Network Coverage Using Low Duty-Cycled Sensors: Random & Coordinated Sleep Algorithms*

Chih-fan Hsin
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI
chsin@umich.edu

Mingyan Liu
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI
mingyan@eecs.umich.edu

ABSTRACT

This paper investigates the problem of providing network coverage using wireless sensors that operate on low duty cycles (measured by the percentage time a sensor is on or active), i.e., each sensor alternates between active and sleep states to conserve energy with an average sleep period (much) longer than the active period. The dynamic change in topology as a result of such duty-cycling has potentially disruptive effect on the operation and performance of the network. This is compensated by adding redundancy in the sensor deployment. In this paper we examine the fundamental relationship between the reduction in sensor duty cycle and the required level of redundancy for a fixed performance measure, and explore the design of good sensor sleep schedules. In particular, we consider two types of mechanisms, the *random sleep* type where each sensor keeps an active-sleep schedule independent of another, and the *coordinated sleep* type where sensors coordinate with each other in reaching an active-sleep schedule. Both types are studied within the context of providing network coverage. We present specific scheduling algorithms within each type, and illustrate their coverage and duty cycle properties via both analysis and simulation. We show with either type of sleep schedule the benefit of added redundancy saturates at some point in that the reduction in duty cycles starts to diminish beyond a certain threshold in deployment redundancy. We also show that at the expense of extra control overhead, a coordinated sleep schedule is more robust and can achieve higher duty cycle reduction with the same amount of redundancy compared to a random sleep schedule.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Reliability, availability, and serviceability

*This work was supported in part by the Engineering Research Centers Program of the National Science Foundation under NSF Award Number EEC-9986866.

IPSN'04, April 26–27, 2004, Berkeley, California, USA.

General Terms

Algorithms, Design, Performance

Keywords

Coverage, superposition of alternating renewal processes, energy efficiency, sensor networks

1. INTRODUCTION

Wireless sensor networks are collections of a large amount of small devices equipped with integrated sensing and wireless communication capabilities, and are expected to find widespread use in a variety of applications. These sensors are operated on battery power, and energy is not always renewable due to cost, environmental and form-size concerns. This places a hard, stringent energy constraint on the design of the communication architecture, communication protocols, and the deployment and operation of these sensors. Due to such constraints, it is desirable to operate wireless sensors at a low *duty cycle* – the fraction of time the sensor is on (e.g., have a sensor turned on/active for only 1% of the time) – in order to prolong the lifetime of the sensor. However, alternating sensors between on and off (active and sleep) states inevitably disrupts the network operation, e.g., coverage and connectivity. In order to compensate for potential performance degradation due to such disruption, redundancy in sensor deployment is usually added. Intuitively, the more redundancy there is, the more we can reduce the duty cycle for a fixed performance measure. For a given level of redundancy, how much the duty cycle can be reduced depends on the design of the duty cycling of the sensors, i.e., when to turn the sensors off and for how long. Naturally we would like to achieve the same performance using the lowest possible duty cycle for the same deployment. A principal question of interest is the fundamental relationship between the amount of reduction in duty cycle that can be achieved and the amount of deployment redundancy that is needed for a fixed performance criteria (e.g., is this relationship linear – do we get to halve the duty cycle by doubling the deployment?).

We will examine this fundamental relationship within the context of providing network coverage using low duty-cycled sensors for surveillance purposes. Strictly speaking, turning the sensory device off and turning the radio transceiver off are two different issues and have different implications. The former results in intermittent sensing capability (i.e., events can go undetected while the sensor is off), and the latter results in intermittent communication capability[1]. For simplicity in this study we will assume that

[1]Usually it is more important to duty cycle the radio transceivers

the sensory device and the radio are simultaneously duty cycled. We assume that a number of sensors are randomly deployed over a field, each alternating between on and off states. We are interested in constantly monitoring the sensing area for certain events, e.g., intrusion, etc.

Specifically, we will study the design of *random sleep* schedules, whereby each sensor enters the sleep state (turned off) randomly and independent of each other, and *coordinated sleep* schedules, whereby sensors coordinate with each other to decide when to enter the sleep state and for how long. An obvious advantage of the random sleep approach is its simplicity, as no control overhead is incurred. On the other hand, using coordinated sleep leads to a better controlled effective topology and is thus more robust and can adapt to the actual deployment. The price we pay is the overhead and energy consumed in achieving such coordination. Moreover, coordinated sleep schemes are much harder to analyze and optimize. In general it is not clear whether it is better to have a more elaborate sleep coordination scheme (thus a more energy-consuming one) or better to have a more simplistic sleep mode (controlled by simple timers). The performance-energy trade-off could also lie in some combination of these two schemes. The ultimate answer is likely to depend on both the application and the design of the sleep/active mechanisms. In this paper these two schemes are separately studied and comparisons are made.

If a sensor node is equipped with more than one radio, then one of them can be used as a paging channel that is turned on periodically while the other radio used for data communication can be kept off. To communicate with a sleeping neighbor, a node sends out periodic paging signals or beacons on the paging channel until it is picked up by the neighbor. An example of system designed based on this assumption can be found in [1, 2]. While having two radios significantly simplifies the sleep/active design for data communication, how much benefit it would provide in the coverage problem is less clear. In addition, it does require the presence of a second radio which may not always be available. In this study we will focus our attention on the single-radio scenario and consider random and coordinated sleep schemes under this assumption.

The rest of the paper is organized as follows. Section 2 presents our network model and performance measures. In Section 3 we provide the coverage analysis under random sleep schemes. Section 4 presents a coordinated sleep scheme along with simulation evaluations. Related work is reviewed in Section 5 and Section 6 concludes the paper.

2. NETWORK MODEL AND PERFORMANCE MEASURES

We assume that static sensors are deployed in a two-dimensional field as a stationary Poisson point process with intensity λ. Thus given any area A, the probability that there are m sensors in this area is $\frac{(\lambda A)^m e^{-\lambda A}}{m!}$. Alternatively, one may consider a network with a fixed number of sensors where the node density is λ, and the sensing range of an individual sensor is very small compared to the area of the network. Thus the probability of having m sensors in a small area A is well approximated by $\frac{(\lambda A)^m e^{-\lambda A}}{m!}$. We will use the Boolean sensing model and assume that the sensing area of each sensor is a circle with radius r centered at the location of the sensor. The sensing/detection is binary, i.e., any point event E that occurs within this circular area can be detected by that sensor if it is active, and cannot be detected if it is outside the circular area. This is a rather simplistic assumption on the sensing device, but nevertheless allows us to analyze the problem of interest and to obtain insight. Equivalently, any point event E can be detected if and only if there is a sensor that lies within a circle of radius r of this event. The probability that there are m sensors that can detect an arbitrary point event is

$$P(\text{m detecting sensors}) = \frac{(\lambda\pi r^2)^m e^{-\lambda\pi r^2}}{m!}. \qquad (1)$$

The average proportion of time that the sensor spends in sleep state is denoted by p, i.e., the duty cycle is $1-p$. p alternatively is also the long term percentage of sensors that are in sleep state in the network, called the *sleep sensor ratio*. We assume that the sensors operate in discrete time and the switching between on (active) and off (sleep) states occurs only at time instances that are integer multiples of a common time unit called *slot*. This assumption also implies that sensors are clock-synchronized, which needs to be realized via synchronization techniques [3].

