Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 365421, 12 pages
doi:10.1155/2012/365421

Research Article

Statistically Bounding Detection Latency in Low-Duty-Cycled Sensor Networks

Yanmin Zhu[1,2]

[1] *State Key Laboratory of Software Development Environment, Beijing, China*
[2] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*

Correspondence should be addressed to Yanmin Zhu, yzhu@cs.sjtu.edu.cn

Received 18 August 2011; Accepted 11 October 2011

Academic Editor: Mo Li

Detecting abnormal events represents an important family of applications for wireless sensor networks. To achieve high performance of event detection, a sensor network should stay active most of the time, which is energy inefficient for battery driven sensor networks. This paper studies the fundamental problem of bounding detection delays when the sensor network is low duty cycled. We propose a novel approach for statistically bounding detection latency for event detection in sensor networks. The key issue is the wakeup scheduling of sensor nodes and minimization of wakeup activity. We propose a lightweight distributed algorithm for coordinating the wakeup scheduling of the sensor nodes. A distinctive feature of this algorithm is that it ensures that the detection delay of any event occurring anywhere in the sensing field is statistically bounded. In addition, the algorithm exposes a convenient interface for users to define the requirement on detection latency, thereby tuning the intrinsic tradeoff between energy efficiency and event detection performance. Extensive simulations have been conducted and results demonstrate that this algorithm can successfully meet delay bound and significantly reduce energy consumption.

1. Introduction

Recent years have witnessed the rapid development of wireless sensor networks. The surge of interest in sensor networks is driven by the promising advantage of sensor network as a low-cost solution to a wide range of real-world challenges [1–6]. Event detection is an important class of applications for sensor networks. The key issues of designing a sensor network for distributed event detection are twofold. First, the system needs to provide quality event detection. That is, the detection of any event that occurs in the physical environment should be as timely as possible. Second, energy efficiency is critical since the battery-powered system is supposed to be continuously functional for months or even years.

Existing work [7–11] for event detection has extensively focused on providing full sensing coverage such that any potential event can be immediately detected after it arises. For energy efficiency, only a fraction of sensors are selected to be active, and the rest are put into sleep mode. The advantage of these algorithms is that no detection latency is incurred. The obvious drawback, however, is poor energy efficiency due to the fact that all active sensors need to be powered up constantly. Moreover, if a sensor fails, the sensing coverage supported by this sensor becomes a blind spot, and consequent critical events occurring at this spot will be lost, which is so-called the blind spot problem.

Most physical events are persistent, rather than ephemeral, which exist for a certain time, such as tens of seconds or even minutes, after its occurrence [12]. Examples for such events include fire, radiation, and pollution. This essential property allows sensors to capture the events while being in a low-duty cycle. A straightforward approach for energy-efficient event detection works as follows, like in [12, 13]. Each sensor sleeps most of the time and wakes up every τ_{cycle} time units, as shown in Figure 1, while in active mode, a sensor detects any potential event that occurs in its vicinity. Let τ_{on} denote the active time in every cycle of τ_{cycle}. The duty cycle of the sensor in Figure 1 is clearly

$$\delta = \frac{\tau_{\text{on}}}{\tau_{\text{cycle}}}. \tag{1}$$

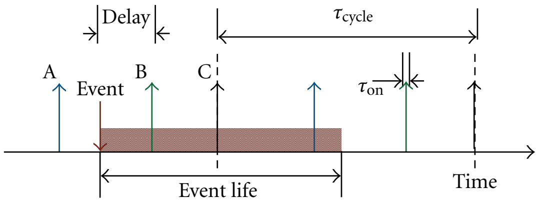

Figure 1: Example timing of three equally cycled sensors.

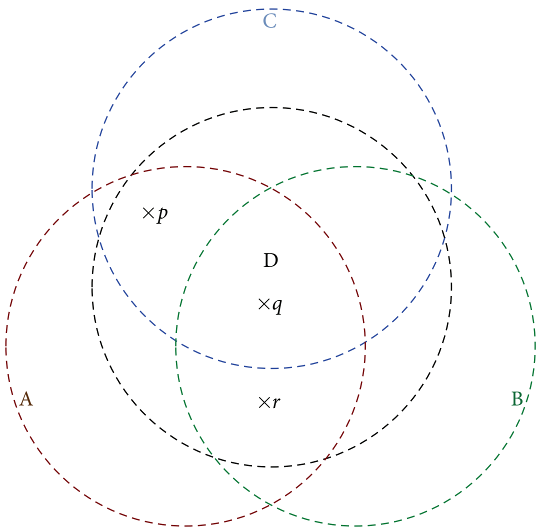

Figure 2: Illustration of the overdetection problem.

This suggests that the lifetime of the sensor node can be roughly extended by a factor of $1/\delta$. Nevertheless, it should be noted that the lifetime extension comes at the cost of additional detection latency. However, we argue that most practical applications tolerate detection latency. Note that it also takes time for the network to report the event back to the base stations through many hop-by-hop transmissions.

Recent study [6, 14] has shown that to guarantee full sensing coverage of the field, the density of sensors needs to be high. This implies that an event can possibly be detected by several ambient sensors. A critical challenge emerges in this situation. The dense deployment can cause a serious *overdetection problem* with Figure 2 when every sensor blindly wakes up in each cycle. As illustrated in Figure 2, an event that occurs at point q will be detected, within a single cycle, four times, and an event at point p will be detected three times since its emergence. Not only does the overdetection problem waste precious energy but also incurs additional energy overhead for event transmissions. Extra efforts are further needed to make a decision on whether or not event reports from different sensors are actually caused by the same physical event. The overdetection problem strongly motivates the idea that the duty cycle of each sensor should be reduced for better energy efficiency.

In this paper, we propose an innovative wakeup scheduling algorithm called PAD for energy-efficient event detection. The central idea is to reduce the duty cycle of every sensor via probabilistic wakeup, exploiting the dense deployment of sensor networks. The wakeup of a sensor is not deterministic, but instead probabilistic and adaptive to its sensing neighbors, therein significantly alleviating the overdetection problem. A distinctive feature of the algorithms is that it allows users to specify the requirement on detection latency and meanwhile ensures that the detection of any event is better than this requirement. The algorithm is truly scalable and power efficient, prolonging the system lifetime significantly. We have made the following contributions in this paper.

(i) By recognizing the essential latency-tolerant property of event detection applications, we investigate the energy-efficient approach for event detection, which addresses the serious overdetection problem.

(ii) We propose a soft bound model for detection delay specification and devise a simple yet effective metric to realize such a statistical soft bound.

