2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

Sleep Scheduling of Wireless Sensor Networks Using Hard-core Point Processes

Hassan Jaleel and Magnus Egerstedt

Abstract— **In this paper, we examine a randomly deployed sensor network and propose a probabilistic sleep-scheduling scheme to conserve energy while maintaining a desired level of coverage by using coordination among sensors. To introduce coordination among neighboring sensors, we use a hard-core point process from stochastic geometry, which is an inhibition process and does not allow more than one sensor to be on in a given circular region. As compared to existing schemes, the hard-core point process allows coordination between sensors with little communication overhead. We analyze the proposed scheme and derive an expression for the event detection probability. We also perform Monte Carlo simulations and compare the event detection probability to a random switching scheme for a Poisson process of same intensity in order to confirm that the proposed scheme improves the detection probability.**

I. Introduction

Sensor networks are widely used for monitoring and surveillance purposes. These networks can, for example, be used to detect intruders in sensitive areas or deployed to monitor potentially hazardous environments, e.g., to detect fires in a forest or oil spillage in the ocean. Since these networks are often deployed in inaccessible or hazardous regions, it is desirable to deploy a large number of small, low-power, and cheap devices that can be dropped from a plane flying over the region or from a ground vehicle driving across it. Inasmuch as sensor networks are supposed to operate for long periods of time, power management becomes crucial (e.g., [1] and [3]) because once these devices are deployed, it is almost impossible to replace batteries or perform maintenance. Therefore, it is critical to manage the available power in the best possible manner in order to maximize the lifetime of the sensor network.

A common technique to conserve energy is to add redundancy by increasing the intensity of sensor deployment, where intensity is the expected number of sensors in a unit area, such that only a subset of sensors is sufficient to maintain the required coverage level [5]. Thus, it is not necessary for all the sensors to be on in unison and we can switch sensors off in order to conserve energy. However, it is important to schedule the sensor switching intelligently as critical events can be missed if all the corresponding sensors are off. There is a long list of such scheduling schemes that are available in the literature, e.g., ([2], [4], [5], [9], [10], [11], [13], [15], and [18]) just to name a few. Most of the schemes that have been proposed can be divided into two broad classes: random switching schemes and coordinated switching schemes (e.g., [5] and [15]). It is important to point out that comparing random and coordinated switching schemes is rather difficult. In random switching, each sensor decides to switch its state randomly according to some probabilistic rule regardless of the states of the other sensors. The clocks of the sensors are asynchronous and there is no communication cost involved in scheduling, which makes these schemes efficient in terms of power consumption. However, random switching schemes cannot guarantee complete coverage unless all the sensors are on. In contrast, in coordinated switching schemes, sensors either communicate with their neighbors or acquire exact information about the state and location of their neighbors in order to decide whether to switch on or off. These schemes incur additional communication costs because sensors have to communicate with each other to make switching decisions. However, coordinated scheduling permits the designers to generate various sensor switching patterns depending on the applications. Moreover, these schemes can generally ensure complete coverage of the area of interest (e.g., [9] and [12]). In conclusion, both schemes have their merits and demerits and the decision has to be made based on the application.

In this paper we are presenting a novel sleep scheduling scheme that is a compromise between coordinated and random switching. The proposed scheme is coordinated since sensors communicate with their neighbors for making switching decisions. In order to introduce coordination among the neighboring sensors, we use the concept of a hard-core point process from stochastic geometry, which is an inhibition process that maintains a certain minimum distance, d, between its constituent points and in this way limits the number of redundant sensors covering any area [8]. However, information that is communicated between sensors for coordination consists of randomly generated numbers, which introduces randomness in the switching decisions. By communicating random numbers between the neighboring sensors instead of actual information we can only achieve partial coverage. On the other hand, this coordination improves the coverage as compared to random switching with little communication overhead.

In the literature, several coordinated scheduling schemes are available (e.g., [5], [9], and [17]). In most of the existing schemes, a sensor decides whether to turn on or off based on the exact location of itself and that of its neighbors and this information is made available through a GPS. These schemes make sure that turning a particular sensor off does not deteriorate the coverage, which is important especially

Authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332; Emails: {*hjaleel,magnus*}@*ece.gatech.edu*.