There are two key performance measures within the context of coverage. One is the *extensity* of coverage, or the probability that any given point is not covered by any active sensor, denoted by P_u. We will also be interested in the *conditional* probability, denoted by $P_{u|c}$, that a point is not covered by any active sensor given that it *could* be covered (i.e., given that it is within the sensing range of some sensor which happens to be in sleep state). We are interested in this conditional probability because for a given deployment, it reflects the effectiveness of a sleep schedule and is determined by the topology formed by the active sensors at any instance of time. The un-conditioned probability on the other hand also takes into account the quality of the deployment. Since the on-off of the sensors produces a dynamically changing topology, a second performance measure is the *intensity* of coverage, defined as the tail distribution of a given point not covered by any active sensor for longer than a given period of time n, denoted by $P_u(t \geq n)$ or $P_{u|c}(t \geq n)$. Intuitively, the heavier this tail distribution, the more vulnerable the surveillance network since it implies higher probability of a region not being covered for extended periods of time. Coverage extensity has been widely studied within the context of static networks using stochastic geometry for nodes deployed as a Poisson point process. On the other hand, coverage intensity only arises in a low duty-cycled network and to the best of our knowledge has not been studied before.

In the next two sections we will evaluate the use of random sleep schedules and coordinated sleep schedules according to these performance measures. In the first case we derive both these measures analytically, while simulations are used for the second case. The focus of this study is coverage, and we do not formally address the connectivity issue. One may build on the work in [4] to further obtain conditions on p and the transmission radius R that ensures asymptotic connectivity. This is part of our on-going work and will be dealt with elsewhere.

3. RANDOM SLEEP SCHEDULES

Under random sleep schemes, the network is essentially a collection of independent on/off (active/sleep) process, characterized by the distribution of the on/off periods. In what follows we will examine the two performance measures outlined above.

3.1 Coverage Extensity

Given that the long term average sleep ratio of a sensor is p, regardless of the distribution of the on and off periods (assuming they are both of finite mean which is desirable for coverage purpose), the probability that a given point event is not covered by any

since they consume more energy. But turning off the sensory device may also be desirable if it consumes significant amount of energy, e.g., certain type of imaging sensors.

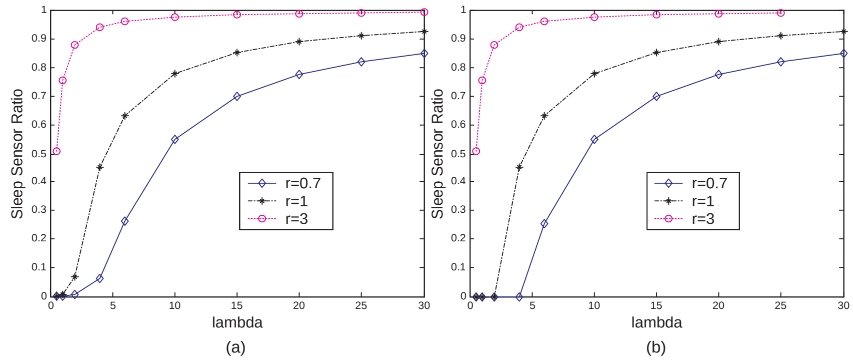

Figure 1: Sleep sensor ratio versus intensity λ under different sensing radius r for (a) $P_{u|c} = 0.001$ and (b) $P_u = 0.001$.

active sensor in a given time slot is

$$P_u = \sum_{n_s=0}^{\infty} p^{n_s} \frac{A^{n_s} e^{-A}}{n_s!} = e^{-A(1-p)} ,$$

where $A = \lambda \pi r^2$ is the expected number of sensors deployed within a circle of radius r around the point event. The associated conditional probability of uncoverage is

$$\begin{aligned} P_{u|c} &= \frac{1}{1-e^{-A}} \sum_{n_s=1}^{\infty} p^{n_s} \frac{A^{n_s} e^{-A}}{n_s!} \\ &= e^{-A(1-p)}(1-e^{-A})^{-1}(1-e^{-Ap}) . \end{aligned}$$

The above equations highlight the relationship between the increase in deployment (λ) and the duty cycle (p) for a fixed coverage measure. They are depicted in Figure 1(a) and (b), respectively, by setting $P_{u|c}$ and P_u to 0.001, respectively, for different values of r. As can be seen in Figure 1(a) and (b), regardless of the value of r, the increase in sleep sensor ratio (or reduction in duty cycle) quickly saturates beyond certain threshold value in λ. This implies that we do not get the same amount of reduction in duty cycle by adding more and more sensors into the network for a fixed performance measure. Beyond certain level of redundancy, there is little that can be gained in terms of prolonging the network lifetime. The threshold value can be obtained based on the preceding analysis. Furthermore, since

$$p = \frac{\log(V(1-e^{-\lambda \pi r^2}) + e^{-\lambda \pi r^2})}{\lambda \pi r^2} + 1$$

for fixed $P_{u|c} = V$, as $0 < \lambda, r < \infty$ and $V \to 0$, $p \simeq \frac{\log(e^{-\lambda \pi r^2})}{\lambda \pi r^2} + 1 = -1 + 1 = 0$ in order to achieve $P_{u|c} = V \to 0$. That is, virtually no sensor can sleep in order to achieve 100% *conditional* coverage.

3.2 Coverage Intensity

We next examine coverage intensity via two special cases, one with geometrically distributed on/off periods (memoryless) and the other with uniformly distributed on/off periods. We then use results from these examples to discuss the design of a random sleep schedule. For the rest of our discussion we will only consider the conditional probability $P_{u|c}$ since this measure focuses more on the the effectiveness of the sleep schedules.

