Design and Analysis of Sensing Scheduling Algorithms under Partial Coverage for Object Detection in Sensor Networks

Shansi Ren, *Student Member, IEEE,* Qun Li, *Member, IEEE,* Haining Wang, *Member, IEEE,* Xin Chen, *Member, IEEE,* and Xiaodong Zhang, *Member, IEEE,*

***Abstract*— Object detection quality and network lifetime are two conflicting aspects of a sensor network, but both are critical to many sensor applications such as military surveillance. Partial coverage, where a sensing field is partially sensed by active sensors at any time, is an appropriate approach to balancing the two conflicting design requirements of monitoring applications. Under partial coverage, we develop an analytical framework for object detection in sensor networks, and mathematically analyze average-case object detection quality in random and synchronized sensing scheduling protocols. Our analytical framework facilitates performance evaluation of a sensing schedule, network deployment, and sensing scheduling protocol design. Furthermore, we propose three wave sensing scheduling protocols to achieve bounded worst-case object detection quality. We justify the correctness of our analyses through rigorous proof, and validate the effectiveness of the proposed protocols through extensive simulation experiments.**

***Index Terms*— Sensor Networks, Object Detection Quality, System Lifetime**

I. Introduction

Detecting and tracking moving objects is a major class of applications in sensor networks, such as vehicle detection in military surveillance [10] and wild animal habitat monitoring [17]. These applications, by their nature, enforce certain detection quality and lifetime requirements. The first requirement determines how fast a sensor network should detect the intrusion of a moving vehicle, or how often the data about a wild animal should be sampled and collected. The second requirement specifies the working duration a sensor network should sustain. These two requirements, however, are conflicting optimization goals due to the stringent energy constraints of sensor nodes.

Full sensing coverage is mandatory for sensor monitoring applications that require either immediate response to detected events or information of all points in the sensing field. Full sensing coverage, however, is too expensive to support long-duration monitoring applications. More often those applications do not need zero response time or information at all points of the sensing field. Full sensing coverage provides over-qualified detection quality for these applications at the cost of exhausting network energy rapidly, who may be willing to sacrifice event detection probability or detection delay to some extent for increasing the network lifetime. A relaxed sensing coverage—partial coverage, where the sensing field is partially sensed by active sensors at any time—is a more appropriate approach to balancing object detection quality and battery power consumption.

A partial coverage scheme allows sensor nodes to periodically wake up and go back to sleep. A node in sleep mode cannot sense events; its sensing capability is resumed after it wakes up. Therefore, the sensor network provides only a fraction of the maximal coverage of all the sensors. Battery power, however, is conserved for the nodes in sleep mode. How much time and how frequently a sensor node should stay in active mode determine detection quality and power saving. Our study aims to characterize the interplay among sensor scheduling, detection quality, and power saving.

Detection quality requirements are classified into average-case detection quality requirements and worst-case detection quality requirements. The average-case detection quality can be characterized by the probability that a moving object is detected in a given observation duration, and by the average distance an object travels before detection. In contrast, the worst-case detection quality can be characterized by the lower limit of time duration to detect moving objects, and by the upper limit of distances that objects travel before detection. This paper considers both applications with stringent average-case detection quality requirements and applications with stringent worst-case detection quality requirements.

Our work, together with [7], is the first to propose the concept of partial coverage to meet the required average and worst case object detection quality while minimizing network energy consumption. In the first part of this paper, we consider the fundamental tradeoff between average-case object detection quality and energy consumption under different sensing scheduling schemes, and establish an analytical framework by examining simple random sensing schedules and synchronized sensing schedules. The random sensing scheduling algorithm is simple yet effective in satisfying the required detection quality with less energy consumption than more sophisticated schemes, such as PECAS and Mesh proposed in [7]. On the other hand, the synchronized sensing scheduling algorithm is useful in providing bounds for object detection quality. Taking into consideration the fact that sensors may have

S. Ren and X. Zhang are with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210. E-mail: {sren, zhang}@cse.ohio-state.edu.

Q. Li and H. Wang are with the Department of Computer Science, College of the William and Mary, Williamsburg, VA 23187. E-mail: {liqun, hnw}@cs.wm.edu.

X. Chen is with the Ask Jeeves Inc., Piscataway, NJ 08854. E-mail: xchen@ask.com.

different initial energy, based on our analysis, we design a power-aware sensing scheduling protocol that is highly beneficial to optimizing the network energy consumption. Our analytical framework can not only provide accurate guidelines for sensing scheduling protocol design and optimal sensor network deployment, but also be used to derive the necessary speed of an object wanting to evade sensor detection. Prior to our work, sensing scheduling protocols with object detection quality requirements are mainly designed using heuristics, and are evaluated through simulation. Our work shows that many of these protocols, such as PECAS, are variations of the random schedule or the synchronized schedule, thus, can be incorporated into the analytical framework by setting parameters appropriately. We apply our analytical framework to previously-proposed heuristic sensing scheduling protocols in the literature, and successfully analyze their object detection quality.

The second part of this paper complements the first part with the design of three wave sensing scheduling protocols to provide the bounded worst-case object detection quality with node coordination. In these wave protocols, at any moment, active nodes on the sensing field form connected curves. These curves move back and forth both horizontally and vertically across the field, so that every geographical point is scanned at least once within a given duration. We prove the bounds on worst-case object detection quality of these protocols, and validate their effectiveness through extensive simulation experiments. These protocols are simple yet energy-efficient, and provide bounded detection time, thus, can be used by many monitoring and surveillance applications that demand the stringent worst-case object detection quality. Based on the protocol design, we are able to choose appropriate sensing scheduling parameters to provide the bounded worst-case object detection quality, while optimizing the average-case object detection quality and network energy consumption.

The remainder of this paper is organized as follows. Section II sketches related work. Section III introduces object detection quality metrics. Section IV presents the analysis of the average-case object detection quality of the random and synchronized sensing schedules. Section V details the design and analysis of the three wave sensing scheduling protocols. Finally, Section VI concludes our work.

II. Related Work

Related to object detection, tracking a moving object in sensor networks has been extensively studied from different perspectives: system design and deployment ([11], [17]), maintaining high tracking precision ([2], [8], [28]), utilizing node collaborations ([3], [13], [16], [32], [33]), and reducing energy consumption ([22], [23], [27]). Under a minimalist binary sensor model, Aslam *et al.* [2] provided a particle filter-based tracking scheme. Leader-based tracking schemes have been proposed in [16] and [33], in which a single node is used to track the target. Zhao *et al.* ([16], [33]) proposed leader-based tracking schemes by using the information of a single sensor node. Compared with our work, these studies address the object tracking problem, i.e., how to reduce the differences between the measured object location and its real location, and how to capture the trajectory of a moving object. While our work addresses a different problem, i.e., the detection quality problem under different sensing scheduling protocols. We establish an analytical framework to quantify the probability that an object is detected in a given duration, and to determine the expected distance a moving object travels.