(iii) We present insightful analysis on the nonadaptive scheme, in which the sensors wake blindly, and reveal the necessity to make adaptive control on wakeup frequency.

(iv) We propose a lightweight algorithm in which each sensor works adaptively and reduces its power dissipation substantially, hence remarkable prolonging system lifetime.

The rest of the paper is structured as follows. In Section 2, we present the system model, statistical bound modelfor detection latency, and the problem description. In Section 3, we analyze DoC and detection delay with a simplified wakeup scheduling and motivate the scheduling algorithm design. Section 4 proposes the wakeup scheduling algorithm and an extension. The performance evaluation is presented in Section 5. Related work is reviewed in Section 6. The paper is concluded in Section 7.

2. System Model and Problem Description

It is intuitive that there is an intrinsic tradeoff between system lifetime and detection latency. Thus, it is unrealistic to minimize detection latency and meanwhile to maximize system lifetime. For real-world surveillance applications, the system should deliver twofold performance. On the one hand, the detection delay of an event should not be arbitrarily large. Instead, it should be constrained to a certain range. On the other hand, the system should operate in a very power-efficient fashion. A longer system lifetime certainly requires the wakeup scheduling to be energy efficient. To extend the network lifetime, it is crucial to reduce the duty cycle of each individual sensor.

In this section, we present the system model, propose a soft bound model for event detection, and give the problem description. In the rest of this paper, we adopt the notations in Table 1.

2.1. Network Model. We consider a square field F with side length L, and the sensors are deployed in the field according

Table 1: Notations and descriptions.

Notation	Description
r	The detection range
δ	The duty cycle of the sensing device
φ	The duty cycle of the transceiver
$F_D(\cdot)$	The cumulative distribution function of random variable D
D_L	The longest delay specified by the user
D_0	The corresponding delay for χ_0
DoC	Detectability in one cycle
χ_p	The DoC of an event at point p
χ_0	The lowest DoC all over the field
γ_Q	The wakeup probability of sensor Q
$\gamma_Q(p)$	The necessary wakeup probability of sensor Q at point p
$S(p)$	The set of sensors covering point p
$U(Q)$	The set of grid points contained by sensor Q

to a 2-dimensional Poisson process with rate n/L^2. We focus on persistent events which exist for a certain time before they disappear. The event life is much longer than the wakeup cycle, and we can safely assume that an event is always detected. Each sensor has the knowledge of its location. A good number of power-efficient algorithms [15, 16] have been proposed for practical localization. Finally, each sensor has a detection range defining a detection disk centered at the sensor. An event is reliably detected by an active sensor if it resides in the detection range of this sensor.

The power consumption of a sensor node is mainly attributed to three units: processor, sensing device, and radio transceiver. Ideally, each unit has separate power control [17]. The duty cycle of the transceiver is subject to the control of communication protocols. We focus on the study of the duty cycling of the sensing devices. The transceiver may have a different duty cycle from the sensing devices. This indeed increases the flexibility for the algorithm to work with different communication protocols.

A sensor node can be attached with multiple sensing devices of different types. In the algorithm design, we assume, for simplification, that a sensor node is equipped with a single sensing device. However, such design can be easily extended to accommodate the situation of multiple sensing devices. In the rest of the paper, we call a sensor node simply a sensor if not confused with the sensing device.

2.2. Soft Bound for Detection Delay. The detection delay of an event is a random variable dependent on the arrival time of the event, the number of sensors covering the event, and the wakeup schedules of these sensors. It is ideal that the system provides a hard bound for detection delay, that is, any detection delay is less than a given value. However, this compels sensors to wake up at least once in every cycle, which will cause the serious aforementioned overdetection problem.

Providing *soft bound* for event detection is also very valuable for users. More specifically, the user specifies a longest delay (D_L) that is characterized by a cumulative distribution function (CDF). For example, it may be desirable for the user that 30% of events are detected within 1 s, 50% are within 2 s and 80% are within 3 s. Note that this longest delay specified by the user is actually a random variable. The system should then ensure that the detection delay (D) at any point is less than D_L.

Definition 1. Random variable D_1 is less than D_2, denoted by $D_1 \preceq D_2$, if the following condition holds:

$$F_{D_1}(d) \geq F_{D_2}(d), \quad d > 0, \tag{2}$$

where D_1 and D_2 share the same domain.

To specify the requirement on detection latency, the users can simply set the CDF of D_L. The objective of the system then becomes to ensure that the detection delay of any event is less than D_L. However, we have to address a new critical issue, that is, how to realize such a soft bound. To address this, we devise a simple yet effective metric, *detectability in one cycle* (DoC).

Definition 2. The DoC of point p (denoted by χ_p) is the probability that any event at p is detected, by at least one sensor, within a single wakeup cycle since its occurrence.

In fact, the DoC of point p characterizes the detection delay of any event at p, denoted by D_p. We derive the CDF of D_p which reveals the essential relationship between χ_p and D_p.

Theorem 3. *The CDF of D_p is given by*

$$F_{D_p}(d) = 1 - \left(1 - \chi_p\right)^c \left(1 - \frac{d - c\tau_{cycle}}{\tau_{cycle}}\chi_p\right), \quad \text{where } c = \left\lfloor \frac{d}{\tau_{cycle}} \right\rfloor. \tag{3}$$

Proof. By definition, the CDF of D_p is

$$F_{D_p}(d) = \Pr\left(D_p \leq d\right) = 1 - \Pr\left(D_p > d\right). \tag{4}$$

This implies that there is no sensor wakeup in the duration of d since the emergence of the event. There are c full cycles and an additional length of $d - c\tau_{\text{cycle}}$. The probability that no wakeup happens within one single cycle is $1 - \chi_p$, and that within duration of $d - c\tau_{\text{cycle}}$ is $1 - \chi_p(d/(\tau_{\text{cycle}} - c))$. □

2.3. Problem Description. With the introduction of DoC, it becomes possible to realize the soft bound on detection latency. First, we determine such a DoC χ_0 that the corresponding D_0 is less than D_L, that is,

$$D_0 \preceq D_L. \tag{5}$$

Second, we let the DoC of any point within the sensing field meet the following constraint:

$$\chi_p \geq \chi_0, \quad \forall p \in F. \tag{6}$$

It is apparent that a higher DoC at a point implies a shorter latency of event detection at this point. Thus, the derived D_p is less than D_0, that is,

$$D_p \preccurlyeq D_0, \quad \forall p \in F. \tag{7}$$

By combining (5) and (7), we can conclude that

$$D_p \preccurlyeq D_L, \quad \forall p \in F. \tag{8}$$

Thus, by guaranteeing that the DoC of any point is larger than χ_0, we are able to ensure that the detection delay of any event is less than the user's requirement D_L. Note that a more rigid requirement on real-time detection needs a higher χ_0. In the following, we derive the expected value of D_0, which follows a theorem.