In the first case, a sensor determines independently for each slot to be off with probability p. In the second case, the sleep (off) duration t_s is uniformly distributed within $[M_s - V_s, M_s + V_s]$ and

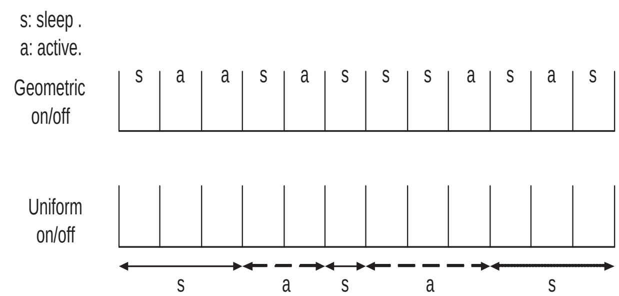

Figure 2: One Realization of Geometric and Uniform On/Off Schedules.

the active (on) duration t_a is uniformly distributed within $[M_a - V_a, M_a + V_a]$, where M_s and M_a are the mean of each. The variance of sleep and active lengths are $\frac{\sum_{i=0}^{V_s}(V_s-i)^2}{2V_s+1}$ and $\frac{\sum_{i=0}^{V_a}(V_a-i)^2}{2V_a+1}$, respectively. The probability that a sensor is asleep during an arbitrary time slot, denoted by p, is $p = \frac{E[t_s]}{E[t_s]+E[t_a]} = \frac{M_s}{M_s+M_a}$. Figure 2 gives us a realization of the geometrical on/off schedule and the uniform on/off schedule.

When the on/off durations are geometrically distributed, the tail distribution that a given point event is uncovered for at least n slots is simply

$$\begin{aligned} P_{u|c}(t \geq n) &= P_{u|c} - \sum_{i=1}^{n-1} P_{u|c}(t=i) \\ &= P_{u|c} - \sum_{i=1}^{n-1} \frac{1}{1-e^{-A}} \sum_{n_s=1}^{\infty} p^{n_s i}(1-p^{n_s}) \frac{A^{n_s} e^{-A}}{n_s!} \\ &= P_{u|c} - \sum_{i=1}^{n-1} \frac{e^{-A}(e^{Ap^i} - e^{Ap^{i+1}})}{1-e^{-A}} . \qquad (2) \end{aligned}$$

When the on/off periods are not memoryless, $P_{u|c}(t \geq n)$ is much more complicated. This is because the tail distribution is essentially determined by the *superposed* on-off process as a result of ORing the individual constituent on-off processes (or alternating renewal processes [5]), i.e., if at least one of the individual on-off processes is on, the superposed process is on, and only when all the constituent processes are off the superposed process is off. We next extend the approach presented in [6] to derive $P_{u|c}(t = n)$ for this case.

Consider $N \geq 2$ independent, identically distributed discrete-

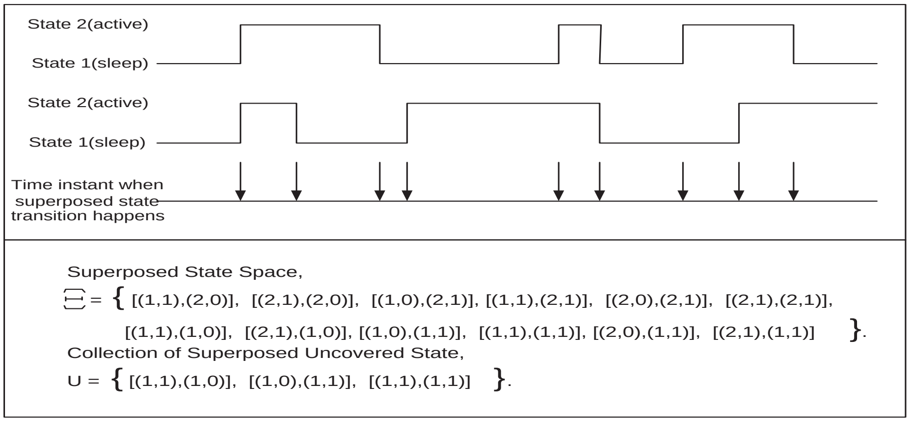

Figure 3: An example when there are $N = 2$ MRP's. State 1/2 is the sleep/active state.

time Markov renewal processes (MRP's) with only 2 states, the sleep state (denoted as state 1) and the active state (denoted as state 2). Each individual MRP *i* is characterized by a semi-Markov kernel $G_i(k) = [g_i(x, y, k)]$ defined over the set of states $1, 2$, where $g_i(x, y, k)$ is the probability that process *i* goes from state *x* to state *y* in *k* slots where $x, y = 1, 2$. Thus,

$$G_i(k) = \begin{bmatrix} g_i(1,1,k) & g_i(1,2,k) \\ g_i(2,1,k) & g_i(2,2,k) \end{bmatrix} = \begin{bmatrix} 0 & f_i^s(k) \\ f_i^a(k) & 0 \end{bmatrix},$$

where $f_i^a(k)$ and $f_i^s(k)$ are the probabilities that the duration of active and sleep periods of process *i* are k time slots, respectively. Define the state transition of the superposition of N such MRP's to occur at time instants whenever one or more of the individual processes experience a state transition. The upper half plot of Figure 3 illustrates an example when there are $N = 2$ MRP's. The superposed states are described by the tuple $[(x_1, t_1), (x_2, t_2), ..$ $..., (x_N, t_N)]$, where $x_i \in \{1, 2\}$ is the state of process *i* observed immediately after a transition occurs, and $t_i \in \{0, 1\}$ indicates whether process *i* has changed state, with $t_i = 1$ iff process *i* has changed state. Ξ is the state space of the superposed process, with the total number of states being $2^N(2^N - 1)$. The bottom half plot of Figure 3 gives us an example of the state space Ξ when $N = 2$.

Suppose that there are N sensors which can detect a given point event. The probability that the event is uncovered is the probability that all N sensors are in sleep state, corresponding to states of the superposed process with tuple $[(x_1, t_1), (x_2, t_2), ..., (x_N, t_N) : x_1 = x_2 = ... = x_N = 1]$. The set of all such states is denoted by U. $C := \Xi \setminus U$ is the set of states under which the event is covered. The bottom half plot of Figure 3 also gives us an example of U when $N = 2$. Let the transition probability from state *u* to state *v* in *k* slots be denoted by $q^N(u, v, k)$, $u, v \in \Xi$. Then $Q = [q^N(u, v, k)]$ is the semi-Markov kernel of the superposed process. Details on the calculation of Q can be found in [6]. Let $P^N(v), v \in \Xi$ be the stationary state distribution of Ξ when there are N MRP's [2]. Given that there are N sensors which can detect an arbitrary event, the probability that the event is uncovered for n time slots, $P_u(t = n|N)$, is given by:

$$P_u(t = n|N) = \sum_{u \in U} \sum_{v \in C} q^N(u, v, n) P^N(u), \forall N \geq 2. \quad (3)$$

It is easy to see that $P_u(t = n|N = 0) = 1$ and $P_u(t = n|N = 1) = f_i^s(n)p$. (Remember that $p = \frac{E[t_s]}{E[t_s]+E[t_a]}$.) Since the sensors are deployed according to a Poisson point process with intensity λ, the conditional probability that the event is uncovered for n slots, $P_{u|c}(t = n)$, is given by:

$$P_{u|c}(t = n) = \frac{1}{1 - e^{-A}} \times \sum_{n_s=1}^{\infty} P_u(t = n|n_s) \frac{(A)^{n_s} e^{-A}}{n_s!}. \quad (4)$$

(Remember that $A = \lambda \pi r^2$.) Note that the above approach can be applied to any discrete-time random sleep schedules that can be modeled as Markov renewal processes.