Obviously, a higher degree of sensing coverage gives us an advantage of producing a better quality of object tracking. A large number of sensing coverage maintenance protocols, aiming to conserve energy under various conditions, have been proposed ([1], [6], [12], [15], [29]). Yan *et al.* [29] presented an energy-efficient random reference point sensing protocol to achieve a targeted coverage degree. Nodes decide their active periods by exchanging reference points among neighbors. In [12], Hsin and Liu investigated coverage intensity and extensity of random sleep schedules and coordinated sleep schedules. In [31], Zhang and Hou studied the system lifetime of a k-covered sensor network, and proved that it is upper bounded by k times node continuous working time. In [20] and [21], Onur *et al.* investigated the effect of false alarm rate and path-loss exponent on the quality of deployment using a probabilistic approach, and proposed a method to determine the required number of sensors being deployed under the weakest breach path problem. In [18], Megerian *et al.* proposed the optimal polynomial time worst and average case algorithm for the coverage calculation of homogeneous isotropic sensors, by combining computational geometry and graph theoretic techniques. Liu and Towsley [15] studied the coverage and detectability problem in sensor networks. In summary, these studies all focus on the static coverage of the points on a sensing field. In contrast, our work considers the problem of detecting moving objects, and analyzes object detection quality and system lifetime under an analytical framework. In essence, coverage and object detection are two different problems.

The closest previous work to ours include [4], [5], [7]. Gui and Mohapatra [7] considered the trade-off between power conservation and quality of surveillance in target detection and tracking by using non-full coverage. Although we aim to address similar problems, our work differs from theirs in many aspects. We provide fundamental analytical results on the relationship between object detection quality and network lifetime under different sensing schedules, and on how to utilize the analytical results to direct and evaluate protocol design; while in [7], most protocols are designed using heuristics, and their performance evaluation is based on simulation only. In [4], Cao *et al.* presented an optimized framework for rare event detection that compromises between event detection delay and lifetime while maintaining point coverage. In comparison, the work in [4] is only useful for a specific class of surveillance applications, where events are rare and sensor active ratio is extremely low. While our work provides a more generic analytical framework that can be widely used for sensing scheduling protocol design with detection quality requirements. In [5], Cao *et al.* analyzed target detection quality by investigating some special cases. Compared with [5], in this paper, we give more thorough and complete analysis on both average-

case and worst-case object detection quality. In addition, our work comprehensively characterizes how different sensing scheduling algorithms affect the system lifetime of a sensor network.

III. Object Detection Assumptions, Quality Metrics, and Applications

In this section, we first delineate the assumptions of our object detection model. Then we formally define the average-case and worst-case object detection quality metrics, and describe related object detection applications. This section serves as the basis for our algorithm design and analysis in the following sections.

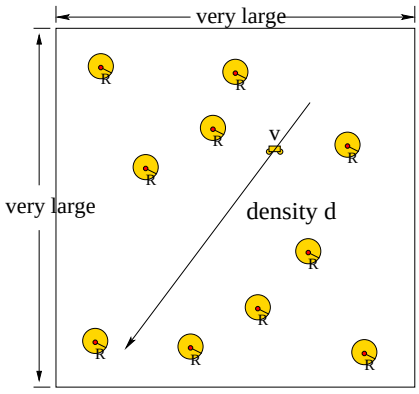


Fig. 1. An object detection and tracking scenario.

Fig. 1 shows a typical scenario of the object detection in a sensor network. There are a number of sensors deployed on a sensing field. A small object moves across the field along a randomly selected direction. The sensors perform their sensing tasks under some sensing schedules. In a sensing schedule, a node periodically wakes up and goes to sleep to conserve energy while meeting the object detection quality requirement.

We state the assumptions of our model as follows.

- Sensors are randomly and independently distributed on the sensing field, with a density d.
- The sensor network is homogeneous, i.e., all sensors are identical. We denote the sensing range of a sensor as R.
- The size of an moving object can be neglected, considering that it is significantly smaller than the sensing range of an individual sensor.
- The object speed does not change during the detection process. We denote the object speed as v. In reality, a moving object may change its moving speed and direction during detection, thus, it may be difficult for the surveillance center to obtain the precise speed of the object as well. However, as shown in the later analysis of Section IV, the average-case object detection quality is a monotonous function of speed v when other parameters are fixed. Thus, given the range of object speed v based on past experiences, we can estimate the range of the corresponding object detection quality. Therefore, our model and analytical results are still useful for the cases of changing object speed.

In order to evaluate the average-case object detection quality of a sensor network, we define two metrics detailed as follows:

- *Detection Probability* (DP). The detection probability is defined as the probability that an object is detected in a certain observation duration.
- *Average Stealth Distance* (ASD). The average stealth distance is defined as the average distance an object travels before it is detected for the first time.

For worst-case object detection quality of the network, we have the following two metrics:

- *Sufficient Phase* (SP). The sufficient phase is defined as the smallest time duration in which an object is detected with 100% probability starting from any time for any position on the field where the object is initially located.
- *Worst-case Stealth Distance* (WSD). The worst-case stealth distance is defined as the longest possible distance that an object travels before it is detected for the first time.

Taking energy constraints into account, we further define other two metrics.

- *Lifetime* (LT). The system lifetime is the elapsed working time from system startup to the time when the object detection quality requirement cannot be met for the first time when live nodes continue sensing with their current periods.
- *Maximum Working Time* (MWT). The maximum working time is the longest possible working time of the system that satisfies the object detection quality requirement. Contrary to the definition of the lifetime, in which nodes have fixed sensing periods, in the definition of the maximum working time, when some nodes deplete their power, the remaining nodes can adjust their sensing periods to sustain the object detection quality.

Object detection applications, such as military surveillance [10] and habitat monitoring [17], may have different object detection quality requirements. For given sensing scheduling schemes, we assess their object detection quality using above metrics with respect to different system parameters. We study how each parameter affects the metrics, and how we can adjust them to reach the object detection quality goal while minimizing energy consumption.

IV. Average-Case Object Detection Quality Analysis of Random and Synchronized Sensing Schedules

parameter	definition
d	density of sensors
R	sensing radius of a sensor
v	object moving speed
P	sensing period of sensors
f	active ratio of sensors in P
H	active duration $H = f \cdot P$
t_a	observation duration

TABLE I

System modeling parameters in random and synchronized sensing schedules.