This work was sponsored in part by a grant from AFOSR.

for time critical events that must be detected immediately. However, the exact information about the location of a sensor and its neighbors, which is used to make decisions, is not always available. Even if it is available, it is a strain on the battery of a system as GPS devices consume significant power. In contrast, our proposed scheme does not require any such information related to the location of sensors. Moreover, our scheme allows any level of partial coverage by varying the control parameter d. This partial coverage is still useful in scenarios when we have persistent events that can be detected with some time delay. Thus, in applications where we can tolerate some delay in the detection of events, even a partial coverage can fulfil the purpose (e.g., [12] and [16]).

II. System Description

Consider a domain $\mathcal{D} \subset \mathbb{R}^2$ in which a large number of sensors have to be deployed for monitoring purposes. We assume that all the sensors are identical, i.e., they have the same initial power levels and same sensing capabilities. The footprint of each sensor is a ball of radius r, $B(x, r)$, where x is the location of the sensor, and a sensor can detect anything within its footprint, but nothing outside it. The communication range (R_c) of each sensor is twice that of its sensing range, i.e., $R_c = 2r$. The size of the domain, $\|\mathcal{D}\|$, is very large as compared to the footprint of an individual sensor in order to avoid any boundary effects. Each sensor can switch between on and off states only at discrete time instances $k\Delta t$, where Δt is the sampling interval. A sensor can sense only when it is in the on state and also power is consumed only when a sensor is on.

In order to deploy a sensor network, one approach is to place each sensor at a particular location in a deterministic manner where the location of each sensor is calculated using some algorithm. This approach is useful because it allows network designers to deploy all sensors at optimum locations by solving an optimization problem with a suitable cost function under feasible constraints. However, the optimization problem becomes intractable when the number of sensors becomes very large. Moreover, for a huge number of sensors it becomes physically impossible to place each sensor at exact location. Another approach that is commonly used for the deployment of a large number of sensors is random deployment in which sensors can either be dropped in an area of interest from a plane flying over the field or from a ground vehicle driving across it. One major advantage of random deployment is its simplicity since no planning is required prior to deployment. Another advantage is that it allows us to use the well-developed theory of spatial point processes to analyze any sensor network [7].

Definition 2.1: [8] A point process Φ on $\mathbb{R}^2$ is a random sequence of points,

$$\Phi = \{x_1, x_2, \ldots\} \qquad x_i \in \mathbb{R}^2,$$

that is locally finite and simple.

Locally finite means that any bounded subset of $\mathbb{R}^2$ contains only a finite number of points of Φ and *simple* means that two points cannot overlap, i.e., $x_i \neq x_j$ if $i \neq j$.

In this paper we assume that a large number of sensors are randomly deployed in $\mathcal{D}$ with intensity λ such that the location of each sensor is uniformly distributed over the domain $\mathcal{D}$ and is completely independent of the locations of all other sensors. This deployment scheme implies that the location of each sensor in $\mathbb{R}^2$ corresponds to a point in a point process. From stochastic geometry [8], we know that if the exact number of deployed sensors, N, is known then the sensor deployment scheme can be modeled as a binomial point process. However, if the number of sensors increases (goes to ∞ in the limiting case) in such a manner that $N/\|\mathcal{D}\| \to \lambda$, where $\|\mathcal{D}\|$ is the area of domain $\mathcal{D}$ and λ is the expected number of sensors per unit area, then the sensor deployment can be modeled as a Poisson point process.

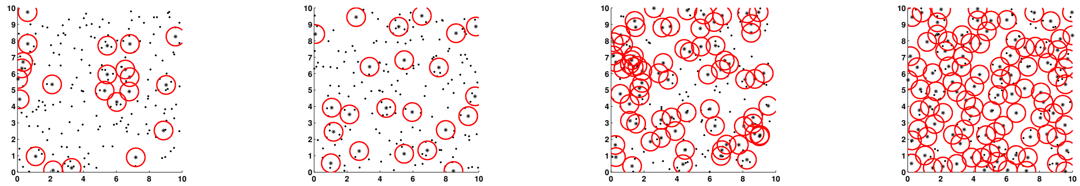
Definition 2.2: [6] A point process Φ is a Poisson point process if it satisfies the following two conditions

1) P $(\bigcup_i(\phi(B_i) = n_i)) = \sum_i P(\phi(B_i) = n_i)$, where $B_1, B_2, \ldots, B_n$ are disjoint subsets of $\mathbb{R}^2$ and $\phi(B_i)$ is the number of points of Φ in the set B_i.
2) For any set $B \subset \mathbb{R}^2$, $\phi(B)$ is Poisson distributed.