We verify the correctness of this approach by comparing the results from computation with simulation, both done in Matlab. Chapter 2.6 in [7] states how to simulate a Poisson deployment. The number of sensors detecting a given point event is chosen based on a Poisson stationary distribution. Each sensor performs an iid uniformly-distributed random sleep schedule. Figure 4 shows the comparison for the case where the on/off periods are uniformly distributed. These results are the averages of 10000 random deployments, obtained using $P_{u|c}(t = n) = \frac{C_u(n)}{\sum_m C_u(m)} \frac{T_u}{T}$, where T_u is the total time the event is uncovered, T is the total simulation time, and $C_u(n)$ is the total number of times the event is uncovered for n slots, all computed for the case when there is at least one sensor within the detection range. We see that the analysis matches well with simulation. Same is true for other parameter settings, but the results are not shown here due to space limit.

Now that we have the complete description of both performance measures, we next use the above analysis to explore the design of a good random sleep schedule using the geometric schedule and uniform schedule as examples.

3.3 Design of Random Sleep Schedules

For different sleep schedules that have the same sleep ratio p, the coverage extensity measure is the same, as we have shown earlier. The performance comparison thus lies in the coverage intensity measure $P_{u|c}(t \geq n)$. Figure 5 compares this tail distribution for the geometric sleep schedule and the uniform schedule under a range of parameter settings. Some observations are immediate. Note that the geometric sleep schedule has an infinite tail while the tail of the uniform schedule is limited. On the one hand, the geometric schedule seems to perform better than the uniform schedules (see Figure 5(a)) in that the tail diminishes much faster, when the duty cycle is reasonably high ($1 - p = 0.5$). As the duty cycle

[2] $P^N(v), v \in \Xi$ can be obtained since we can calculate the state transition probabilities from Q.

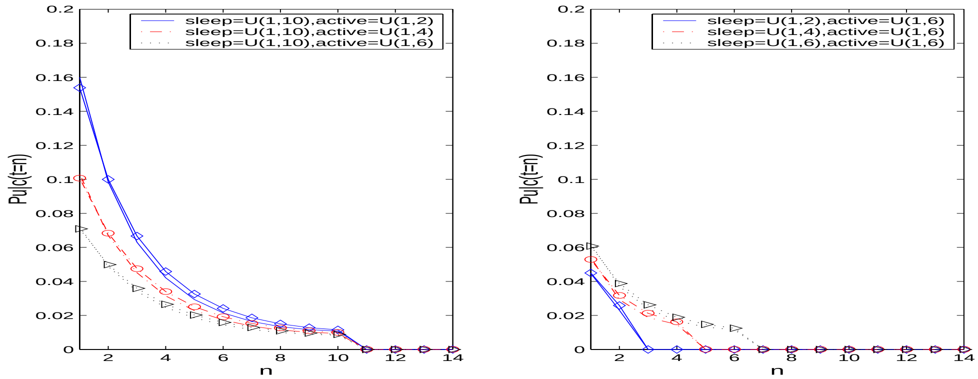

Figure 4: Comparison between analysis and simulation with $\lambda = 1, r = 1$. Results with/without markers are analytical/simulation results.

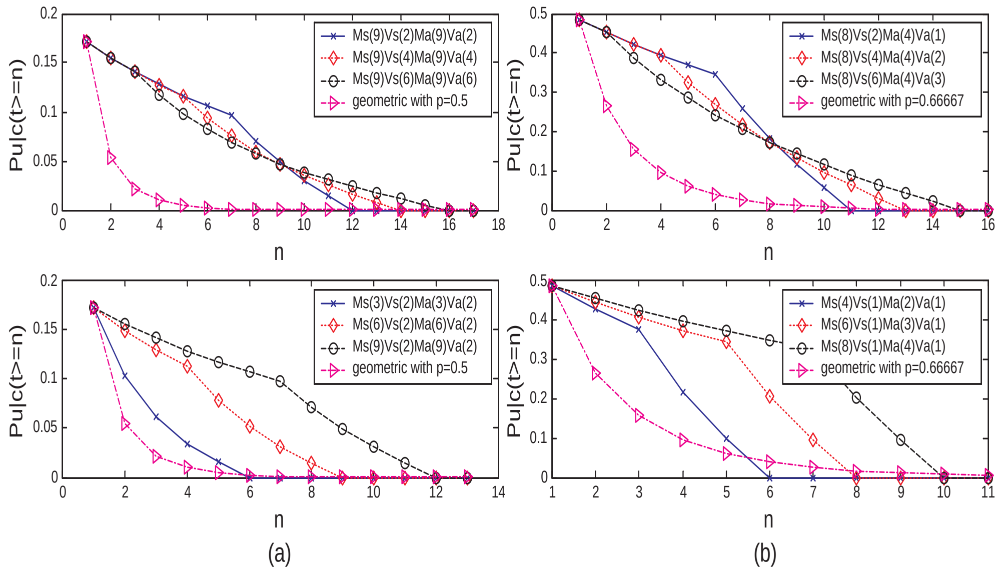

Figure 5: Tail Distribution $P_{u|c}(t \geq n)$ (a) $p = 0.5$, $\lambda = 1$, $r = 1$; (b) $p = 0.66667$, $\lambda = 0.5$, $r = 1$.

decreases ($1 - p = 0.3333$) the comparison is not as straightforward (see Figure 5(b)), in that though the geometric schedule has a smaller tail value, it also lasts longer, and this remains true as we further increase p (decrease the duty cycle).

Within the class of uniform schedules, we see from Figures 5(a) and (b) (upper plots) that with fixed mean M_s and M_a, smaller V_s results in larger $P_{u|c}(t \geq n)$ for $n \leq M_s$ and smaller $P_{u|c}(t \geq n)$ for $n \geq M_s$. With fixed variance (by fixing V_s), smaller M_s results in smaller $P_{u|c}(t \geq n)$, shown in Figures 5(a)(b) (bottom plots).

Note that the duration of one discrete-time slot should not be too small. This is because switching between on and off states itself consumes energy and adds latency [8]. This sets a (device-dependent) threshold value T_{th} below which there is net energy loss, see for example [9]. One can also show that the geometric schedule always results in more on-off transitions/switches than a uniform sleep schedule, for the same sleep ratio p.