In the random sensing schedule, a node randomly and independently chooses the starting time of its active duration H in a sensing period P; while in the synchronized sensing schedule, all nodes start their active duration H at the same time in every sensing period P. In these two types of schedules, sensors have a sensing period P and an active ratio f. We denote the observation duration for object detection as t_a. Note that in the synchronized sensing schedule, P and f should be appropriately set, so that the time interval $P - H$ is significantly smaller than t_a. Under such a condition, even a very fast object has a large chance of being detected. These system parameters of a sensor network under the random and the synchronized sensing schedules are summarized in Table I. Because both random and synchronized sensing schedules are simple and representative schedules, they can be used to analyze other more complicated sensing schedules. For example, as shown in Section IV-F, we use both the random and the synchronized sensing schedules to analyze the previously-proposed heuristic sensing scheduling protocols, and achieve satisfactory results.

In this section, we first present the theoretical analyses on the average-case object detection quality of the random sensing schedules and the synchronized sensing schedules. We next design three practical sensing scheduling protocols, and study energy consumption and system working time of different schedules. Then, we show analytical results of these schedules and their simulation validations. Finally, we apply our model to two formerly proposed protocols in the literature, and analyze their average-case object detection quality.

A. Random Sensing Schedule Analysis

The random sensing schedule is a simple but usually efficient schedule due to its distributed nature. It can serve as a baseline for analysis and comparison to other schedules. In this subsection, under the random sensing schedule, we analyze the DP and ASD when sensors have the same sensing period P. Then, we study the DP for a special case of fast objects. We introduce this special case analysis because it yields more simplified numerical results and eases choosing appropriate network parameters. Finally, for fast objects, we study how nodes can sense with different periods to achieve the same DP as those having the same period.

1) Detection Probability: We study the random sensing schedules in which all the nodes have the same sensing period P and the same active duration H.

Consider a moving object moves from left to right on the x-axis. The object size is negligible, since it is significantly smaller than sensing ranges of the sensors on the field. Suppose it starts at the point $\frac{-vt_a}{2}$, travels a distance of vt_a, and arrives at the point $\frac{vt_a}{2}$ after the observation duration t_a. Define the *active area* AA of this object as the oblong area in Fig. 2, including the rectangle area with length vt_a and width of $2R$, and the two half disks with radius R attached to the rectangle. We can see that $AA = vt_a \cdot 2R + \pi R^2$.

Proposition 1: Let $Pr(x_s, y_s)$ denote the detection probability of a sensor located at (x_s, y_s) in the active area within t_a, and $\tilde{Pr}$ denote the probability that one single sensor can detect this object within t_a, then

$$\tilde{Pr} = \frac{1}{AA}\int_{-R}^{R} \mathrm{d}y_s \int_{-\frac{vt_a}{2}-R}^{\frac{vt_a}{2}+R} Pr(x_s, y_s)\mathrm{d}x_s. \quad (1)$$

Proof: For a specific sensor located at position (x_s, y_s) to detect this object, two conditions must be satisfied: (i) the sensor must be in the active area, (ii) the sensor must be active when the object crosses its sensing range. The detection probability of this sensor depends on the length of the segment that the object moving path intersects its sensing range. Since this probability closely relies on the intersecting length $l(x_s, y_s)$, which is defined as the length of the object trajectory that lies both in the sensing range of the sensor at (x_s, y_s) and the active area, we first look at how to compute it. As shown in Fig. 2, the intersecting length $l(x_s, y_s)$ can be described as

$$l(x_s, y_s) = min(\frac{vt_a}{2}, x_b) - max(\frac{-vt_a}{2}, x_a),$$

where $x_a = x_s - \sqrt{R^2 - {y_s}^2}$ and $x_b = x_s + \sqrt{R^2 - {y_s}^2}$ are the x coordinates of two intersecting points.

According to Fig. 3, the detection probability of this sensor is the probability that the intersecting interval $(l(x_s, y_s))$ intersects the sensing span, i.e., the object must pass the sensor's sensing range when the sensor is awake. By a little calculation, we get

$$Pr(x_s, y_s) = \begin{cases} f + \frac{t}{P} & \text{if } l(x_s, y_s) < (1-f)vP. \\ 1 & \text{if } l(x_s, y_s) \geq (1-f)vP. \end{cases}$$

where $t = \frac{l(x_s, y_s)}{v}$. Notice that $l(x_s, y_s) = 0$ and $Pr(x_s, y_s) = 0$ when (x_s, y_s) is outside the active area. Then, $\tilde{Pr}$ can be obtained by computing the expectation of $Pr(x_s, y_s)$ over the active area as in (1). □

For the case of multiple sensors, since the nodes are randomly deployed, the number of sensors in the active area follows a Poisson distribution with an expected value of $\lambda = d \cdot AA$ (for justification, please see [9] page 39).

Theorem 1: The detection probability under the random sensing schedule is

$$DP = 1 - e^{-\lambda \tilde{Pr}}. \quad (2)$$

Proof: We briefly describe the steps of the proof here. To derive the detection probability DP that at least one sensor can detect the moving object within duration t_a, given the probability $\tilde{Pr}$ that one sensor can detect this object, we can easily obtain the probability that one sensor cannot detect the object. Then, we obtain the probability that multiple sensors cannot detect the object, and derive the probability that no sensor can detect the object, which leads to the result of DP.

The probability that there are k sensors in the active area is $Pr(k) = \frac{e^{-\lambda} \cdot \lambda^k}{k!}, k = 0, 1, \ldots, \infty$, while the probability that a sensor cannot detect the object is $1 - \tilde{Pr}$. The probability that there exist k sensors in the active area and at least one of them can detect this object is $Pr(dt \wedge k) = \frac{e^{-\lambda}\lambda^k}{k!}(1 - (1 - \tilde{Pr})^k)$. Particularly, when $k = 0$, we have $Pr(dt \wedge k = 0) = Pr(k = 0) = \frac{e^{-\lambda} \cdot \lambda^0}{0!} = e^{-\lambda}$. Because $\sum_{k=0}^{\infty} \frac{e^{-\lambda} \cdot \lambda^k}{k!} = 1$, we have $\sum_{k=1}^{\infty} \frac{e^{-\lambda} \cdot \lambda^k}{k!} = 1 - e^{-\lambda}$. Also $Pr(dt \wedge k = 0) = Pr(k = 0) =$

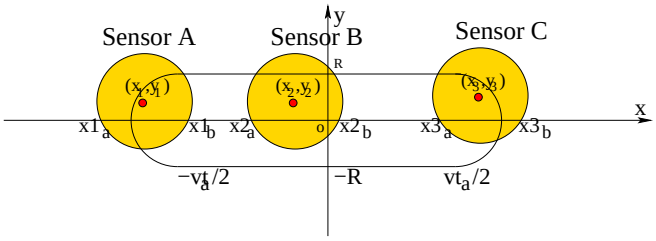


Fig. 2. Three sensors are located in the active area of a moving object.