Moreover, if λ is constant throughout the area, then we have a homogeneous Poisson process. Once it is established that the sensor deployment is modeled as a stationary Poisson point process with some intensity λ, then the number of sensors in any region of area A can be determined using Poisson distribution.

$$P_n(A) = \frac{(\lambda A)^n e^{-\lambda A}}{n!}. \tag{1}$$

For the purpose of analysis, we assume that the number of sensors goes to ∞. However, the analysis is still valid for any practical network with a sufficiently large number of sensors as is verified through Monte Carlo simulations. The reason for deploying a large number of sensors is that it allows the individual devices to be cheap with low sensing and computing capabilities and low-power requirements [1]. This redundancy in the number of deployed sensors implies that there is no need for all the sensors to be on all the time, and this fact is used quite frequently for increasing the lifetime of the network. Thus, an important task is to design sleep scheduling strategies to govern which sensors should be on at any particular time in order to maintain some desired performance level. As sensor networks are often deployed for monitoring purposes, in this paper we have selected coverage as our performance criterion. To be more specific, we require a sensor network to maintain partial coverage at any particular time, yet the sleep scheduling scheme should ensure that every point in the network must be covered at some point in time. Indeed, this relaxation of partial coverage is practical when we have persistent events whose detection is not time critical because all the events will be detected but possibly after some time delay. In any case, the relaxation allows us to design simple scheduling schemes with small communication cost and simplifies the analysis of the system.



(a) Poisson process q = 0.125 (b) Hard-core process q = 0.125 (c) Poisson process q = 0.4323 (d) Hard-core process q = 0.4323

Fig. 1. Comparison between possible realizations of a sensor network modeled as a Poisson process with random switching (1(a) and 1(c))and a hard-core process with coordinated switching (1(b) and 1(d)).

III. Problem Motivation

In this paper, we consider a sensor network in which a large number of sensors are randomly deployed with intensity λ and this random deployment is modeled as a homogeneous Poisson point process. The objective of the network is to monitor an area of interest by maintaining partial coverage for all the time. A simple approach for achieving this objective is through random switching in which each sensor randomly decides to be in the on state with some given probability q and in the off state with probability $1-q$. For this scheme, the probability that a non-persistent event will be detected is

$$P_d = 1 - e^{-\lambda A q}, \tag{2}$$

where A is the area of the footprint of a sensor and q is the probability of a sensor to be in the on state. Here, we consider a non-persistent event as an event that does not leave a mark in the environment and must be detected when it occurs. Thus, Equation (2) is effectively the level of coverage that is provided by the network. Using Equation (2), we proposed a random switching scheme in [10], in which given any desired coverage level, P_{des}, we first compute the probability of a sensor to be on.

$$q = -\frac{\ln(1 - P_{des})}{\lambda A}. \tag{3}$$

Then each sensor uses this value of q to decide whether to turn on or not and in this way the network maintains the desired coverage. One characteristic of this scheme is that each sensor makes its decision randomly and completely independent of all the other sensors, which keeps the analysis simple because of the Poisson point process. *However, this lack of communication between sensors in decision making can result in more sensors to be on than what are necessary to maintain the desired coverage*. Figures (1(a) and 1(c)) depict possible realizations of a sensor network simulated in Matlab under random switching for two values of q, in which the circles represent the footprints of the sensors that are on. It can be observed that the sensors that are on have formed clusters in certain areas leaving other areas completely uncovered.

The clustering of the sensors in the on state can be avoided by inhibiting any two sensors that are closer than a certain minimum distance to be on simultaneously. We call this minimum allowed distance between the sensors in the on state as the *inhibition distance*. However, to enforce the inhibition distance, sensors have to communicate with their neighbors.

