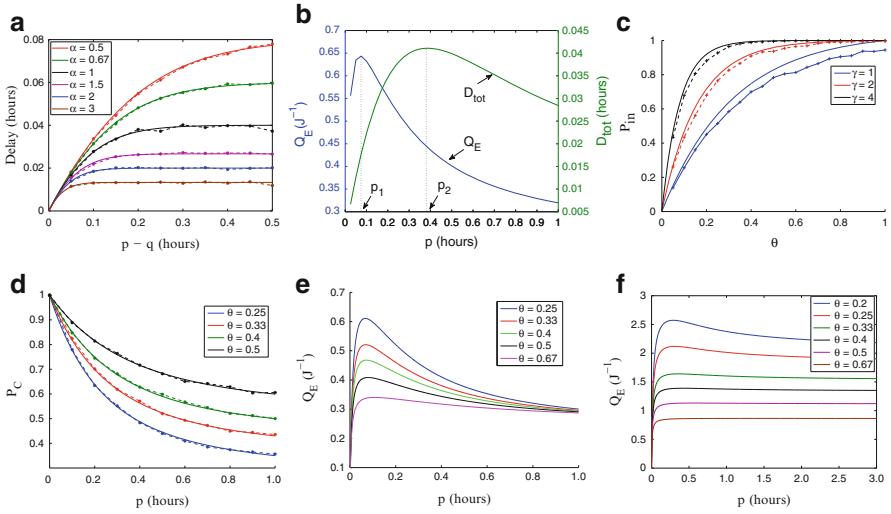

Fig. 2.3 Numerical results. (**a**) For events captured with positive delay, the average delay, D_{del}, against $p - q$ for varied α. (**b**) Q_E and D_{tot} against p for $\theta \triangleq \frac{q}{p} = 0.25$. (**c**) Plot of P_{in} of asynchronous network against $\theta \triangleq \frac{q}{p}$ for different γ. (**d**) Monotonically decreasing P_c of synchronous network against p for different $\theta \triangleq \frac{q}{p}$. (**e**) Plot of Q_E of synchronous network against p for different $\theta \triangleq \frac{q}{p}$. Optimal Q_E is achieved at an intermediate p. (**f**) Q_E of asynchronous network against p for different $\theta \triangleq \frac{q}{p}$. Peak of Q_E is less pronounced than synchronous network

2.8.2 Network Simulations

2.8.2.1 Asynchronous Network

We evaluate the asynchronous network with and without coordinated sleep by CSP. We assume that the user desires a stringent $P_{in}^{\min}$ close to 1, so that the required $\frac{q}{p} = 0.4$ by Theorem 2.3. Given $\frac{q}{p} = 0.4$, $p^* = 0.4\,\text{h}$ by optimizing Q_E numerically according to (2.6). Hence, $q^* = 0.16\,\text{h}$.

As a representative trace, Fig. 2.4a shows the achieved P_{in} as a function of the deployment time for RACP and A-CSP. Notice that A-CSP has quite similar performance as RACP before either network dies, but A-CSP has about 74 % longer network lifetime. When the RACP network dies, the coverage drops quickly to zero. Death of the A-CSP network is much more gradual. After A-CSP starts dying, it takes about 30 h more before the network dies completely. Plots of P_c for the two networks are very similar to the corresponding P_{in} plots.

Figure 2.4b shows the achieved P_{in} for RACP and the asynchronous network *without* coordinated sleep (i.e., A-nc case). In this case, the asynchronous network has quite similar performance as RACP before either network dies, but it lasts about 74 % longer than RACP. Hence, A-nc lasts about as long as A-CSP before A-CSP