Theorem 4. *The expected value of D_0 is*

$$E(D_0) = \frac{(2-\chi_0)\tau_{cycle}}{2\chi_0}. \tag{9}$$

Proof. The expected delay is $\tau_{\text{cycle}}/2$ if the event is detected within the first cycle. If it is detected in the jth cycle, $j > 1$, then additional $(j-1)\tau_{\text{cycle}}$ latency is introduced. Let M denote the number of full cycles that an event undergoes before it is detected. The probability mass function (*PDF*) of M is given by

$$\Pr(M = k) = (1-\chi_0)^{k-1}\chi_0, \quad k \geq 0. \tag{10}$$

We derive the expected delay by conditioning on M,

$$\begin{aligned} E(D_0) &= \sum_{i=0}^{\infty}\left(\frac{\tau_{\text{cycle}}}{2\times\Pr(M=i)}\right) \\ &= \frac{(2-\chi_0)\tau_{\text{cycle}}}{2\chi_0}. \end{aligned} \tag{11}$$

The expected delay is a function of χ_0 and is inversely proportional to χ_0. □

The goal of the network is to make sure that any event e that occurs in the sensing field is detected by the sensor network with a detection delay, D_e, that is statistically bounded by D_0:

$$D_e \preccurlyeq D_0. \tag{12}$$

At the same time, the sensor network should be as energy efficient as possible.

3. Analysis of DoC and Detection Delay

In this subsection, we are interested in the nonadaptive scheme (NAS) in which the wakeup probability of each sensor is identical, that is, not adaptive to its neighborhood. To guarantee that the DoC of any point is greater than χ_0, NAS simply sets the wakeup probability in each cycle of every sensor to χ_0. The problem is that when the density is high, the actual DoC of a point can be much higher than χ_0, resulting in unnecessary waste of energy.

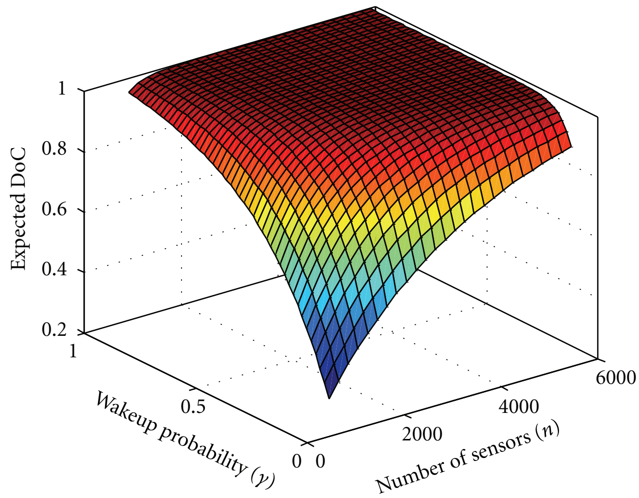

Figure 3: Expected χ_p as a function of n and γ, L = 300 m, and r = 10 m.

3.1. DoC Analysis. First, we analyze the DoC of any point in the field of NAS and present it in a theorem.

Theorem 5. *With NAS, the expected DoC of any point in the sensing field is*

$$E(\chi_p) = 1 - e^{-\lambda\gamma}. \tag{13}$$

Proof. Let point p be an arbitrary point in the field. Note that we do not consider the special points on the edge. The DoC of p is actually a random variable because it relies on the number of covering sensors and the wakeup probabilities of the sensors covering p. The number of sensors covering point p is a random number, denoted by X_p, which is Poison distributed with the PDF as

$$\Pr(X_p = k) = \frac{1}{k!}\lambda^k e^{-\lambda}, \quad \text{where } \lambda = \frac{n\pi r^2}{L^2}. \tag{14}$$

Let γ denote the identical wakeup probability. By conditioning on X_p, we have

$$E(\chi_p) = \sum_{k=1}^{n}\left(\left(1-(1-\gamma)^k\right)\times\Pr(X_p = k)\right) = 1 - e^{-\lambda\gamma}. \tag{15}$$

□

Figure 3 plots the expected value of χ_p as a function of n and γ. We can find that both increasing density and increasing the wakeup probability can increase the DoC of p. When the density is high, even a relatively low wakeup probability can produce a high DoC close to one. This strongly suggests that the wakeup probability can be reduced so that the DoC is close to χ_0, therefore conserving more energy.

3.2. Delay Analysis. Next, we consider the detection delay achieved by NAS.

Theorem 6. *With NAS, the expected detection delay of an event that happens at any point is given by*

$$E(D) = \left(\frac{1-(1+\lambda\gamma)e^{-\lambda\gamma}}{\lambda\gamma} + \frac{e^{-\lambda\gamma}}{1-e^{-\lambda\gamma}}\right)\tau_{cycle}. \quad (16)$$

Proof. Let Y_p denote the number of sensors being active in a cycle. We derive the PDF of Y_p by conditioning on X_p,

$$\begin{aligned}\Pr\left(Y_p = i\right) &= \sum_{j=0}^{n}\Pr\left(Y_p = i \mid X_p = j\right) \times \Pr\left(X_p = j\right) \\ &= \frac{1}{i!}(\lambda\gamma)^i e^{-\lambda\gamma}.\end{aligned} \quad (17)$$

Interestingly, Y_p turns out to be Poisson distributed with rate $\lambda\gamma$.