Combining the above observations, we can conclude that the geometric sleep schedule may be more desirable if there is a very high coverage intensity requirement since it achieves low $P_{u|c}(t \geq n)$ when n is small. However it may result in higher $P_{u|c}(t \geq n) \neq 0$ when n is large and the duty cycle is relatively low. On the other hand, the uniform sleep schedule guarantees $P_{u|c}(t \geq n) = 0$ for $n > M_s + V_s$, but this probability may be significant for $n < M_s + V_s$. If such guarantee is important, then the uniform sleep schedule is preferable. This argument also applies to any sleep schedule with an upper bound on the sleep duration.

Ultimately the design of such a schedule lies in the specific application requirements. With the approaches outlined here one can easily analyze the tradeoff between different schedules. While random sleep schedules may be attractive for its simplicity and sufficient for some applications, one major drawback is that a sensor's sleep ratio p has to be pre-set and it cannot adjust to the actual density in the network or in its neighborhood. In order for it to be adaptive communication is needed between sensor nodes. In the next section we discuss the design of a coordinated sleep schedule that can dynamically adapt to the network environment to achieve a desirable level of duty cycle.

4. COORDINATED SLEEP SCHEDULES

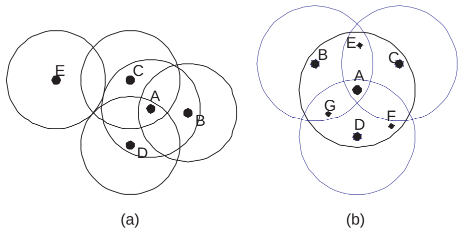

Figure 6: (a) An Example of Redundancy. Small black circles are sensors A, B, C, D, and E. Large circles are the sensing areas of these sensors. (b)The Eligibility Rule.

The basic idea behind coordinated sleep while maintaining high coverage is illustrated in Figure 6(a). In this example Sensor A's sensing coverage area is completely contained in the union set of the coverage areas of Sensors B, C, and D. Consequently Sensor A is completely redundant if Sensors B, C, and D are active, and by turning A off there is no loss of coverage. Sensor A or any other sensor in a similar situation can reach such a decision by learning its neighbors' and its own locations through communication. This idea was explored in [10] and a scheme was developed to conserve energy by turning off redundant sensors, while maintaining good coverage. (The same low-duty-cycling, coverage-preserving problem was also studied by [11, 12, 13], which are compared in Section 5.) The scheme proposed in [10] does not provide continuous coverage since there is no guarantee that two sensors whose redundancy relies on each other do not decide to go to sleep simultaneously. In the example shown in Figure 6(a), A is completely redundant given B, C, and D. At the same time, B could also be redundant given some other nodes not shown in the figure. However, if A and B both go to sleep, there might be an uncovered area until either A or B or both become active again. In addition, this approach did not address how it evolves over time, i.e., how do sensors take turns entering the sleep state and keep balanced energy depletion among sensors.

In what follows we present a coordinated sleep schedule that builds on the same principle shown in Figure 6(a) that achieves continuous coverage and at the same time balances energy depletion to provide better robustness. The idea is a simple one – a sensor decides whether to enter the sleep state not only based on its relative location to its neighbors but also based on its residual energy. We will show that this scheme achieves 100% conditional coverage probability (i.e., $P_{u|c} = 0$) while the reduction in duty cycle is significantly greater as the node density increases compared to random sleep schedules shown in Figure 1.

4.1 A Role-Alternating, Coverage-Preserving, Coordinated Sleep Algorithm (RACP)

We assume that the communication radius is at least as large as the sensing radius. Our algorithm can be easily modified for the scenario when the communication radius is less than the sensing radius. We further assume that each sensor knows its own location either via GPS devices or other localization algorithms (see review in Section 5). Throughout our discussion a neighbor of a sensor is any other sensor within its sensing coverage area.

A sensor assumes multiple *roles* in this scheme, namely, the roles of a *head*, a *sponsor*, and a *regular* node. A sensor informs its neighbors its location by sending coordinate packets COR periodically. COR contains the source ID, the source role, the source coordinates, and the source residual energy. All nodes start as *regular* nodes. A sensor may decide to enter the sleep state after realizing that its sensing area is fully contained by the union set of its neighbors', e.g., Sensor A in Figure 6(a). This sensor is considered *eligible* to enter the sleep state and all its neighbors are potential *sponsors*, e.g., Sensors B, C, and D in Figure 6(a). Eligibility checking is done via basic geometry calculations. A potential *sponsor* formally assumes the role of a *sponsor* when it receives such a request (an REQ packet) of the eligible sensor and confirms it (via an ACK packet). It will then stay active till the sleep time specified by the eligible sensor expires. The eligible sensor, upon receiving confirmation from the sponsors, assumes the role of a *head* and goes to sleep for the specified period of time. If there are no packet losses, then an eligible sensor is guaranteed to receive confirmation from all *sponsors* and safely go to sleep. However, if packets can get lost, then a potential *sponsor* node may not get the initial REQ packet, and the eligible node may not get all ACKs from the *sponsors*. In this case the eligible sensor may simply assume the role of a *head* and go to sleep upon sending an REQ, or do so after receiving ACKs from a subset of the *sponsors*. In the following we will first assume that packets never get lost or corrupted and then address the lossy case toward the end of this section.

Under RACP, an eligible sensor selects only the necessary sensors as *sponsors* instead of all neighboring sensors. For example, in Figure 6(b), Sensor A is eligible and it is sufficient to select either Sensors B, C, D, or Sensors E, F, G to be the sponsor nodes. A will select one set (say B, C, D), and inform these nodes via an REQ packet that contains the source ID, the source role (*head*), the source coordinates, the source residual energy, the *sponsor* list, and the sleep duration. Upon receipt of such information B, C, and D can no long enter the sleep state and they assume the role of *sponsors*. In this case E, F, and G are not the *sponsors* for A and are free to enter the sleep state if they also become eligible.

A *regular* node remains in the active state and checks the eligibility rule described above periodically[3]. If a sensor is not eligible to enter the sleep state, it remains a *regular* node and continuous to check the eligibility rule periodically. If a sensor is eligible to enter the sleep state, it starts a random delay counter. When this counter expires, it broadcasts an REQ packet to its immediate neighbors. After receiving all the confirmation ACK from its *sponsors*, it becomes a *head* and enters the sleep state. If it does not receive all ACKs for a pre-specified period, it remains a *regular* node. If an eligible sensor receives an REQ and becomes another sensor's *sponsor* before the delay counter expires, it stops the counter, assumes the role of a *sponsor*, and replies an ACK. The *sponsors* chosen by a *head* remain in their role and are active until the *head* becomes a *regular* node after the pre-specified sleep duration[4] ends. When this happens, these *sponsors* become *regular* nodes again and perform periodic eligibility checking.