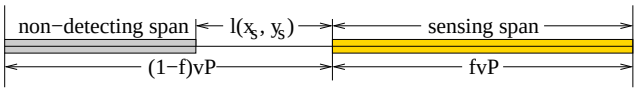


Fig. 3. The distance an object crosses in one sensing period.

$\frac{e^{-\lambda}\lambda^0(1-\tilde{P}r)^0}{0!} = e^{-\lambda}$, and $\sum_{k=0}^{\infty}\frac{e^{-\lambda}\lambda^k\cdot(1-\tilde{P}r)^k}{k!} = e^{-\lambda\tilde{P}r}$. The detection probability DP is the probability that at least one sensor detects the object, which is

$$\begin{aligned} DP &= \sum_{k=1}^{\infty} Pr(dt \wedge k) = \sum_{k=1}^{\infty}\frac{e^{-\lambda}\cdot\lambda^k}{k!}(1-(1-\tilde{P}r)^k) \\ &= \sum_{k=1}^{\infty}\frac{e^{-\lambda}\cdot\lambda^k}{k!} - \sum_{k=1}^{\infty}\frac{e^{-\lambda}\cdot\lambda^k}{k!}(1-\tilde{P}r)^k \\ &= (1-e^{-\lambda}) - (e^{-\lambda\tilde{P}r} - e^{-\lambda}) = 1 - e^{-\lambda\tilde{P}r}. \end{aligned}$$

□

2) Average Stealth Distance: The average stealth distance is an important metric to characterize the object detection quality. Here we derive the average stealth distance for the random sensing schedule.

Theorem 2: The average stealth distance under the random sensing scheme is

$$ASD = \int_0^{\infty} ve^{-\lambda\tilde{P}r}\mathrm{d}t_a. \tag{3}$$

Proof: First we summarize main ideas behind the proof. Because DP is a cumulative distribution function of the probability to detect the object with a variable t_a, from DP we can obtain the probability density function cdf. By definition, the average detection time is the expected value of the time it takes to detect the object, which can be computed as the integral of the product of t_a and the probability density function of t_a. Then, we obtain the average stealth distance by computing the product of object speed v and the average detection time.

Denote $cdf(x)$ and $pdf(x)$ as the cumulative distribution function and the probability density function of a numerical random variable x. We know $cdf'(x) = pdf(x)$. Also define $(1-cdf)(x) = 1 - cdf(x)$.

The DP in (2) basically denotes the probability that any sensor detects the object before time point t_a; it is a cdf function that can be written in the form $Pr(t \le t_a)$, where t is the time that the object is detected for the first time, and t_a can be viewed as a variable. We will next use DP to derive the expected detection time and then the expected stealth distance. Based on (2),

$$(1-cdf)(t_a) = Pr(t > t_a) = e^{-\lambda\tilde{P}r}.$$

Because $lim_{t_a\to\infty}cdf(t_a) = 1$, and $lim_{t_a\to\infty}(1-cdf)(t_a) = 0$, and they approach their limits exponentially when t_a approaches ∞ linearly, we do some integral transformation and get the expected detecting time

$$E(t_a) = \int_0^{\infty} pdf(t_a)\cdot t_a\mathrm{d}t_a = -\int_0^{\infty} t_a\mathrm{d}(1-cdf)(t_a).$$

Through integration by parts, we get

$$\begin{aligned} E(t_a) &= t_a\cdot(1-cdf)(t_a)|_0^{\infty} + \int_0^{\infty}(1-cdf)(t_a)\mathrm{d}t_a \\ &= \int_0^{\infty}(1-cdf)(t_a)\mathrm{d}t_a. \end{aligned}$$

Therefore, $E(t_a) = \int_0^{\infty} e^{-\lambda\tilde{P}r}\mathrm{d}t_a$. The stealth distance bears linear relationship to the expected detection time, we get

$$ASD = vE(t_a) = \int_0^{\infty} ve^{-\lambda\tilde{P}r}\mathrm{d}t_a.$$

□

3) Detection Probability For Fast Objects: For fast objects, we can obtain more simplified close-form results for the detection probability, as described in the following corollary. These close-form results can ease computation of nodes in practice when they need to set P and f to meet the quality requirement.

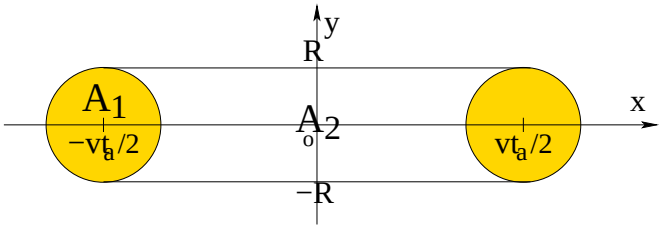


Fig. 4. The active area for detection probability calculation.

Corollary 1: We consider a special case, in which an object moves with a high speed v such that $vt_a > 2R$ and $(1-f)vP > 2R$. Then, the probability of a single sensor detecting this fast object is

$$\tilde{P}r = f + \frac{\pi R^2 t_a}{(vt_a\cdot 2R + \pi R^2)P}. \tag{4}$$

Proof: The main idea in the proof is to divide the active area of a single sensor into several parts, and derive the probability on each part, and sum up the results. On the other hand, we solve $l(x_s, y_s)$ by considering different conditions in which the detecting sensor s is located at different places.

Recall that $Pr(x_s, y_s)$ in the proof of Theorem 1, the probability that a sensor at location (x_s, y_s) detects the object is described in two forms according to the intersecting length.

For a fast object with a speed v such that $vt_a > 2R$ and $(1-f)vP > 2R$, $Pr(x_s, y_s)$ takes the form $f + t/P$ where $t = \frac{l(x_s,y_s)}{v}$. Then $\tilde{Pr}$ in (1) can be simplified as

$$\tilde{Pr} = f + \frac{1}{AvP}\int_{-\frac{vt_a}{2}-R}^{\frac{vt_a}{2}+R} \mathrm{d}x_s \int_{-R}^{R} l(x_s, y_s)\mathrm{d}y_s.$$

In the following we try to compute this probability by considering all the possible sensor locations, which is basically pure algebraic manipulation. Consider a sensor s located at (x_s, y_s). Denote $\xi_1 = \iint_{A_1} l(x_s, y_s)\mathrm{d}x_s\mathrm{d}y_s$, and $\xi_2 = \iint_{A_2} l(x_s, y_s)\mathrm{d}x_s\mathrm{d}y_s$, where A_1 is the circle on the left and A_2 is the unfilled area in the middle as shown in Fig. 4. Due to the symmetry of the integrating area, we have

$$\int_{-\frac{vt_a}{2}-R}^{\frac{vt_a}{2}+R} \mathrm{d}x_s \int_{-R}^{R} l(x_s, y_s)\mathrm{d}y_s = 2\xi_1 + \xi_2.$$