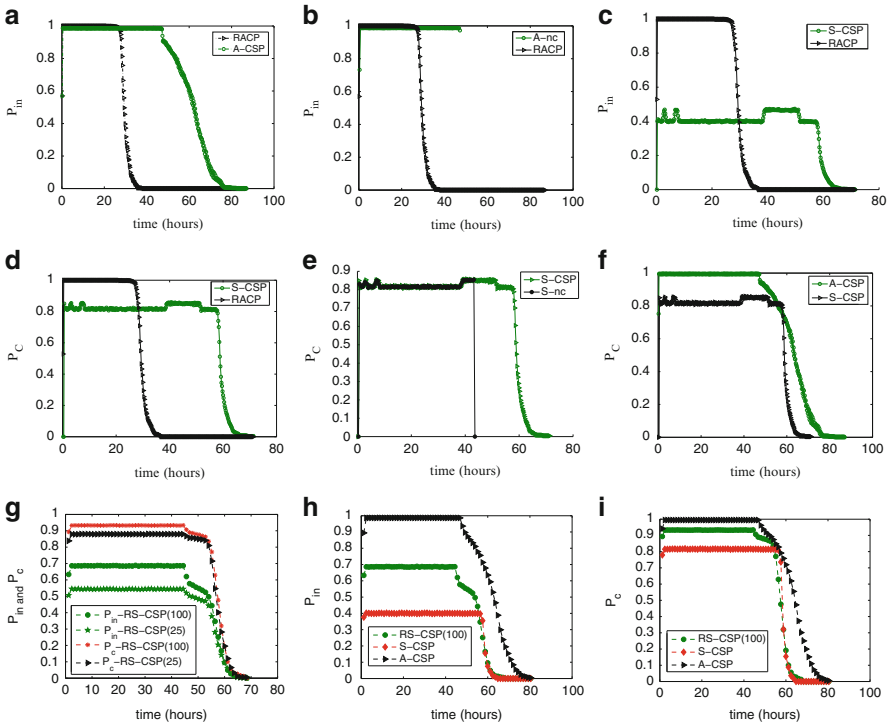

Fig. 2.4 Simulation results. (**a**) Achieved P_{in} against deployment time for RACP and A-CSP. A-CSP has longer lifetime and more gradual death than RACP. (**b**) P_{in} vs. deployment time for RACP and A-nc. A-nc runs longer than RACP. Its death is instantaneous since all the sensors work equally hard and run out of energy simultaneously. (**c**) P_{in} against deployment time for S-CSP and RACP. S-CSP achieves $P_{in}^{\min} = 0.4$ as specified by the user, and runs longer. (**d**) P_c vs. deployment time for S-CSP and RACP. S-CSP achieves $P_c = 0.8$ (cf. $P_{in} = 0.4$) and performs closer to RACP in terms of P_c. (**e**) P_c against deployment time for S-CSP and S-nc. CSP prolongs the network lifetime and achieves good energy balance between the sensors in the synchronous network. (**f**) P_c against deployment time for A-CSP and S-CSP. S-CSP operates longer before it starts to die and its death is less gradual than A-CSP. However, A-CSP performs better in terms of capture performance. (**g**) P_c and P_{in} as a function of the deployment time for the regionally synchronous network denoted as RS-CSP(K), where the number of clusters K is set to be 25 and 100 in two runs. (**h**) P_{in} against the deployment time for the synchronous, regionally synchronous, and asynchronous networks. (**i**) P_c against the deployment time for the synchronous, regionally synchronous, and asynchronous networks

starts to die. When A-nc dies, however, the coverage immediately drops to zero. This is because all the sensors work equally hard and run out of energy at the same time. Therefore, the usefulness of CSP in the asynchronous network lies mainly in its graceful degradation, where partial (but decreasing) coverage remains available over a significantly longer time duration. Because P_{in} is close to 1 for both networks, performance in terms of P_c is practically the same as P_{in}.

2.8.2.2 Synchronous Network

We now evaluate the synchronous network. We assume that the user is willing to relax the requirement for instantaneous event capture, and set $P_{in}^{\min} = 0.4$. Hence, $\frac{q}{p} = 0.4$. Optimizing p for Q_E in (2.6) yields $p^* = 0.075$ h. Hence, $q^* = 0.03$ h. Figure 2.4c compares RACP with S-CSP in terms of P_{in}. Notice that the network lifetime of S-CSP is about 2.1 times that of RACP. However, RACP has maximum instantaneous event-capture performance before it dies, whereas S-CSP achieves the relaxed P_{in} specified by the user (0.4). The performance comparison between RACP and S-CSP in terms of P_c is shown in Fig. 2.4d. Note that S-CSP has performance fluctuating between 0.81 and 0.85, which reflects the periodic on/off schedule of the network. Hence, the performance gap with RACP closes significantly in terms of P_c. This is because the synchronous network is designed to exploit the possibility of capturing an event before the event leaves a PoI. Hence, S-CSP allows the tradeoff of performance (17 % lower for P_c) for longer network lifetime (more than 2 times as long).

To evaluate the impact of coordinated sleep on the synchronous network, we compare the performance of S-CSP and S-nc in terms of P_c. The results are shown in Fig. 2.4e. Notice that (i) S-CSP lasts about 34 % longer than S-nc before S-CSP starts to die, and (ii) the death of S-nc is abrupt since the sensors work equally hard and they run out of energy at the same time, whereas the death of S-CSP is relatively more gradual.