The role alternation is done by the control of the random delay counter. By COR packet exchange, a sensor collects residual energy information about its neighbors. If this sensor is eligible, it then determines its random delay based on the relative residual energy. The more residual energy it has compared to its neighbors, the more likely it selects a longer random delay. There are a variety of ways to achieve this goal. In the simulation evaluation we conducted, a sensor sorts these residual energy values in descending order and divides these values (and the associated nodes) into 3 classes. For example, if the largest residual energy is E_l and the smallest residual energy is E_s, the 3 classes correspond to

[3]The reason that a sensor checks the rule periodically is because of possible topology changes due to sensor death or sensor movement.
[4]The sleep duration is chosen to be large compared to the random delay duration, so that the control overhead does not overwhelm the energy consumption.

residual energy values within the ranges $[E_l, E_l - \frac{E_l - E_s}{3})$, $[E_l - \frac{E_l - E_s}{3}, E_l - \frac{2(E_l - E_s)}{3})$, and $[E_l - \frac{2(E_l - E_s)}{3}, E_s]$. It then associates each class with random delays uniformly distributed with ranges $U(M + 3V, M + 5V - 1)$, $U(M + V, M + 3V - 1)$, and $U(M - V, M + V - 1)$, respectively. Therefore, the eligible sensors with high residual energy are less likely to enter the sleep state than the eligible sensors with low residual energy. Since each sensor maintains its own delay counter, eligibility checking period, and COR packet transmission period, RACP does not require nodes to be synchronized. Due to space limit, we will skip the state diagram and pseudo code of this scheme and present its simulation evaluation next.

4.2 Simulation Evaluation

In this section we present simulation results done in Matlab. We will first look at the scaling properties of the coordinated sleep algorithm RACP introduced above by increasing the deployment redundancy, and then briefly compare this approach with the coverage algorithm (denoted by CA) proposed in [10]. We will also discuss the scenario where packets may get lost.

Sensors are deployed according to a Poisson point process with intensity λ in a square field of dimension $L \times L$. All results presented here are averages over 100 runs. We assume that the communication radius is the same as the sensing radius and packet transmissions are always successful. The following energy model from [14] is adopted. The energy consumption on packet transmission is $\alpha_{11} + \alpha_2 d_{1,2}^2$ J/bit, and the energy consumption on packet reception is α_{12} J/bit, where $\alpha_{11} = 45nJ/bit$, $\alpha_{12} = 135nJ/bit$, $\alpha_2 = 10pJ/bit/m^2$, and $d_{1,2}$ is the distance between the transmitting and receiving nodes. We will assume that there is no energy consumption when a sensor is in the sleep state and the idle energy consumption in the active state is $35mW$ [15]. The sizes of packets COR, REQ and ACK are 13 bytes, 46 bytes, and 19 bytes, respectively.

The coverage probability is determined as follows. The entire field is divided into many small squares with size 0.5×0.5. Point events are scheduled to occur as the center of these squares. Assuming all sensors are active, the total number of such events that can be covered by some sensors are counted at time t, denoted by $N(t)$. Then the coordinated sleep algorithm is applied and the total number of events covered by active sensors are counted, denoted by $N_a(t)$. The instantaneous area coverage ratio is estimated by $\frac{N_a(t)}{N(t)}$. The conditional uncoverage probability $P_{u|c}$ is estimated by the long run average of $1 - \frac{N_a(t)}{N(t)}$.

For RACP, the random delay parameters are set at $M = 6$ sec and $V = 5$ sec, and eligibility checking period is set at 3 sec. The sleep duration is 5 times the expected random delay. The period of the coordinate packet COR equals the sleep duration. For CA, all sensors perform the scheduling algorithm and are turned on/off according to the scheduling algorithm. After the sleep duration, same as the one in RACP, all sensors perform the scheduling algorithm again. All other parameters in CA are set to be comparable with the parameters in RACP.

Figure 7(a) illustrates the sleep sensor ratio that can be achieved under RACP vs. under a random sleep schedule. Here the sleep sensor ratio p is calculated as the ratio between the number of sleep sensors and the number of total sensors averaged over time before the first sensor death. This comparison clearly shows that RACP outperforms a random sleep schedule in that much greater reduction in duty cycle is achieved as the deployment redundancy increases. Note however that the benefit in increasing deployment redundancy saturates again after a certain point. Note that though the two are being compared, they do not represent exactly the same coverage quality. This is because RACP achieves a coverage ratio $\approx 100\%$, i.e. $P_{u|c} \approx 0$, while for the random sleep schedule $P_{u|c}$ is set at 0.000001 (this cannot be made 0 unless $p = 0$ as we discussed earlier). Therefore strictly speaking, the difference between the two is even greater than shown here. It would be very interesting to further investigate what is the best one could achieve in terms of this scaling relationship using a decentralized, coordinated mechanism in a random network.

Figure 7(b) shows how RACP scales with deployment density λ while fixing the average node degree by reducing the sensing range r. The degree $d = \lambda \pi r^2$ is the average number of sensors in the circle of area πr^2. Degree is one of the most important factors in deciding the eligibility of a sensor node. In Figure 7(b), as we increase λ, r is reduced to keep the average degree remain fixed at $d = 6$. As can bee seen, for the most part the sleep ratio increases very slowly as λ increases except for very small λ. The reason is because when λ is small, the total number of sensors in the filed is also small and there is a higher percentage of edge sensors (sensors located toward the edge of the field). As a consequence, the number of sensors eligible to enter the sleep state is small. The edge sensor effect remains an important factor until λ becomes sufficiently large, i.e., when the total number of sensors in the field is sufficiently large.

Figures 8(a) and (b) compare the area coverage ratio under RACP and CA for $L = 50$, $\lambda = 0.06$, and sensing radius $r = 7$ and $r = 10$, respectively. In Figure 8(a) the area coverage ratio of CA oscillates for the reason discussed earlier. RACP on the other hand has almost 100% area coverage ratio till simulation time reaches around 20000, when the area coverage ratios drop suddenly. This is due to the death of some sensors which are always active. (These sensors include the sensors not eligible to enter the sleep state due to the initial deployment and the sensors that are always chosen to be a sponsor, again due to deployment.) Furthermore, RACP has a more graceful degradation of area coverage ratio compared to CA. This is due to the role-alternating function in RACP. Fewer sensors run out of energy simultaneously in RACP than in CA. Note that after time 28000 the area coverage ratio of CA is higher than RACP. This is due to the fact that sensors on average sleep more under CA as a result of the oscillating area coverage ratio. These observations remain true when we further increase r to 10 in Figure 8(b), as well as for other values of L and λ.