Let x_a and x_b ($x_b > x_a$) be the x coordinates of the two intersecting points between the object path and the sensing circle of node s. Notice that $l(x_s, y_s) = max(x_b, -\frac{vt_a}{2}) - max(x_a, -\frac{vt_a}{2})$ when $(x_s, y_s) \in A_1$. Now we compute $l(x_s, y_s)$ under following conditions:

- $x_b > x_a > -\frac{vt_a}{2}$. We have $x_s > \sqrt{R^2 - {y_s}^2} - \frac{vt_a}{2}$ and $l(x_s, y_s) = x_b - x_a = 2\sqrt{R^2 - {y_s}^2}$.
- $x_b > -\frac{vt_a}{2}$ and $x_a < -\frac{vt_a}{2}$. We have $-\frac{vt_a}{2} - \sqrt{R^2 - {y_s}^2} < x_s < -\frac{vt_a}{2} + \sqrt{R^2 - {y_s}^2}$ and $l(x_s, y_s) = x_b + \frac{vt_a}{2} = x_s + \frac{vt_a}{2} + \sqrt{R^2 - {y_s}^2}$.
- $x_b < -\frac{vt_a}{2}$ and $x_a < -\frac{vt_a}{2}$. We have $l(x_s, y_s) = 0$.
- $x_b < -\frac{vt_a}{2}$ and $x_a > -\frac{vt_a}{2}$. Because $x_b > x_a$, this case is impossible.

We can get

$$\begin{aligned}\xi_1 &= \int_{-R}^{R} \mathrm{d}y_s \int_{-\sqrt{R^2-{y_s}^2}-\frac{vt_a}{2}}^{\sqrt{R^2-{y_s}^2}-\frac{vt_a}{2}} (x_s + \frac{vt_a}{2} + \sqrt{R^2 - {y_s}^2})\mathrm{d}x_s \\ &= \frac{8R^3}{3},\end{aligned}$$

and

$$\begin{aligned}\xi_2 &= \int_{-R}^{R} \mathrm{d}y_s \int_{\sqrt{R^2-{y_s}^2}-\frac{vt_a}{2}}^{-\sqrt{R^2-{y_s}^2}+\frac{vt_a}{2}} 2\sqrt{R^2 - {y_s}^2}\mathrm{d}x_s \\ &= \pi R^2 vt_a - \frac{16R^3}{3}.\end{aligned}$$

Therefore,

$$2\xi_1 + \xi_2 = \pi R^2 vt_a.$$

We can get

$$\tilde{Pr} = f + \frac{1}{AvP}(\pi R^2 vt_a),$$

which leads to (4). □

4) Sequential Schedule and k-Set Schedule: As an extension of our previous results, here we show two equivalent scheduling schemes that can achieve the same detection quality as the random schedule with a constant sensing period P. We assume $2R < (1-f)vP$, which implies that $l(x_s, y_s)$ is always less than $(1-f)vP$, and H is constant. Under these assumptions, according to (2), we know $\tilde{Pr}$ can be written in the form of $\frac{a}{P}$, where a is a variable that is independent of P and λ. In the following analysis, we only vary P and λ while leaving all other system parameters unchanged.

Lemma 1: Let A be a schedule with sensing period kP, where k is a non-negative value. Let the expected node density be λ. We randomly divide the nodes into k equal-sized sets, and nodes in each set are randomly distributed in the field. Consider a sequential schedule B, where nodes in ith set are active only in the duration of $[(i-1)P + nkP, iP + nkP)$ for $1 \le i \le k$, then the schedule A and the schedule B have identical detection probability, i.e., $DP_A = DP_B$.

Proof: In schedule B, all sets have identical detection probabilities. Consider the ith set S_i, the detection probability is

$$DP_B(S_i) = 1 - e^{-\frac{\lambda}{k}\tilde{Pr}} = 1 - e^{\frac{\lambda a}{kP}},$$

which is the same as that of schedule A. □

Lemma 2: We randomly divide the nodes into k sets $S_1, S_2, \cdots, S_k$. For any set S_i with density $x_i\lambda$, we associate a sensing period $g_i P$ with it. Let $DP(S_i)$ denote the DP for the nodes in set S_i. If $\frac{x_1}{g_1} + \frac{x_2}{g_2} + \cdots + \frac{x_k}{g_k} = 1$, the detection probability DP of this k-set schedule is equal to that of the schedule with all nodes having the same period P.

Proof: We know that $DP(S_i) = 1 - e^{-x_i\lambda\cdot\frac{a}{g_i P}}$. Let $\overline{DP(S_i)}$ be the probability that no node in S_i detects this object, so

$$\overline{DP(S_i)} = 1 - DP(S_i) = e^{-x_i\lambda\cdot\frac{a}{g_i P}}.$$

Thus, we have

$$\begin{aligned}DP &= 1 - \overline{DP(S_1)}\cdot\overline{DP(S_2)}\cdots\overline{DP(S_k)} \\ &= 1 - e^{\frac{-\lambda a}{P}(\frac{x_1}{g_1} + \frac{x_2}{g_2} + \cdots + \frac{x_k}{g_k})} = 1 - e^{\frac{-\lambda a}{P}}.\end{aligned}$$

□

B. Synchronized Sensing Schedule Analysis

The synchronized sensing schedule represents a typical class of sensing scheduling protocols, in which sensors synchronize their sensing duties. Compared to the random sensing schedule, its average-case object detection quality, measured by detection probability and average stealth distance, might be worse. However, the synchronized sensing schedule has the benefit that the object non-detecting traveling distance, i.e., the distance an object travels before detection, is bounded. If the field is fully covered by all active sensors, the object non-detecting traveling distance is bounded by the maximum distance this object travels in one sensing period. As mentioned earlier, both the random and the synchronized schedules can be used to analyze other more complex heuristic sensing scheduling protocols.

Under the synchronized sensing schedule, we first analyze the DP under the given system parameters. Based on the DP analysis, we then derive the ASD. Note that all nodes have the same sensing periods here.

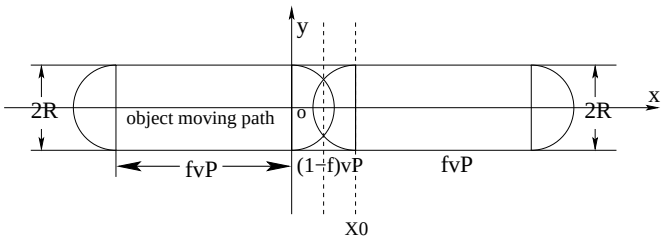


Fig. 5. The active area in the synchronized schedule when R is large.