According to the analysis in [6], the expected delay of an event is given by $D_c = \tau_{\text{cycle}}/Y_p$, if it is detected in the first cycle. The expectation of D_c is given by

$$E(D_c) = \sum_{k=1}^{n}\left(\frac{\tau_{\text{cycle}}}{k+1}\cdot\Pr\left(Y_p = k\right)\right). \quad (18)$$

Let N denote the number of full cycles that elapsed before an event is detected. The PDF of N is

$$\Pr(N = i) = (1-\theta)^{i-1}\theta, \quad (19)$$

where θ is the probability that an event is detected within one cycle. It is apparent that

$$\theta = 1 - \Pr(Y = 0) = 1 - e^{-\lambda\gamma}. \quad (20)$$

If an event is detected in the ith cycle, an additional latency of $(i-1)\tau_{\text{cycle}}$ is introduced. Thus, we can compute the expectation by conditioning on N,

$$\begin{aligned}E(D) &= \sum_{i=1}^{\infty}\left(\left(E(D_c) + (i-1)\tau_{\text{cycle}}\right) \times \Pr(N = i)\right) \\ &= \left(\frac{1-(1+\lambda\gamma)e^{-\lambda\gamma}}{\lambda\gamma} + \frac{e^{-\lambda\gamma}}{1-e^{-\lambda\gamma}}\right)\tau_{\text{cycle}}.\end{aligned} \quad (21)$$

□

Figure 4 plots the expected delay as a function of wakeup probability. We have two observations. First, increasing wakeup probability produces a decreasing delay expectation, as is obvious in the sense that an event is more likely to be detected in earlier cycles. Second, expected delays of NAS are all much smaller than D_0, especially when the density of sensors is high. This highly suggests that the system should introduce adaptive control over the wakeup probability such that each sensor operates more energy efficiently.

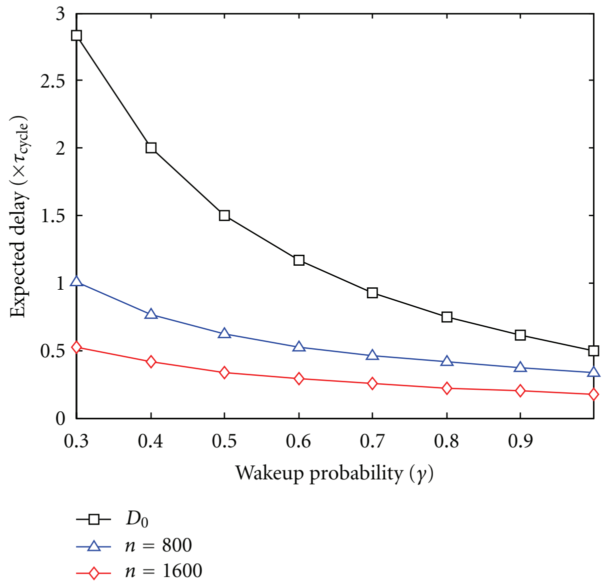

FIGURE 4: Expected delay as a function of γ, L = 300 m, and r = 10 m.

4. Probabilistic Wakeup

The design goals of the sensor system are (1) to extend system lifetime by reducing the duty cycle of every sensor; (2) to ensure that the detection latency of any event is statistically bounded by the requirement posed by the users. As discussed previously, this is to be achieved by ensuring that the DoC of any point is larger than χ_0. PAD adopts a probabilistic approach and solves the overdetection problem by adaptively tuning wakeup frequency, exploiting the natural dense deployment.

Following the probabilistic approach, a sensor Q wakes up in each cycle with probability γ_Q and remains in sleep mode with probability $1-\gamma_Q$. The key issue is clearly the determination of the wakeup probability. The wakeup probability should be as small as possible for the power efficiency purpose. At the same time, however, it ought to be sufficiently large to guarantee the DoC of location points within its ambient neighborhood.

An event can arise anywhere in the sensing field, and it is impossible to predict the arising location of the event. Thus, we need to consider any location point in the field. As there are infinite number of points, we divide the whole sensing field into virtual grids and consider the finite set of grid points. It is obvious that with division of smaller grids, we can have more fine-grained guarantee on detection latency. At the same time, however, a smaller grid size will introduce more grid points, and thus a higher space and time complexity will be incurred on each sensor node.

The state transition diagram of the algorithm is depicted in Figure 5.

4.1. Design of Wakeup Scheduling Algorithm. The algorithm executes in two phases. In the initialization phase, each sensor discovers its neighbors. Based on the neighborhood information, a sensor determines a conservative wakeup probability. This probability is sufficiently large to guarantee the DoC of any point, and hence it results in power inefficiency. In the next phase, in response to energy inefficiency,

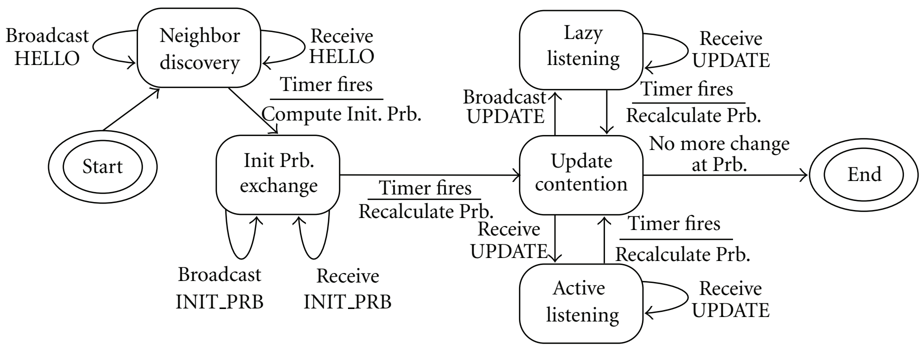

Figure 5: State transition diagram of the PAD algorithm.

an iterative optimization procedure is carried out among the sensors, to further reduce wakeup frequency.

4.1.1. Conservative Initialization. At the beginning, each sensor tries to find its neighbors within $2r$ distance from itself by exchanging HELLO messages with each other. For a given sensor, a neighbor is a *sensing neighbor* (distinguished from a communication neighbor) if its distance to the neighbor is less than $2r$. Every sensor maintains a table for its sensing neighbors. Upon receiving a HELLO, the sensor records the sender in the table if the sender is a sensing neighbor; otherwise, the HELLO is silently dropped.

After the neighbor discovery, the sensors start to compute its initial wakeup probability by executing the *Conservative Wakeup Determination* (CWD) algorithm. The initial wakeup probability guarantees that the DoC of any point is greater than χ_0. Let $S(p)$ denote the set of sensors covering point p. The DoC of point p is

$$\chi_p = 1 - \prod_{B \in S(p)} (1 - \gamma_B). \tag{22}$$

To meet constraint (6), each sensor firstly computes the necessary probability for every grid point (*point level*) within its detection range and then computes the wakeup probability of the sensor (*node level*). In CWD, the sensors covering the same point are supposed to play an equally important role in detecting events at this point. Take sensor Q, for example, and its necessary probability for a point p within its detection range is,

$$\gamma_Q(p) = 1 - \sqrt[k]{1 - \chi_0}, \quad \text{where } k = |S(p)|. \tag{23}$$

To compute its wakeup probability at the node level, Q takes the maximum out of the wakeup probabilities of all grid points within its detection range. Let $U(Q)$ denote the set of all the grid points within the detection range of Q. Then, the node-level wakeup probability of Q is

$$\gamma_Q = \max\{\gamma_Q(p), \forall p \in U(Q)\}. \tag{24}$$

CWD is conservative since each sensor takes the maximum as its wakeup probability. The wakeup probability is sufficiently large for every grid point in its detection range to have a larger DoC than required. The consequence is that the DoC of a point may actually be much larger than the required χ_0. Such conservativeness incurs unnecessary energy consumption.