2.8.2.3 Synchronous/Asynchronous Network Comparison

We now compare the synchronous and asynchronous networks under coordinated sleep. For A-CSP, we use the same evaluation case as in Sect. 2.8.2.1, i.e., $P_{in}^{\min}$ close to 1 and $\frac{q}{p} = 0.4$. For S-CSP, we either (i) keep $P_{in}^{\min}$ close to 1, in which case $\frac{q}{p}$ is also close to 1 (since $P_{in} = \frac{q}{p}$ in the synchronous network); or (ii) we keep $\frac{q}{p} = 0.4$, in which case we can only satisfy $P_{in}^{\min} = 0.4$. In case (i), the performance of S-CSP is similar to RACP's (although as Sect. 2.7 observes, the network lifetimes of RACP are more variable than S-CSP over different runs) since the sensors are active almost all the time. Hence, A-CSP performs better than S-CSP, as it performs better than RACP. In case (ii), it is clear that A-CSP performs better than S-CSP in terms of P_{in}. To see how A-CSP and S-CSP compare in terms of P_c and the network lifetime, refer to Fig. 2.4f. Note that S-CSP keeps the network running longer before it starts to die and the death is more abrupt, showing that S-CSP can achieve a better energy balance between the sensors. A-CSP starts dying sooner but takes a longer time to reach complete death. Considering the event capture performance throughout the deployment, A-CSP performs better than S-CSP overall.

2.8.2.4 Regionally Synchronous Network

We proceed to evaluate the performance of regionally synchronous networks. Since a main property of these networks is that they provide different levels of coordinated sleep opportunities, all the experiments in this section are conducted with CSP enabled.

We begin by evaluating the impact of the size of the synchronization region. We perform two experiments in which the network is divided into 25 and 100 clusters, respectively. For simplicity, we use a square network area, and divide it into regular 5×5 and 100×100 grids, respectively. We then use each cell in the grid to represent a cluster. The parameters p and q/p are set to be 0.075 and 0.4, respectively, which are the same as the corresponding S-CSP parameters used in the earlier experiments. The results comparing the numbers of clusters used are shown in Fig. 2.4g. It is clear that P_{in} and P_c are better in the 100-cluster case than in the 25-cluster case throughout the experiment. This shows that the better coordinated sleep opportunities in the 25-cluster network are not sufficient to offset the more effective event capture performance of the 100-cluster network (as discussed in Sect. 2.6). Hence, a larger number of clusters generally gives better overall performance when all the factors are considered. This general trend is true over a range of cluster sizes we have used in other experiments that are not reported in this chapter for simplicity.

We now compare the performance of the synchronous, regionally synchronous, and asynchronous networks. For the asynchronous network, we use the optimal parameters $p = 0.4$ and $q/p = 0.4$ as prescribed by the optimization procedure in Sect. 2.4. For the synchronous network, we use the optimal parameters $p = 0.075$ and $q/p = 0.4$. For the regionally synchronous network, we use the same setting as the synchronous network, i.e., $p = 0.075$ and $q/p = 0.4$. The P_c results for the three kinds of networks are shown in Fig. 2.4i, and the corresponding P_{in} results are shown in Fig. 2.4h. From the figures, we can see that the synchronous network starts dying later than the other two networks and achieves the best energy balance among them, showing that coordinated sleep is indeed more effective when the on periods of the sensors are more synchronized. However, similar to the previous experiments, this advantage is not enough to compensate for the synchronous network's reduced event capture performance as quantified by the previous analytical results. More generally, even when coordinated sleep is taken into account, the regionally synchronous network achieves an intermediate overall performance between the synchronous and asynchronous networks, and the asynchronous network remains the best option among the different kinds of networks.

2.8.3 Summary of Experiments

The experiments in Sect. 2.8.1 verify the analysis in Sects. 2.3–2.5 and illustrate the optimization of Q_E. The experiments in Sect. 2.8.2 compare S-nc, S-CSP,

A-nc, A-CSP, and RACP in a dense sensor network. We show that coordinated sleep in S-CSP can prolong the network lifetime compared with S-nc. S-CSP provides a performance/energy tradeoff versus RACP. It achieves the P_{in} specified by the user, and its achieved P_c can be significantly higher than P_{in} for events that stay. Moreover, the average delay of events captured can be quantified and used in the network optimization, although including the delay as an optimization objective may lead to tradeoffs with the Q_E objective, so that a user-specified balance between the two goals will be needed. We show that asynchronous periodic scheduling can achieve a higher global coverage intensity than S-CSP by spreading out the sensors' on periods. Importantly, the global operation can allow it to achieve maximum performance even in terms of *instantaneous* event capture (i.e., P_{in} close to 1) while being much more energy-efficient than RACP. A-CSP can further prolong the network lifetime over A-nc. It can achieve significantly longer network lifetimes than RACP with negligible loss of performance. More generally, A-CSP performs better than any regionally synchronized network, although this class of networks provides a broad spectrum of tradeoff between event capture performance and the effectiveness of coordinated sleep. Hence, A-CSP appears to be the best overall method in terms of maximum network lifetime and extremely good event capture performance.