Definition 3.1: The h-neighborhood of a point $x \in \Phi$, where Φ is a point process, is defined as

$$\mathcal{N}_h(x) = \{y : y \in B(x, h) \text{ for all } y \in \Phi\}.$$

The amount of information that is communicated between the sensors is critical because it is directly related to the power consumption but inversely related to the number of redundant sensors. In this paper, we are proposing to use a hard-core point process to accomplish the task of imposing the inhibition distance with little communication overhead.

Definition 3.2: [8] A hard-core point process is a point process in which the constituent points cannot lie together closer than a minimum specified distance.

Figures (1(b) and 1(d)) demonstrate realizations of the same network as in Figures (1(a) and 1(c)) but under the scheme using a hard-core process with inhibition distances $2r$ and r where r is the radius of the footprint. It is apparent that the coverage has improved in the case with the inhibition distance imposed.

IV. Coordinated Sleep Scheduling Scheme

In this section, we propose a new switching scheme in which a sensor communicates with its d-neighbourhood and then decides whether it should turn on or not. We will derive an expression for the probability of a sensor to be on, q, and will later show that for the same q, this scheme has a higher event detection probability than the random switching scheme discussed in Section III. The basic concept that is used is that of a hard-core point process.

In order to decide whether to be in the on or off state at some time instance k, each sensor at location x_i, where $i \in \{1, 2, \ldots\}$, performs the following steps:

Algorithm 1: Proposed scheduling scheme

1) Turns on.
2) Selects a number m_i such that $m_i \sim$ unif $[0, 1]$.
3) Remains on if $m_i < m_j$ for all $j \in N_d(x_i)$.

In the literature on point processes ([7], [8], and [6]), m_i is called a mark of a point i. Under this scheme, two sensors can be on simultaneously if the distance between them is greater than d, which is the inhibition distance.

Lemma 4.1: Under Algorithm 1, the probability of a sensor being on at some time instance k is

$$q = \frac{1 - e^{-\lambda A_d}}{\lambda A_d}, \tag{4}$$

where A_d is the area of a ball with radius d.

Proof: This lemma follows directly from the results in [8], so we only give a sketch of the proof. Since, the switching decisions are made independent of time, so k will have no effect. Now, according to the proposed scheme, a sensor at x_i with a mark m_i remains on if no sensor in $B(x_i, d)$ has a mark lower than m_i. Therefore, the probability $q(m)$ that a sensor at location x with mark m remains *on* is

$$q(m) = \sum_{n=0}^{\infty} \frac{(\lambda A_d)^n e^{-\lambda A_d}}{n!} (p_h)^n,$$

where p_h is the probability that a sensor located in $B(x, d)$ has a mark greater than m. As marks are uniformly distributed in $[0, 1]$, so

$$p_h = \int_m^1 dt = 1 - m,$$

which yields

$$q(m) = e^{-\lambda A_d m}. \tag{5}$$

Because m is uniformly distributed in $[0, 1]$, integrating $q(m)$ from 0 to 1 produces the desired result. ■

From [8], we know that the intensity of a hard-core process that is generated from a Poisson process with intensity λ is

$$\lambda_{hc} = q\lambda, \tag{6}$$

where q is given by Equation (4). Figure (2) demonstrates the relationship between the probability of a sensor to be in the on state and the inhibition distance. It is evident from the figure that increasing the inhibition distance decreases the probability of a sensor to be on which obviously decreases the coverage. Therefore, we can use the inhibition distance as a control parameter to achieve any level of desired coverage. However, to use d as a control parameter, we need to completely characterize the probability of detecting a non-persistent event P_d, which is directly related to coverage, in terms of the inhibition distance. In the next section we will derive an expression for P_d under the proposed scheme for a special case when $d = 2r$, where r is the radius of the footprint of a sensor. The derivation of an expression of P_d for any value of d is a focus of future research in this direction.