1) Detection Probability: Similar to the random sensing analysis, we study the active area under a synchronized sensing schedule to derive the detection probability.

Consider the traveling distance of a moving object in one sensing period P, we divide it into two parts: the first part is the distance the object travels in the duration $(1-f)P$ when all sensors are asleep; the second part is the distance the object travels when all sensors are active. In the first part, the object cannot be detected by any sensor; however, in the second part, the object can be detected when there are active sensors within a distance of R to it. Define the *active area* AA of a moving object as the set of points that are within a distance of R to the second part traveling segments of this object.

As shown in Fig. 5, *AA* is the set of periodically repeated areas, except the last one when t_a is not a multiple of P. Each repeated area is either a rectangle plus two overlapped half circles for a large R (shown in Fig. 5), or a rectangle plus two disjoint half circles for a small R. Denote $X_0 = (1-f)vP$ as shown in Fig. 5. We assume that $t_a > P$.

Let $IA(P)$ be the total covering area of two half disks in one intermediate sensing period P. We consider whether there is overlapping in $IA(P)$. When $R \geq \frac{X_0}{2}$, intersecting points of two half disks are $(\frac{X_0}{2}, -\sqrt{R^2 - \frac{{X_0}^2}{4}})$ and $(\frac{X_0}{2}, \sqrt{R^2 - \frac{{X_0}^2}{4}})$. Then $IA(P) = 4\int_0^{\frac{X_0}{2}} \sqrt{R^2 - x^2}\mathrm{d}x = X_0\sqrt{R^2 - \frac{{X_0}^2}{4}} + 2R^2 \arcsin\frac{X_0}{2R}$. When $R < \frac{X_0}{2}$, $IA(P) = \pi R^2$. To summarize the results, we have

$$IA(P) = \begin{cases} X_0\sqrt{R^2 - \frac{{X_0}^2}{4}} + 2R^2 \arcsin\frac{X_0}{2R} & \text{if } R \geq \frac{X_0}{2}. \\ \pi R^2 & \text{if } R < \frac{X_0}{2}. \end{cases}$$

Therefore, the active area in one intermediate sensing period P is $AA(P) = IA(P) + 2RvfP$.

To calculate the detection probability, we have the following theorem.

Theorem 3: During the observation duration t_a, the active area is $AA(t_a) = \pi R^2 - IA(P) + \frac{t_a IA(P)}{P} + 2Rvft_a$. Let λ_s be the expected number of sensors in the active area, $\lambda_s = d \cdot AA(t_a)$. Then

$$DP = 1 - e^{-\lambda_s} \tag{5}$$

Proof: The probability that no sensor in the active area is $e^{-\lambda_s}$. So, the detection probability that at least one sensor can detect this object under the synchronized sensing schedule is $DP = 1 - e^{-\lambda_s}$. □

2) Average Stealth Distance: Based on the above DP result, we can immediately derive the average stealth distance for the synchronized sensing schedule. We have the following theorem.

Theorem 4: The average stealth distance ASD under the synchronized sensing schedule is

$$ASD = \frac{vP}{d \cdot (IA(P) + 2RvfP)} e^{-d(\pi R^2 - IA(P))}. \tag{6}$$

Proof: Here we briefly summarize the proof. We obtain the detection probability density function based the result of DP for synchronized sensing schedules, and derive the average detection time. The average stealth distance is the product of the object speed v and the average detection time. The derivation here basically follows the same vein as in the random sensing schedule case.

DP in (5) is actually a *cdf* function that can be written in the form $Pr(t \leq t_a)$, where t is the time that the object is detected for the first time, and t_a can be seen as a variable. Then we know

$$(1-cdf)(t_a) = 1 - DP = e^{-d(\pi R^2 + IA(P))} \cdot e^{-d(\frac{IA(P)}{P} + 2Rvf)t_a}.$$

Let $F'(t_a) = (1 - cdf)(t_a)$, then

$$F(t_a) = \frac{-Pe^{-d(\pi R^2 - IA(P))}}{d(IA(P) + 2RvfP)} e^{-d(\frac{IA(P)}{P} + 2Rvf)t_a} + C,$$

where C is constant.

Let $E(t_a)$ be expected detecting time, then we have

$$E(t_a) = \int_0^\infty (1 - cdf)(t_a)\mathrm{d}t_a,$$

then

$$E(t_a) = F(t_a)|_0^\infty = \frac{Pe^{-d(\pi R^2 - IA(P))}}{d(IA(P) + 2RvfP)}.$$

So

$$ASD = vE(t_a) = \frac{vP}{d \cdot (IA(P) + 2RvfP)} \cdot e^{-d(\pi R^2 - IA(P))}.$$

□

Now we study a special case of $f = 100\%$, which means nodes are awake the whole time and never sleep. We have $(1 - cdf)(t_a) = e^{-d\pi R^2 - 2dRvt_a}$, therefore

$$ASD = \int_0^\infty ve^{-(d\pi R^2 + 2dRvt_a)}\mathrm{d}t_a = \frac{e^{-d\pi R^2}}{2dR}. \tag{7}$$

C. Practical Power Efficient Sensing Protocols

Applying our analytical framework to practice, in this section, we propose three practical sensing protocols that ensure: (1) the object detection quality requirement is satisfied, (2) low sensing duty cycles are utilized to save sensing energy, and (3) only moderate communication and computation overhead are incurred. In these protocols, H is fixed, and there are n sensors in the network. Particularly, the superior part of the third protocol PAAS is that all sensors with different energy deplete their energy simultaneously to extend the system lifetime, which is highly useful in practice. The protocols are detailed as follows.

(1) *Global Random Schedule (GRS)*: In this protocol, the global node density d is known to all sensors. According to

Theorem 1, each node calculates the maximum sensing period P_{max} that satisfies the DP requirement, and senses the field with P_{max}.

(2) *Localized Asynchronous Schedule (LAS)*: This protocol is based on the fact that sensors in a dense region can have a larger P than those in a scarce region to reach the same object detection quality. After a node boots up, it broadcasts beaconing messages and infers the relative distance to its neighbors based on their signal strength. Then, it computes its local node density d_l by dividing the number of nodes in its communication range over the area of that range. According to Theorem 1, each node uses its local density d_l to compute the maximum period P_{max} that meets the object detection quality requirement as its sensing period. Thus, this algorithm achieves an object detection quality close to the targeted one.