4.1.2. Optimization. It is imperative to further improve the energy efficiency after the initial selection. Therefore, we propose a *cooperative refinement procedure* (CRP) to refine the wakeup frequency of each sensor node. Following this procedure, each sensor derives a new wakeup probability based on the wakeup probabilities of its sensing neighbors. If the newly computed wakeup probability is smaller, it tries to adjust its wakeup probability, attempting to reduce its wakeup frequency. CRP executes round by round. In each round, a sensor can update its probability at most once. It is guaranteed that CRP terminates in a finite number of rounds.

After determining the initial wakeup probability, sensors exchange their wakeup probabilities by local broadcast. Each sensor recalculates a feasible wakeup probability based on the wakeup probabilities of its sensing neighbors. Similar to CWD, a sensor firstly computes a new wakeup probability for each grid point. The new feasible wakeup probability for point p is given by

$$\gamma_Q{}^{(k+1)}(p) = \begin{cases} 0, & \text{if } \prod_{B \in S(p) - \{Q\}} (1 - \gamma_B) > 1 - \chi_0 \\ 1 - \dfrac{1 - \chi_0}{\prod_{B \in S(p) - \{Q\}} (1 - \gamma_B{}^{(k)})}, & \text{otherwise}, \end{cases} \tag{25}$$

where (k) denotes the number of generation to which the corresponding wakeup probability belongs.

To compute the new wakeup probability at the node level, Q also takes the maximal probability among those of all the grid points within its detection range,

$$\gamma_Q{}^{(k+1)} = \max\left\{\gamma_Q{}^{(k+1)}(p),\ p \in U(Q)\right\}. \tag{26}$$

If the new probability is smaller than the original one, the sensor will update the probability to the new one for the energy efficiency purpose. Thus, any sensor that obtains a smaller new probability makes an update attempt, trying to reduce its probability.

Due to the computation dependence, it is critical to avoid parallel updates. CRP requires that before a sensor can actually update its wakeup probability, it must broadcast the new probability to its sensing neighbors and suppress them from updating simultaneously. An UPDATE message is used to enclose the ID and the new probability of a sensor. Before an UPDATE is broadcast, the sensor undergoes a random backoff to minimize UPDATE transmission collisions. If a sensor receives an UPDATE from its sensing neighbor before its own UPDATE is broadcast, it suppresses its planned UPDATE broadcast and cancels its own update attempt (if any). However, if a sensor successfully broadcasts its UPDATE, it commits the update attempt, updating its wakeup probability.

After successfully broadcasting an UPDATE, in theory, a sensor would not receive any UPDATE from its sensing neighbors. However, unreliable wireless transmissions make it still possible that the sensor receives some. In CRP, a sensor that has successfully committed its update stays in the *lazy* state, where it ignores any UPDATE following its broadcast. For those sensors that cancelled its update attempt, they actively listen, receive all the UPDATE from its sensing neighbors, and update the corresponding wakeup probability in the locally maintained table.

4.2. Extension for Differentiation. It is sometimes necessary for some areas to be more carefully monitored, necessitating detection differentiation. The differentiation can be in either *detection latency* or *detection degree*. When the differentiation is in detection degree, we modify the definition of `DoC` to accommodate the need of higher degrees. Recall that the `DoC` of a point is the probability that an event is detected, by at least one sensor, within one cycle. If a point requires a degree of two, we define the *quadratic* `DoC` of a point as the probability that an event at this point is detected, by at least *two sensors*, within one cycle. A higher degree provides more robust event detection against sensor failure.

(i) To have a shorter latency for a specific point q, we can easily set a larger `DoC` for point q, for example, $\chi_0(q)$. Then all sensors covering q should replace χ_0 with $\chi_0(q)$ in all previous computations.

(ii) To have a higher degree, more sophisticate modifications over PAD are necessary, which are elaborated as follows.

Suppose the degree for grid point q is two, rather than one. In the process of computing the initial probability, likewise, the sensors covering q should play an equal role, therefore having an identical probability, denoted by γ. For differentiation, we term the `DoC` at q as quadratic `DoC`, denoted by $\hat{\chi}_q$

$$\hat{\chi}_q = 1 - (1-\gamma)^k - k(1-\gamma)^{k-1}\gamma, \quad \text{where } k = |S(q)|. \tag{27}$$

Each sensor covering q needs to compute the necessary initial probability for q. It is difficult to compute the exact root of (27) since it needs to solve a high-dimensional equation. Nevertheless, the quadratic `DoC` monotonously increases with increasing γ. Thus, we can develop a numerical procedure to find a desirable γ that is close to the real root.

In the refinement procedure, a sensor adjusts its wakeup probability based on the wakeup probabilities of its sensing neighbors. When the detection degree is two, the formula (25) should be reformulated as follows:

$$\gamma_Q{}^{(k+1)}(q) = \begin{cases} 1 - \dfrac{1-\chi_0-v}{v\omega}, & \text{if } 1-\chi_0-v < v\omega, \\ 0, & \text{otherwise,} \end{cases}$$
$$\text{where } v = \prod_{B\in S(q)-\{Q\}} \left(1-\gamma_B{}^{(k)}\right), \tag{28}$$
$$\omega = \sum_{B\in S(q)-\{Q\}} \frac{\gamma_B{}^{(k)}}{1-\gamma_B{}^{(k)}}.$$

5. Evaluation

In this section, we first introduce the experiment methodology and simulation setting. Next, we discuss the evaluation results.

5.1. Methodology and Simulation Setup. To evaluate the performance of PAD, we conduct extensive simulation experiments with a simulator developed for simulating a low-duty-cycled sensor network. We adopt the data of the eXtreme Scale Mote [12]. The setting of the key simulation parameters is shown in Table 2, if not specified elsewhere. A sensor is usually powered by two AA batteries, which can typically provide about 2×10^4 J energy. In our simulations, however, the initial energy for every sensor is set to 50 J to reduce lengthy simulations. The results presented in this section are averaged over 20 independent experiments with different sensor deployments.