A. Event Detection Probability when d = 2r

In this section, we will consider a particular scenario when the inhibition distance is equal to twice the radius of the footprint of a sensor, i.e., $d = 2r$. This value of d will

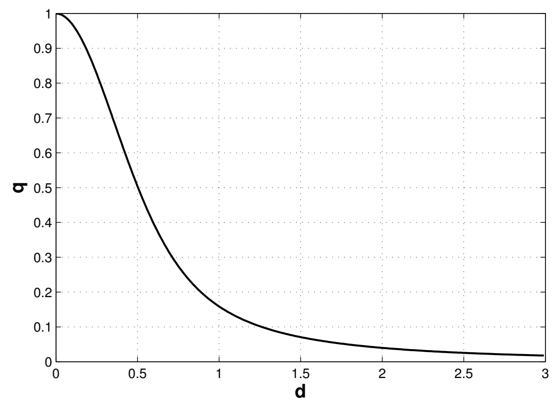


Fig. 2. Variation in the probability of a sensor being on, q, with variation in the inhibition distance, d. Here radius of the footprint of a sensor is $r = 1$.

ensure that there is no overlap between the footprints of sensors, so no redundancy. Although this case is a little restrictive since the level of coverage that can be achieved is not very high, yet this is a good starting point for the complete characterization of this scheme.

Theorem 4.2: Under Algorithm 1 when $d = 2r$, the probability of an event, located at some point $x_e \in \mathcal{D}$, being detected is

$$P_d = -\frac{1}{4} e^{-4\lambda A} - \sum_{n=1}^{\infty} \left(-\frac{1}{3}\right)^n e^{-\lambda A} \sum_{k=0}^{n-1} \frac{(-3\lambda A)^k}{k!}, \tag{7}$$

where A is the area of the footprint of a sensor.

Proof: For the proof of this theorem we assume that an event has occurred at an arbitrary location x_e and we want to find its detection probability. Let us define a random variable Y_i such that

$$Y_i = \begin{cases} 0 & \text{if sensor at } x_i \text{ is off} \\ 1 & \text{if sensor at } x_i \text{ is on} \end{cases} \tag{8}$$

Let m_i be the mark associated with the sensor at x_i. We know that the event will be detected if there is at least one sensor in the on state in $B(x_e, r)$. Since, $d = 2r$, it means that no two sensors having a mutual distance less than $2r$ can be on simultaneously. This implies that at most one sensor can be on in $B(x_e, r)$, which is the sensor having the lowest mark.

$$P_d = \sum_{n=1}^{\infty} \frac{(\lambda A)^n e^{-\lambda A}}{n!} P(\text{sensor with lowest mark is on}). \tag{9}$$

As there are n sensors in the $B(x_e, r)$, we define another random variable X such that

$$X = i \text{ if } \; m_i = \min\{m_1, m_2, \ldots, m_n\},$$

i.e., X is equal to the index of the sensor with the lowest mark in $B(x_e, r)$. Since the mark of any sensor in $B(x_e, r)$ can be the smallest, so

$$P_d = \sum_{n=1}^{\infty} \frac{(\lambda A)^n e^{-\lambda A}}{n!} \sum_{i=1}^{n} P(Y_i = 1 | X = i) P(X = i).$$

For notational convenience, we represent $P(Y_i = 1|X = i)$, which is a function of m_i, with $P_{on}(m_i)$. Now, the random variable X will be equal to i if the marks of all the other $n - 1$ sensors in $B(x_e, r)$ are greater than m_i. Thus,

$$P_{on}(m_i) P(X = i) = \int_0^1 P_{on}(m_i)(1 - m_i)^{n-1} dm_i.$$

From Equation (5), we know that a sensor at x_i with mark m_i is on with probability $e^{(-\lambda A_d m_i)}$, where $A_d = \pi(2r)^2 = 4A$. Figure (3) illustrates that for a sensor at x_i to be on there should be no sensor with a mark lower than m_i in $B(x_i, 2r)$. We already know that the sensor at x_i has the lowest mark in $B(x_e, r)$, which is the shaded region in Figure (3). Therefore, we only need to confirm that there does not exist any sensor with a mark lower than m_i in the unshaded region, whose area is $\pi(2r^2) - \pi r^2 = 3A$. Therefore, we use Equation (5) to find the probability that a region of area $3A$ has no mark less than m_i.

$$P(Y_i = 1 | X = i) = e^{-3\lambda A m_i},$$

which implies

$$P_d = \sum_{n=1}^{\infty} \frac{(\lambda A)^n e^{-\lambda A}}{n!} \sum_{i=1}^{n} \int_0^1 e^{-3\lambda A t_i} (1 - t_i)^{n-1} dt_i.$$

Integrating m_i from 0 to 1 yields the same result for all i, which indicates that the same quantity is being added n times.

$$P_d = \sum_{n=1}^{\infty} \frac{(\lambda A)^n e^{-\lambda A}}{n!} n \int_0^1 e^{-3\lambda A t} (1 - t)^{n-1} dt.$$

Solving this integral and performing some simple algebraic manipulations yields the desired result.