(3) *Power-Aware Asynchronous Schedule (PAAS)*: This protocol takes the diversity of power capacity among sensor nodes into consideration. The whole set of nodes are divided into k sets $S_1, S_2, \cdots, S_k$, such that all nodes in set S_i have approximately the same power capacity E_i, where $1 \leq i \leq n$. Based on Lemma 2, we can set $g_i = \frac{\sum_{i=1}^{k} x_i E_i}{E_i}$ to achieve the same object detection quality as GRS does with a constant sensing period P for each node. If each set has one and only one node, given the sum of the power capacities $E = \sum_{i=1}^{n} E_i$, we can schedule a node that has a power capacity E_i with a sensing period $\frac{E}{nE_i}P$ to achieve the same object detection quality as the GRS protocol has.

D. Energy Consumption and Working Time Analysis

System lifetime is a critical factor that indicates the quality of sensor networks, since energy is extremely scarce resource in each node. In this subsection, we study the system lifetime under different sensing schedules. We make two assumptions in the analysis of system lifetime. First, we neglect the sensor wake-sleep transition energy cost in these schedules. As shown above, in our model, it is possible to vary either P or f to achieve the required object detection quality. Although there is wake-sleep transition energy cost when a sensor wakes up in practice, by choosing relatively large sensing period P, this wake-sleep transition energy cost is much smaller than the energy saved by the sensor nodes being inactive in the period. In our paper, we assume that in all schedules, the active duration H is long enough to ignore the wake-sleep transition energy cost. The second assumption is that, sensor nodes can monitor their remaining energy, and transmit their energy data to other nodes in the network, and thus, sensors are energy-aware. Monitoring remaining energy can be done easily in hardware, and energy information can be collected and disseminated through periodic communication.

Let T be the continuous working time of a single node, and all nodes have the same T. Under the random sensing schedule and the synchronized sensing schedule, if all nodes have the same P and f, one node spends H energy in a period P. This node will last for $\frac{T}{H}$ periods, thus, its working time is $\frac{T}{H} \cdot P = \frac{T}{f}$. Therefore, the system lifetime is $LT = \frac{T}{f}$. Particularly, when H is constant, $LT = \frac{T}{f} = \frac{TP}{H}$, which means that a small f or a large P can yield a long system lifetime.

Define the first failure time and the last failure time as the time when the first live node and the last live node in the system deplete their power. For a sensor network with n nodes, we denote T_i as the time when the ith node runs out of its power for $i = 1, 2, \ldots, n$, and define T_f and T_l as the first failure time and the last failure time of the network. Note that H is fixed here.

In GRS, all nodes have the same sensing period P and the same active ratio f. Therefore, $T_i = \frac{E_i}{f}$ for $i = 1, 2, \ldots, n$. So, $T_f(GRS) = min(T_1, T_2, \ldots, T_n) = min(\frac{E_1}{f}, \frac{E_2}{f}, \ldots, \frac{E_n}{f})$. In PAAS, because nodes have different sensing period, they have different active ratio as well. Let P and f be the fixed sensing period and the fixed active ratio in the GRS protocol, respectively. Denote f_i as the active ratio of the ith node, where $i = 1, 2, \ldots, n$, then we have $f_i = \frac{H}{g_i P}$. On the other hand, because $g_i P = \frac{E}{nE_i}P$, we can get $f_i = \frac{nfE_i}{E}$. Note that in PAAS, all nodes have the same elapsed working time, i.e., $T_f = T_l = T_1 = T_2 = \ldots = T_n$. Therefore, $T_f(PAAS) = \frac{E}{nf}$. Because $\frac{E}{n} \geq min(E_1, E_2, \ldots, E_n)$, we know $T_f(GRS) \leq T_f(PAAS)$. In other words, PAAS has a larger first failure time than GRS.

The maximum working time is always longer than the lifetime in the previous definition, thus, it can better characterize the energy consumption property of the network. Here we consider a *simple random sensing schedule*, in which all nodes have identical sensing periods at any moment, and only wake up once in one sensing period. We have the following theorem.

Theorem 5: With the same DP requirement, the simple random sensing schedule and the PAAS have the same energy consumption rate, thus, have the same maximum working time.

Proof: We know $DP = 1 - e^{-\lambda c/P}$, where c is constant if H and other detection parameters are fixed. The energy consumption rate that meets the required detection quality is fixed and is proportional to λ/P. This is because the number of participating sensors is proportional to λ, and the energy consumption of each sensor is proportional to $1/P$. Therefore, for any simple random sensing schedule with a given detection probability requirement, the energy consumption rate is nf, where n is the total number of nodes and f is the active ratio of each sensor node.

For the PAAS, even though each node sets its P according to its remaining power, the total power consumption of all nodes is still constant. Consider the ith node in all n nodes, where $1 \leq i \leq n$. Its energy consumption rate is $er_i = \frac{H}{P_i}$. Because $P_i = \frac{E}{nE_i}$ and H is constant, then $er_i = \frac{H}{P_i} = \frac{H}{\frac{E}{nE_i}P} = \frac{nHE_i}{EP}$. The total energy consumption rate is $\sum_{i=1}^{n} er_i = \sum_{i=1}^{n} \frac{nHE_i}{EP} = \frac{nfEP}{EP} = nf$. Therefore, the PAAS has the same maximum working time as the simple random schedule, in which all nodes have fixed sensing periods. □

E. Analysis Validation and Protocol Evaluation

In our simulation experiments, we generate a 500×500 grid field, and randomly place $d \times 250,000$ sensors on it. Sensors

use either the random sensing schedule or the synchronized sensing schedule. A small object moves along a straight line towards a randomly selected direction with a constant speed v. We run each simulation scenario for hundreds of times. Then, we use the ratio of detection times over the number of experiments to estimate DP, and use the average non-detecting distance to estimate ASD.

metric	$d\uparrow$	$R\uparrow$	$v\uparrow$	$t_a\uparrow$	$P\uparrow$	$f\uparrow$
DP	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\downarrow$
ASD	$\downarrow$	$\downarrow$	$\uparrow$		$\downarrow$	$\uparrow$

TABLE II

DP AND ASD CHANGE WHEN SYSTEM PARAMETERS INCREASE.

1) Evaluation of Random and Synchronized Schedules: We plot both analytical curves and simulation results under different combinations of six parameters as shown in Fig. 6, 7, and 8, respectively. Our observations are summarized as follows.

- The simulation results match the analytical curves well, which validates the correctness of our derivations.
- DP and ASD monotonically increase or decrease with the increase of the parameters, as shown in Table II.
- The random schedule outperforms the synchronized schedule on both DP and ASD, which is shown in Fig. 8. This is because the synchronized schedule causes more overlapping sensing areas than the random schedule.
- The non-detecting distance distributions have long tails: most non-detecting distances are short, while a few have large values. The worst case of non-detecting distance in the random schedule is longer than that of the synchronized schedule.

2) Evaluation of GRS, LAS, and PAAS Protocols: We use the DP to evaluate the effectiveness of the GRS, LAS, and PAAS protocols, and use the first failure time, the last failure time, and the system lifetime to compare their power consumption properties.