We compare the performance of PAD with NAS and the upper theoretical bound in terms of system lifetime extension. We define the *hard lifetime* as the time from the starting time to the time instant when the `DoC` of any point within the field drops below χ_0. We define *α-lifetime* as the time until less than α% area of the field can meet the `DoC` requirement. The hard lifetime is highly subject to the influence of irregular deployment. To address this issue and study the energy conservation ability of PAD, soft lifetime is more suitable, which is less sensitive to sensor deployment.

Table 2: Simulation setting.

Parameter	Value	Parameter	Value
R	30 m	L	300 m
r	10 m	ξ	50 J
ρ_S	19.4 mW	τ_{cycle}	10 s
ρ_P	20 mW	τ_{on}	0.5 s
ρ_R	24 mW	n	2700
φ	0.01	χ_0	0.8

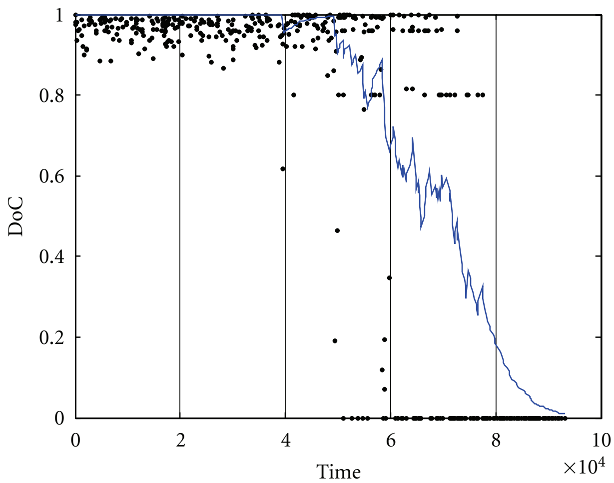

Figure 6: DoC over time.

It is difficult to derive the tight bound of the hard system lifetime. We give an optimistic upper bound of the lifetime. Let ρ_P, ρ_S, and ρ_R denote the power rates of the processor, the sensing device, and the transceiver, respectively. A point in the field is covered by λ sensors. Ideally, these sensors share the same wakeup probability, which is $1-(1-\chi_0)1/\lambda$. Thus, the actual power consumption rate of the sensor unit is $\rho_s\tau_{\text{on}}(1-\sqrt[\lambda]{1-\chi_0})/\tau_{\text{cycle}}$. The upper bound of the hard lifetime can be computed accordingly,

$$\mathcal{T}_{ub} = \frac{\xi}{\rho_P\varphi + \rho_S\delta' + \rho_R(\varphi + \delta')}, \quad \text{where } \delta' = \frac{\tau_{\text{on}}}{\tau_{\text{cycle}}}\left(1 - \sqrt[\lambda]{1-\chi_0}\right). \tag{29}$$

5.2. Typical Run. The first set of experiments investigates the performance of PAD in a typical run. Figure 6 reports DoCs of events over time. Each point in the figure represents the DoC of a random point. The exponentially weighted moving average of DoC is also shown using a solid curve. We can see that PAD successfully guarantees that the DoC of any point is larger than 0.8 before 4×10^4 s. After that time, some region becomes uncovered because of sensor depletion. Beyond the time 4×10^4 s, the DoC in some area falls below 0.8. This results in the factor that different areas in the sensing field have different numbers of covering sensors.

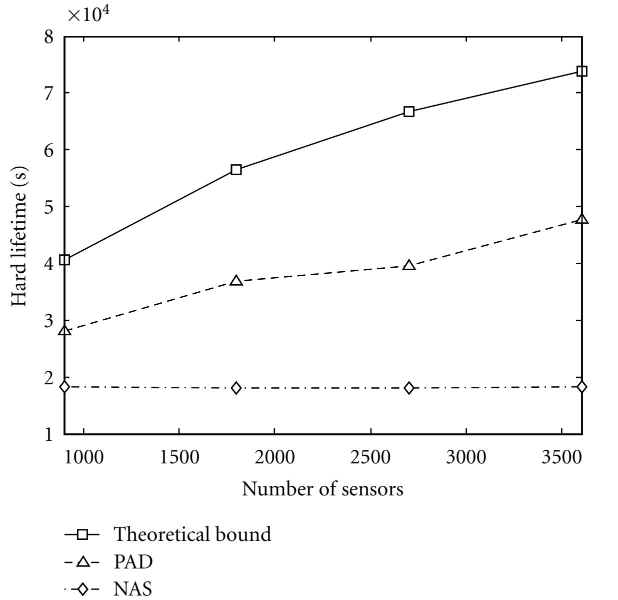

Figure 7: Hard lifetime versus number of sensors.

5.3. Lifetime Extension. The second set of experiments investigates lifetime extension of the algorithm, in comparison to NAS and the theoretical bound. We vary the number of sensors to study lifetime extension under different density configurations. As we can see in Figure 7, the hard lifetime of the upper bound increases proportionally with the increasing number of sensors. We can see that PAD extends the hard lifetime remarkably, compared with NAS. With the increasing number of sensors, the lifetime extension becomes more significant. This demonstrates that PAD can effectively exploit high sensor density. NAS fails to extend system lifetime even if the sensor density becomes higher. It is because with NAS every sensor wakes up blindly with probability χ_0 in each cycle. In Figure 8, we show the soft lifetime of PAD and NAS. As we can see, the soft lifetime significantly increases with the increasing number of sensors. For NAS, however, the soft lifetimes for different α remain the same. In addition, the increasing density does not lead to a longer system lifetime.

5.4. Detection Delay. In the third set of experiments, we explore detection delay for different schemes. Figure 9 reports the average detection delay under different sensor densities. We can see that the average delay achieved by PAD is below the bound but larger than that of NAS. PAD effectively reduces wakeup probability of the sensors and consequently increases detection delay. In contrast, in NAS, each sensor has the same wakeup probability. A point is covered by more sensors when the density increases. This suggests that detection delay decreases. It is worth noting that with increasing density, PAD's detection delay decreases, but the decrease is much slower than NAS.