■

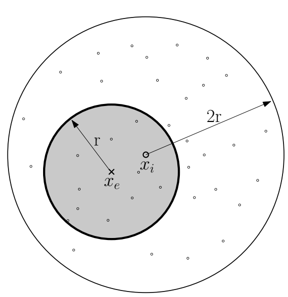


Fig. 3. x_e is the location of the event and x_i is the location of the sensor with lowest mark in the shaded region.

In the next section, we will show through Monte Carlo simulations how different coverage levels can be achieved by varying the inhibition distance. It is important to remember that the probability of detecting a non-persistent event corresponds to the coverage provided by a sensor network.

B. Simulation

To validate the results proved in Section IV, we perform Monte-Carlo simulations of the sensor network. For the simulations, we consider a rectangular domain of dimensions [10 x 10] having an area $A_t = 100$. In this domain, sensors are scattered with an intensity of 2 sensor per unit area, i.e., $\lambda = 2$, which implies that the expected number of sensors in the total area is $\Lambda = A_t \lambda$. The footprint area of each sensor is $A = 1$, so that it is very small as compared to the total area. Events are generated randomly at each time instance throughout the domain and we consider total of 20,000 events in a single iteration of the simulation. To increase the accuracy of the results even further, the values of P_d and P_d are averaged over 100 iterations of the simulation. To ascertain the effects of the inhibition distance on the coverage, we consider four different values of d, i.e., $d \in \{r/2, r, 3r, 2r\}$. For each of these values of d, we compute the intensity of the corresponding Hard-core point process $\lambda_{hc} = q\lambda$, where q is given by Equation (4).

We start by selecting $d = r/2$, apply the coordinated switching scheme proposed in Section IV, and measure the detection probability, P_d. Next, we use the value of q corresponding to $d = r/2$ and measure the detection probability P_d under random switching. The same process is repeated for the remaining values of d. Figures (4 and 5) show the results of the simulations. The simulations demonstrate that as d increases, the detection probability decreases exactly as we expected. Moreover, for all the cases, detection probability under the proposed scheme is better than the random scheme which validates our scheme. Finally, we consider the special case of $d = 2r$ and compare the detection probability under proposed scheme with the value computed analytically. The simulation yields $P_d = 0.2527$, very close 0.2498, which is the value computed analytically, indicating the validity of our analysis

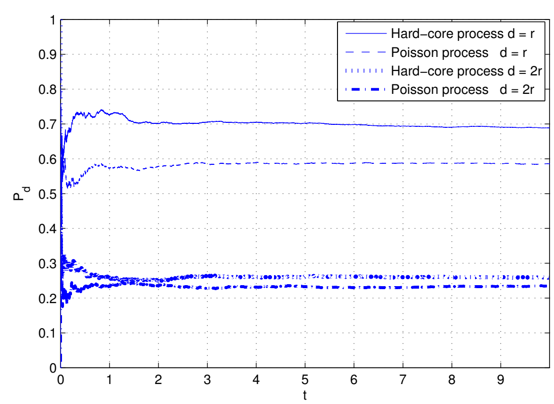


Fig. 4. Comparison of event detection probability between the proposed scheme and the random switching scheme for the inhibition distances $d = r$ and $d = 2r$. For this simulation $\lambda = 2$ and $A = 1$.

V. CONCLUSION

In this paper we presented a novel sleep scheduling scheme for wireless sensor networks to conserve power while

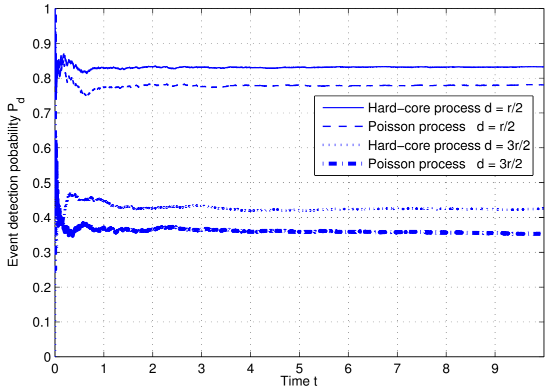


Fig. 5. Comparison of event detection probability between the proposed scheme and the random switching scheme for the inhibition distances $d = r/2$ and $d = 3r/2$. For this simulation $\lambda = 2$ and $A = 1$.