Furthermore, we see that the good coverage in RACP comes at the price of an relatively short time-to-first-sensor-death. From Figures 8(a) and (b) the first few sensor deaths for RACP and CA happen at roughly the same time. This is because some sensors are never eligible due to the initial deployment regardless of the coordination algorithm. In addition this time for RACP and CA does not increase significantly with network density λ for the same reason. However, under a random sleep schedule all sensors die at almost the same time and this time increases with λ, since the duty cycle achieved under a predefined coverage measure decreases with λ. Thus for large λ the random sleep schedule has a larger time-to-first-sensor-death than the coordinated schedule presented here, but it comes with a lesser coverage quality.

All the results shown so far assume that all packets are received successfully. Figure 9 shows the simulation results when packet transmission fails with a certain probability p_f, either due to collision or corruption. As can be seen, the area coverage ratio does not drop much for $p_f = 0.1$ compared to that shown in Figures 8(a) and (b). However, when most control packets are lost, e.g. $p_f = 0.5$ in Figure 9, sensors cannot enter the sleep state efficiently and most sensors run out of energy at almost the same time around

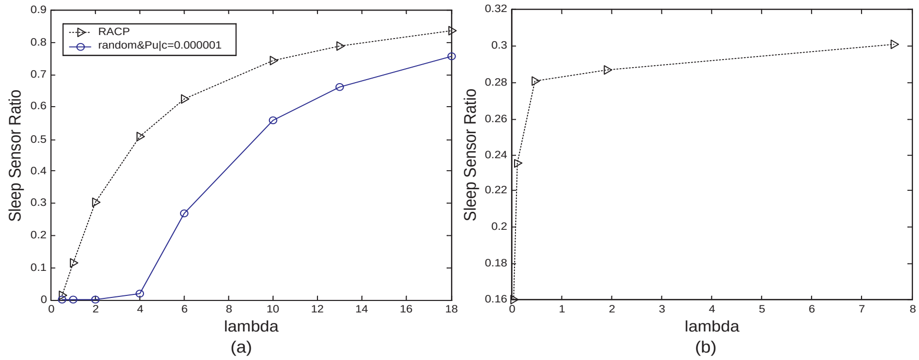

Figure 7: (a) Sleep sensor ratio of RACP compared to random sleep schedules with increasing λ, $L = 20$, $r = 1$; (b) Sleep sensor ratio with increasing λ for fixed degree $d = 6$, $L = 20$.

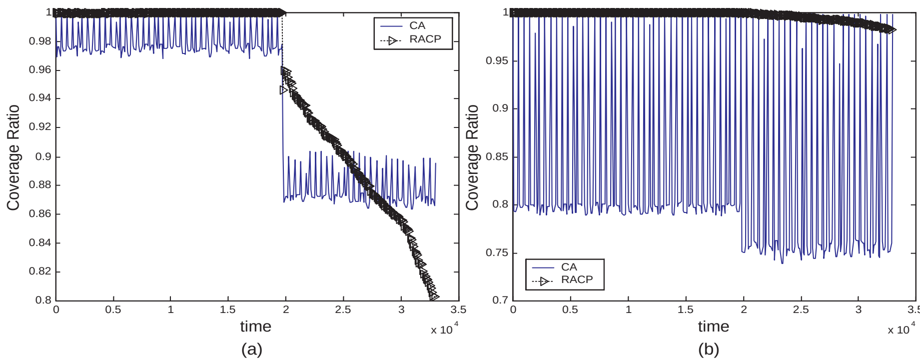

Figure 8: Area coverage ratio when $L = 50$, $\lambda = 0.06$, (a) $r = 7$ and (b) $r = 10$.

20000. This situation can be alleviated by modifying RACP such that a sensor becomes a head right after transmitting REQ instead of after the reception of ACKs. However in this case the area coverage ratio is also likely to oscillate because some sensors may not be informed of their sponsor roles without successfully receiving REQ.

5. RELATED WORKS

Different applications usually induce different definitions and measures of coverage, e.g., see [16] for a number of such measures. A coverage and connectivity problem was considered in [17] based on a grid network where sensors may be unreliable, which is equivalent to sensors entering the sleep state. Conditions on the sensing radius, the network density and the reliability probability were derived to achieve asymptotic coverage and connectivity. Conditions for asymptotic connectivity for a general network were derived in [4].

A related but different problem is the detectability of a network, where the goal is the design of the path along which a target is least/most likely to be detected (known as the worst/best-case coverage). In particular, in [18, 19] the network was modeled as a grid and the shortest path algorithm was used to to find the path with the worst-case coverage and it was also solved using Voronoi diagram in [20]. [20] used Denaulay triangulation to find the best-coverage path and [21] used the local Denaulay triangulation, the relative neighborhood graph, and the Gabriel graph to find the path with best-case coverage.

In this paper we studied the use of duty-cycled sensors to provide coverage. We did not discuss the effect of such duty-cycling on data transmission. Low duty-cycled sensors and their operation in the presence of data transmission have been quite extensively studied. For example, [1, 2] achieved energy saving at the expense of increased data forwarding latency, by turning the radios on and off while keeping a paging channel/radio to transmit and receive beacons. How to turn sensors off in a coordinated way to conserve energy has also been studied in [22, 23]. [1, 2] also studied energy consumption and performance tradeoffs using low duty-cycled sensors.

The sensing model used in this work is the Boolean sensing model, which means that a target can be detected if the distance between the target and any sensor is within the sensing radius r. A more general sensing model may be defined, whereby the target detection is based on the signal strength received from the target of interest. Although the general sensing model represents the real sensory device more accurately, it introduces more analytical complexity than the Boolean sensing model. For example, in the general sensing model we need to consider the environmental noise and the signal attenuation factors; furthermore, the performance metrics may be the probability of detection, probability of false alarms, or simply the probability of errors, which are more complicated than the coverage area metric used in the Boolean sensing model. For this reason, the Boolean sensing model is used widely in literature [11, 10, 12]. As cited before, [10] calculated the overlapping coverage areas. [12] did not calculate the overlapping coverage areas but divided the field into grids. In this approach sensors schedule on/off time to let each grid point covered by at least one sensor at

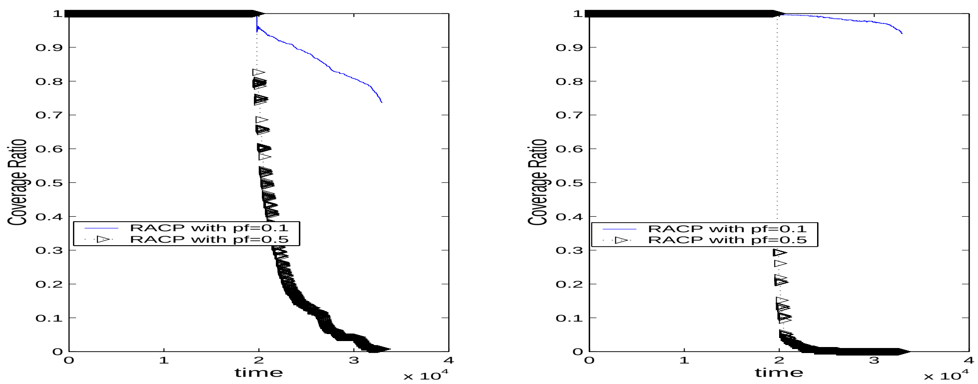

Figure 9: Area coverage ratio in a noisy environment, $L = 50$, $\lambda = 0.06$, (a) $r = 7$ and (b) $r = 10$.