5.5. Algorithm Convergence. In this set of experiments, we explore the convergence by investigating the number of

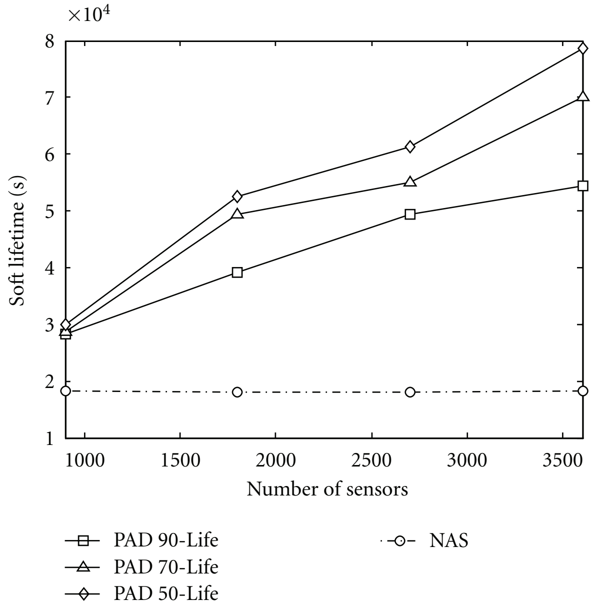

Figure 8: Soft lifetime versus number of sensors.

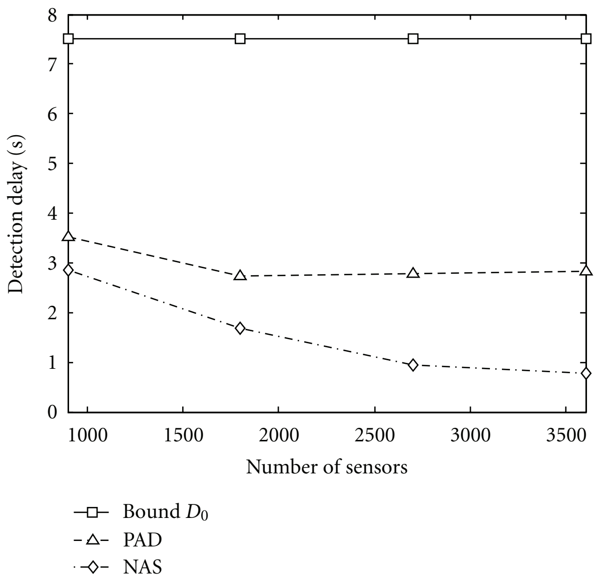

Figure 9: Detection delay versus number of sensors.

requesting sensors (the sensors that try to update wakeup frequency) in each round over time. The PAD algorithm converges when there are no more requesting sensors. Figure 10 illustrates the number of requesting sensors over time. We can see that the number of requesting sensors in a round decreases as time elapses. There is no more requesting sensor, and the refinement procedure terminates when the time reaches 28 s. This shows that the refinement procedure of PAD is able to quickly converge and starts functioning soon after the deployment of the sensor network.

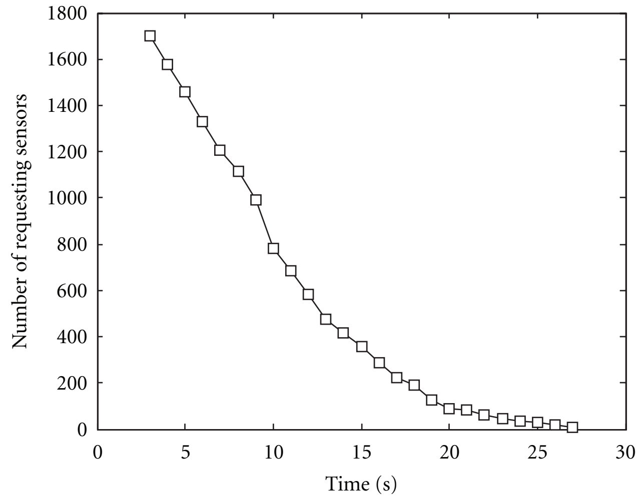

Figure 10: Number of requesting sensors over time.

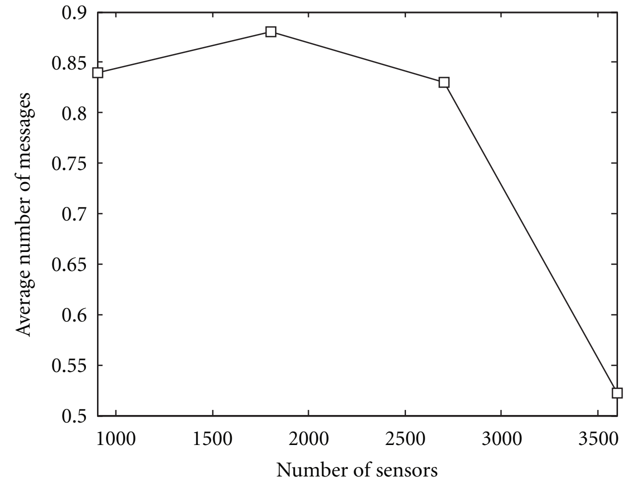

Figure 11: Number of messages per sensor versus number of sensors.

5.6. Overhead. We also study communication overhead introduced by the PAD algorithm. Figure 11 shows the number of algorithm messages per sensor under different configurations with varying number of deployed sensor nodes. We can see that the average number is below 0.9 messages per sensor. More interestingly, the number of algorithm messages per sensor is decreasing when the number of sensors is increasing. This shows that increasing density does not incur higher overhead per sensor. Such communication complexity is affordable for sensors, and hence PAD is scalable with respect to sensor density.

6. Related Work

It has been an effective approach for conserving power of sensor nodes with duty cycling [6, 12, 13, 18–22]. Many methods have been proposed in the literature. Armstrong made a good survey on energy-conserving methodologies

[23]. A lot of power-scheduling algorithms [18] are aimed to extend network lifetime via scheduling sleep/active states of sensors. Asynchronous wakeup scheduling [10] is superior since it is not dependent on time synchronization. However, it introduces additional packet delivery delay. On the MAC layer, the low-duty cycling of the radio transceiver can be employed to reduce energy consumption of the senor node [24–26].

For object tracking applications, when the sensor nodes are duty cycled, energy-quality tradeoffs are intrinsic [2, 4]. Probabilistic coverage in sensor networks has been studied in the context of object tracking [27]. In [28], the authors deployed a test of 70 sensors to track positions of mobile vehicles. In the testbed, only 5% of sensor nodes were kept active, and the rest of the senor nodes operated at a very low-duty cycle (4%). Under this configuration, the network was still capable for tracking vehicles.

Maintenance of full sensing coverage has been of significant importance for many sensor network applications. Several algorithms have been proposed to select a small subset of the sensor nodes to stay active to maintain full coverage and turn off the rest sensor nodes for energy conservation. PEAS [10] makes use of a heuristic that when a sensor is active its neighborhood sensors can go to sleep. Each sensor periodically sends probe signals based on which neighbor sensors can decide to sleep or not. Yan et al. [9] identifies a redundant sensor node whose sensing coverage is jointly covered by its active neighbors. In [29], random and coordinated algorithms have been studied for maintaining the network coverage of a sensor network in which sensor nodes are low duty cycled.