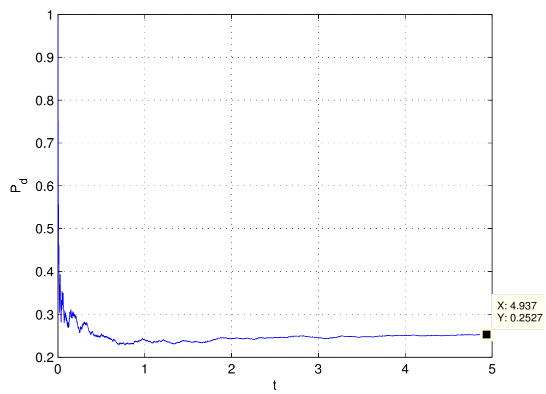


Fig. 6. Event detection probability under the proposed scheme for the case $d = 2r$. It is very close to the analytically computed value, which is 0.2498.

maintaining partial coverage. We introduced the concept of inhibition distance as the minimum distance allowed between the sensors in the on state and used it to control the number of redundant sensors in the on state over the entire domain of interest. To enforce the inhibition distance with little communication overhead, we used hard-core point processes from stochastic geometry and proposed a simple sleep scheduling scheme, which accomplished the task. Then, we considered one special case, in which the inhibition distance was twice the radius of the sensor footprint and derived an expression for the event detection probability in this particular case. Finally we showed through Monte Carlo simulations that the detection probability under our proposed scheme was better than the detection probability under random switching scheme. We also showed that inhibition distance was a valid control parameter for achieving any desired level of coverage.

References

[1] I. F. Akyildiz, E. Cayirci, W. Su and Y. Sankarasubramaniam, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, issue 8, Aug. 2002, pp.102-114.

[2] A. Alfieri, A. Bianco, P. Brandimarte and C.F. Chiasserini, "Maximizing system lifetime in wireless sensor networks," in *European Journal of Operational Research*, vol. 181, issue 1, Aug. 2007, pp. 390 - 402.

[3] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," in *Proc. of the IEEE* , vol. 91, issue 8, Aug. 2003, pp. 1247 - 1256.

[4] R. Subramanian and F. Fekri, "Sleep scheduling and lifetime maximization in sensor networks: Fundamental limits and optimal solutions," in *Proc. IPSN*, ACM Press, 2006, pp. 218 - 225.

[5] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications.*, 2002, pp. 32 - 41.

[6] P. Hall, *Introduction to the Theory of Coverage Processes*, New York, Wiley, 1988.

[7] A. Karr, *Point processes and their statistical inference*, 2nd Ed. Marcel Dekker Inc., 1991.

[8] D. Stoyan, W. S. Kendall and J. Mecke, *Stochastic Geometry and its Applications*, Wiley (Chichester W. Sussex and New York), 1987.

[9] C. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms," in *Proc. IPSN*, ACM Press, Apr. 2004, pp. 433 - 442.

[10] H. Jaleel, A. Rahmani, and M. Egerstedt, "Probabilistic Lifetime Maximization of Sensor Networks," in *IEEE Tran. on Automatic Control*, Issue 99, 2012.

[11] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. IEEE ICC*, vol. 2, June 2001, pp. 472 - 476.

[12] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang, "Design and Analysis of Sensing Scheduling Algorithms under Partial Coverage for Object Detection in Sensor Networks," in *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 3, pp. 334-350, Mar. 2007

[13] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *Proc. ICDCS*, 2003, pp. 28 - 37.

[14] V. Rodoplu and T. Meng, " Minimum energy mobile wireless networks," in *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 8, pp. 1333 - 1344, 1999.

[15] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare event detection," in *Proc. IPSN*, Apr. 2005.

[16] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proc. of ACM Mobicom*, 2004.

[17] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. of ACM MobiCom*, Rome, Italy, July 2001.

[18] Q. Dong, "Maximizing system lifetime in wireless sensor networks," in *Proc. IPSN*, 2005.