any time. The grid size and the time synchronization skew affect the performance. While [10, 12] proposed distributed algorithms, [11] proposed a centralized approach (Set K-Cover) to turn off sensors. This approach selects mutually exclusive sets (or covers) of sensors, where the members of each cover completely cover the monitored area. Only one of the covers is active at any time. The energy-efficiency is based on the number of covers (K) one can obtain. As the covers are decided in the beginning, 100% coverage cannot be guaranteed due to changes in the network (e.g., sensor removal or death). This can be alleviated by periodically recomputing the covers, which may introduce significant complexity due to the centralized nature of the algorithm. [13] used a PROBE-REPLY approach to schedule on/off time, which can be applied for both the Boolean sensing model and the general sensing model. This approach reduces the computation overhead but may not be appropriate for applications which requires exact coverage guarantee.

The coordinate sleep algorithm RACP proposed here requires the knowledge of sensor location. In addition to GPS, there are many proposed algorithms that estimate sensor locations without the requirement of a GPS device. Some of these algorithms use multilateration, e.g. [24, 25], some use triangulation/trigonometry, e.g. [26], and some use RF connectivity, e.g., [27].

6. CONCLUSION

In this paper we investigated the problem of providing network coverage using low duty-cycled sensors. We presented both random and coordinated sleep algorithms and discussed their design tradeoffs. We showed that using random sleep the amount of reduction in sensor duty cycle one can achieve quickly diminishes beyond a saturation point as we increase the deployment redundancy. Using coordinated sleep algorithms we can obtain greater reduction in duty cycle at the expense of extra control overhead.

7. REFERENCES

[1] Schurgers, C., Tsiatsis, V., Ganeriwal, S., Srivastava, M.: Optimizaing sensor networks in the energy-latency-density design space. IEEE Transactions on Mobile Computing **1** (2002)

[2] Schurgers, C., Tsiatsis, V., Ganeriwal, S., Srivastava, M.: Topology management for sensor networks: Exploiting latency and density. In: ACM International Symposium on Mobile ad hoc networking and computing (MobiHOC). (2002)

[3] Elson, J., Romer, K.: Wireless sensor networks: A new regime for time synchronization. In: The First Workshop on Hot Topics In Networks (HotNets-I). (2002)

[4] Xue, F., Kumar, P.R.: The number of neighbors needed for connectivity of wireless networks. Wireless Networks (2003)

[5] Cox, D.: Renewal Theory. Methuen and Co. LTD Science Paperbacks (1970)

[6] Elsayed, K.M.F., Perros, H.G.: The superposition of discrete-time markov renewal processes with an application to statistical multiplexing of bursty traffic sources. Applied Mathematics and Computation (Kluwer Academic Publisher) (2000)

[7] Stoyan, D., Kendall, W.S., Mecke, J.: Stochastic Geometry and Its Applications, Second Edition. Wiley Series in Probability and Statistics (1995)

[8] Min, R., Bhardwaj, M., Cho, S.H., Shih, E., Sinha, A., Wang, A., Chandrakasan, A.: Low-power wireless sensor networks. In: IEEE Fourteenth International Conference on VLSI Design. (2001)

[9] Sinha, A., Chandrakasan, A.: Operating systems and algorithmic techniques for energy scalable wireless sensor networks. In: Proc. of the second International Conference on Mobile Data Management (ICMDM). (2001)

[10] Tian, D., Georganas, N.D.: A coverage-preserving node scheduling scheme for large wireless sensor networks. In: First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA). (2002)

[11] Slijepcevic, S., Potkonjak, M.: Power efficient organization of wireless sensor networks. In: IEEE INternational Conference on Communications (ICC). (2001)

[12] Yan, T., He, T., Stankovic, J.A.: Differentiated surveillance for sensor networks. In: Proceedings of the first ACM international conference on Embedded networked sensor systems. (2003)

[13] Ye, F., Zhong, G., Lu, S., Zhang, L.: Peas: A robust energy conserving protocol for long-lived sensor networks. In: IEEE International Conference on NEtwork Protocols (ICNP). (2002)

[14] Bhardwaj, M., Chandrakasan, A.P.: Bounding the lifetime of sensor networks via optimal role assignments. In: Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2002)

[15] Intanagonwiwat, C., Estrin, D., Govindan, R., Heidemann, J.: Impact of network density on data aggregation in wireless

sensor networks. In: IEEE International Conference on Distributed Computing Systems (ICDCS). (2002)
[16] Liu, B., Towsley, D.: On the coverage and detectability of large-scale wireless sensor networks. In: In Proc. of the Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks Conference (WiOpt). (2003)
[17] Shakkottai, S., Srikant, R., Shroff, N.: Unreliable sensor grids: Coverage, connectivity and diameter. In: Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2003)
[18] Meguerdichian, S., Koushanfar, F., Qu, G., Potkonjak, M.: Exposure in wireless ad-hoc sensor networks. In: ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM). (2001)
[19] Clouqueur, T., Phipatanasuphorn, V., Ramanathana, P., Saluja, K.K.: Sensor deplyment strategy for target detection. In: First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA). (2002)
[20] Mequerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage problems in wireless ad-hoc sensor networks. In: Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2001)
[21] Li, X.Y., Wan, P.J., Frieder, O.: Coverage in wireless ad hoc sensor networks. IEEE Transactions on Cumputers **52** (2003)
[22] Chen, B., Jamieson, K., Balakrishnan, H., Morris, R.: Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In: ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM). (2001)
[23] Ye, W., Heidemann, J., Estrin, D.: An energy-efficient mac protocol for wireless sensor networks. In: Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2002)
[24] Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a global coordinate system from local information on an ad hoc sensor network. In: The 2nd International Workshop on Information Processing in Sensor Networks (IPSN). (2003)
[25] Savvides, A., Park, H., Srivastava, M.B.: The bits and flops of the n-hop multilateration primitive for node localization problems. In: First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA). (2002)
[26] Niculescu, D., Nath, B.: Ad hoc positioning system (aps) using aoa. In: Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). (2003)
[27] He, T., Huang, C., Blum, B.M., Stankovic, J.A., Abdelzaher, T.: Range-free localization schemes for large scale sensor networks. In: ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM). (2003)