Gupta et al. [30] proposed a randomized algorithm to determine an active schedule of the sensors. At any time, the set of currently active sensors guarantee to provide full sensing coverage. Some other effort [8] considers both sensing coverage and network connectivity. These algorithms provide full sensing coverage and meanwhile maintain network connectivity.

Shakkottai et al. [31] studied the coverage of a sensor network where the sensor nodes are not reliable. In [30], the sensors were divided into several groups and algorithms were developed to maximize the sum of sensing coverage. Event detection using low-duty-cycled sensors has also been discussed [12, 13]. In [13], the authors propose a two-stage optimization algorithm to minimize detection latency. A node platform eXtreme Scale Mote [12] was designed for long-lived operations detecting ephemeral events.

Some existing works focus on detecting complex events [3, 32, 33]. In these works, an event may span a certain region. The occurrence of an event must meet a certain requirement on its boundary. Thus, contour mapping becomes an important operation for event detection in sensor networks. A few effective distributed algorithms [3, 32, 33] have been proposed for event detection based on contour mapping.

In this work we focus on providing soft detection bound for event detection in sensor network. The unique contribution of this work is twofolds. First, we propose a novel soft bound model for delay tolerance specification by users or applications. Second, we propose a distributed wakeup scheduling algorithm that ensures the detection delay of any event in the sensing field to be statistically bounded. Thus, this work is complementary to existing works for event detection. Some preliminary results of this work have been published in [22].

7. Conclusion

In this paper, we have investigated the probabilistic approach to distributed event detection in sensor networks. We empower the users to define the requirement on desirable detection latency of event detection. The system guarantees that the detection latency of any event is statistically bounded by the latency requirement by the users. The probabilistic paradigm allows each sensor to tune its wakeup frequency and hence minimize its power dissipation. It also finely solves the overdetection problem. The developed algorithm is completely distributed, being scalable up with increasing network scale and sensor deployment density. In addition, it supports fine-grained differentiation of event detection throughout the sensing field. Comprehensive simulation experiments demonstrate that the algorithm remarkably prolongs the functional network lifetime and introduces minimal communication overhead.

Acknowledgments

This research is supported by Shanghai Pu Jiang Talents Program (10PJ1405800), NSFC (no. 61170238, 60903190, 61027009, 60970106, 60673166, and 60803124), Shanghai Chen Guang Program (10CG11), 973 Program (2005CB321901), MIIT of China (2009ZX03006-001-01 and 2009ZX03006-004), Doctoral Fund of Ministry of Education of China (20100073120021), and 863 Program (2009AA012201 and 2011AA010500). In addition, it is partially supported by the Open Fund of the State Key Laboratory of Software De-velopment Environment (Grant no. SKLSDE-2010KF-04), Beijing University of Aeronautics and Astronautics.

References

[1] A. Arora, P. Dutta, S. Bapat et al., “A line in the sand: a wireless sensor network for target detection, classification, and tracking,” *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[2] C. Gui and P. Mohapatra, “Power conservation and quality of surveillance in target tracking sensor networks,” in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom ’04)*, pp. 129–143, October 2004.

[3] M. Li and Y. Liu, “Underground coal mine monitoring with wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 5, no. 2, article 10, 2009.

[4] S. Pattem, S. Poduri, and B. Krishnamachari, “Energy-quality tradeoffs for target tracking in wireless sensor networks,” in *Proceedings of the 2nd International Workshop Information Processing in Sensor Networks (IPSN ’03)*, vol. 2634 of *Lecture*

Notes in Computer Science, pp. 32–46, Palo Alto, Calif, USA, 2003.

[5] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI '06)*, pp. 381–396, 2006.

[6] Y. Zhu, Y. Liu, L. M. Ni, and Z. Zhang, "Low-power distributed event detection in wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 2401–2405, May 2007.

[7] D. Tian and N. D. Georganas, "A node scheduling scheme for energy conservation in large wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 271–290, 2003.

[8] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 28–39, November 2003.

[9] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 51–62, November 2003.

[10] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems (ICDCS '03)*, pp. 28–37, May 2003.

[11] Y. Zou and K. Chakrabarty, "A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks," *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 978–991, 2005.

[12] P. Dutta, M. Grimmer, A. Arora, S. Bibykt, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 497–502, April 2005.

[13] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 20–27, April 2005.

[14] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 144–158, October 2004.

[15] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 320–332, 2010.

[16] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[17] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, November 2004.

[18] C. F. Chiasserini and R. R. Rao, "A distributed power management policy for wireless ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1209–1213, September 2000.

[19] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava, "Estimating clock uncertainty for efficient duty-cycling in sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 130–141, San Diego, Calif, USA, 2005.

[20] Y. Gu, T. He, M. Lin, and J. Xu, "Spatiotemporal delay control for low-duty-cycle sensor networks," in *Proceedings of the Real-Time Systems Symposium (RTSS '09)*, pp. 127–137, December 2009.

[21] Y. Zhu and L. M. Ni, "Probabilistic approach to provisioning guaranteed QoS for distributed event detection," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1265–1273, April 2008.

[22] Y. Zhu and L. M. Ni, "Probabilistic wakeup: adaptive duty cycling for energy-efficient event detection," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07)*, pp. 360–367, October 2007.

[23] T. Armstrong, "Wake-up based power management in multi-hop wireless networks," Tech. Rep., University of Toronto, 2005.

[24] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: a hybrid MAC for wireless sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 90–101, San Diego, Calif, USA, 2005.

[25] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, pp. 1567–1576, June 2002.

[26] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 321–334, November 2006.

[27] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang, "Probabilistic coverage for object tracking in sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM '04)*, Philadelphia, Pa, USA, October 2004.

[28] T. He, S. Krishnamurthy, J. A. Stankovic et al., "Energy-efficient surveillance system using wireless sensor networks," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 270–283, 2004.

[29] C. F. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 433–442, April 2004.

[30] H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: self-organization of sensor networks for efficient query execution," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '03)*, pp. 189–200, June 2003.

[31] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: coverage, connectivity and diameter," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 1073–1083, April 2003.

[32] M. Li and Y. Liu, "Iso-Map: energy-efficient contour mapping in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 699–710, 2010.

[33] M. Li, Y. Liu, and L. Chen, "Nonthreshold-based event detection for 3D environment monitoring in sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 12, pp. 1699–1711, 2008.