2.9 Conclusions

We have investigated the use of periodic sensor scheduling for capturing stochastic events. We started off with the observation that when events can stay for some time, a (q, p)-periodic sensor can capture a fraction of events much higher than q/p. This is possible because events that arrive when a sensor is inactive may still be captured with a delay. We thus expect a tradeoff between energy efficiency (e.g., smaller q/p) and event capture (e.g., capture delay). We verify that such a meaningful tradeoff exists in the synchronous network relative to RACP. A similar tradeoff exists in the asynchronous network when the sensor density is moderately high, but the tradeoff becomes more attractive because the probability of non-instantaneous event capture decreases exponentially with $\frac{q}{p}$ in the asynchronous case (cf. linear decrease in the synchronous case). When the sensor density is high, the tradeoff becomes unnecessary in that the asynchronous network can achieve P_{in} close to one at high energy efficiency. When P_{in} is close to one, the performance is in fact not dependent on the event dynamics. We also considered a class of regionally synchronous networks where the size of the synchronization region is specifiable. This class of networks allows to control the synchronization overhead, and represents a broad spectrum of tradeoff between the effectiveness of event capture and that of coordinated sleep. We have evaluated the effectiveness of CSP for overall energy savings in the synchronous, regionally synchronous, and asynchronous networks.

References

1. S. He, J. Chen, D. Yau, H. Shao, and Y. Sun. Energy-efficient capture of stochastic events under periodic network coverage and coordinated sleep. *IEEE Transactions on Parallel and Distributed Systems*, 23(6):1090–1102, 2012.
2. I. Dietrich and F. Dressler. On the lifetime of wireless sensors networks. *IEEE Transactions Sensor Networks*, 5(1):1–38, Jan. 2009.
3. C. Hsin and M. Liu. Network coverage using low duty-cycled sensors: random and coordinated sleep algorithms. In *Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2004.
4. N. Bisnik, A. Abouzeid, and V. Isler. Stochastic event capture using mobile sensors subject to a quality metric. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2006.
5. D. Yau, N. Yip, C. Ma, N. Rao, and M. Shankar. Quality of monitoring of stochastic events by periodic and proportional-share scheduling of sensor coverage. In *Proceedings of The International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2008.
6. N. Jaggi, K. Kar, and A. Krishnamurthy. Rechargeable sensor activation under temporally correlated events. *Springer Wireless Networks*, 15(5):619–635, 2009.
7. S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu. Mobility limited flip-based sensor network deployment. *IEEE Transactions on Parallel and Distributed Systems*, 18(2):199–211, 2007.
8. X. Li, H. Frey, N. Santoro, and I. Stojmenovic. Strictly localized sensor self-deployment for optimal focused coverage. *IEEE Transactions on Mobile Computing*, 10(11):1520–1533, 2011.
9. T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization and its impact on large scale sensor networks. *ACM Transactions on Embedded Computing Systems*, 4(4):877–906, 2005.
10. Crossbow, http://www.xbow.com. *Crossbow MPR/MIB Users' Manual*.
11. K. Shenai and S. Mukhopadhyay. Cognitive sensor networks. In *Proceedings of International Conference on Microelectronics (MIEL)*, 2008.
12. M. Franceschetti, O. Dousse, D. Tse, and P. Thiran. Closing the gap in the capacity of wireless networks via percolation theory. *IEEE Transactions on Information Theory*, 53(3):1009–1018, 2007.
13. S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.
14. D. Evans and L. Hu. Localization for mobile sensor networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2004.
15. K. Yedavalli and B. Krishnamachari. Sequence-based localization in wireless sensor networks. *IEEE Transactions Mobile Computing*, 7(1):1–14, Jan. 2008.

http://www.springer.com/978-3-319-04647-1

Energy-Efficient Area Coverage for Intruder Detection in Sensor Networks

He, S.; Chen, J.; Li, J.; Sun, Y.

2014, VIII, 97 p. 32 illus., Softcover
ISBN: 978-3-319-04647-1