In our experiments to evaluate these three protocols, each sensor node's energy follows a uniform distribution between $[0, E_{max}]$. We set system parameters as follows: $d = 0.2$, $R = 0.5$, $v = 5$, $t_a = 2$, $P = 1.1$, $H = 0.55$, $r = 3$, and $E_{max} = 30$. Here r is the range to compute the local density in LAS. Given the requirement of $DP \geq 60\%$, Fig. 9 illustrates the degradation of DP as nodes run out of power. Note that every data point in this figure is obtained by averaging hundreds of experiment results.

Based on the simulation results, we have the following observations.

- GRS, LAS, and PAAS can achieve the same DP at the beginning when no sensor depletes its energy.
- The first failure time and the last failure time of PAAS are the same; by contrast, GRS and LAS have smaller first failure time and larger last failure time.
- PAAS has a longer system lifetime than those of GRS and LAS.
- The DPs of GRS and LAS in Fig. 9 degrade exponentially, instead of linearly. This is because for a sensor whose energy is uniformly distributed in $[0, E_{max}]$, the DP at time t is $DP(t) = 1 - e^{-\lambda(t)\tilde{P}r}$, where $\lambda(t) = \lambda_0 - qt$, q is the death rate, and λ_0 is the initial sensor density. Thus, $DP(t) = 1 - e^{\lambda_0 \tilde{P}r} \cdot e^{qt\tilde{P}r}$.

F. Applying the Model to the PECAS Protocol and the Mesh Protocol

Our analytical framework provides guidelines for choosing appropriate parameters in these protocols to achieve the required object detection quality, facilitates their performance evaluation more rigorously, and gives insights on the inherent performance tradeoff in them and many other general sensing scheduling protocols under partial coverage. In this subsection, we further apply our analytical framework to two sensing schedules in the literature, namely the Probing Environment Coordinated Adaptive Sleeping (PECAS) protocol and the Mesh protocol. In particular, we present the analytical $QoSv$ results on the heuristic PECAS protocol, which is evaluated only by simulation in [7]. In comparison, in the random and synchronized schedules we previously proposed, each node periodically wakes up and goes to sleep in one sensing period. The active ratio is initially set, and no communication between neighboring nodes is needed afterwards. By contrast, in the PECAS and Mesh protocols, nodes need to communicate with others and conduct computations frequently. On this aspect, the random and synchronized sensing schedules are simpler than the PECAS and Mesh protocols. Our analysis shows that many heuristic sensing scheduling protocols, such as PECAS, are variations of the random schedule or the synchronized schedule, and can be incorporated into our analytical framework by setting parameters appropriately.

1) Analysis of the PECAS Protocol: The PECAS protocol [7] is an enhanced variance of the Probing Environment and Adaptive Sleeping (PEAS) protocol [30]. In PECAS, every node remains in the working mode only for a duration indicated by parameter $Work\_Time\_Dur$ instead of being active all the time as in the PEAS protocol [30]. When a node starts working, it sets the $Next\_Sleep\_Time$ as the current time plus $Work\_Time\_Dur$ to indicate the time-stamp this working node will stop working and go to sleep. When a working node responds to a PROBE message, the value of $Next\_Sleep\_Time$ timer is piggybacked to the REPLY message. Since the node keeps a record of the earliest $Next\_Sleep\_Time$ value among the collected REPLY messages, the next sleep duration is set as the earliest $Next\_Sleep\_Time$ value minus the current time. In this way, it is assured that when a working node begins sleeping, other sleeping nodes in the neighborhood will wake up and probe.

Here we extract the network parameters out of the PECAS experiments in [7]. Let the node density of the field be d, and the probing range of a node be r. In a circle area of πr^2, the expected number of nodes is $d \cdot \pi r^2$. Because $d \cdot f = \frac{1}{\pi r^2}$, on average the active ratio of a node is $f = \frac{1}{d\pi r^2}$. The system parameters in [7] are as follows: $d = \frac{800}{400m \times 400m} = 0.005/m^2$, $R = 20m$, $v = 10m/s$, and r varies from $20m$ to $56m$. Since $f = \frac{1}{d\pi r^2}$, we know f changes from 0.159 (when $r = 20m$) to 0.0203 (when $r = 56m$). The working

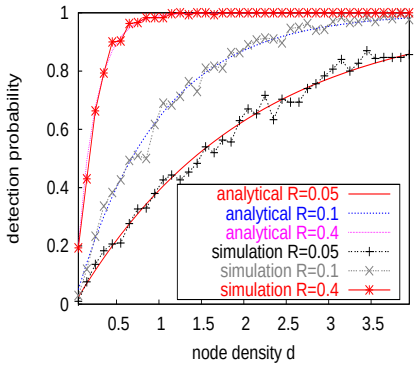


Fig. 6. DP under the random schedule. $v = 1, t_a = 5, P = 0.1, f = 0.5$.

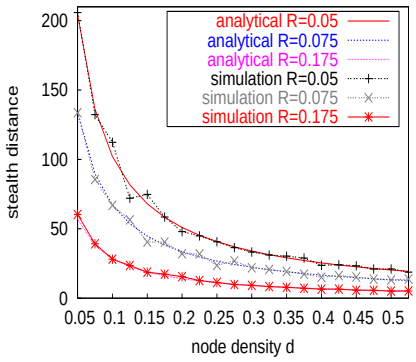


Fig. 7. ASD under the random schedule, $v = 1, P = 0.1, f = 0.5$.

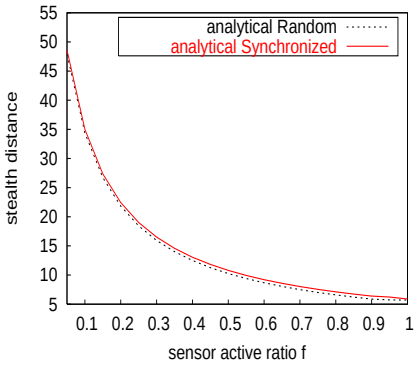


Fig. 8. ASD under the two schedules. $d = 0.4, R = 0.2, v = 2, P = 2$.

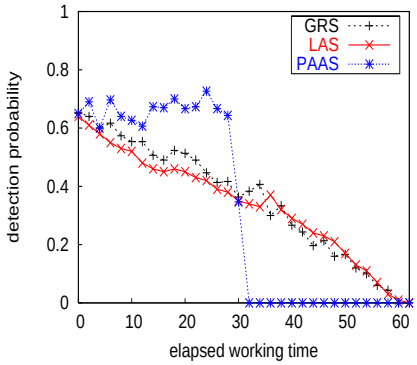


Fig. 9. DP comparison between GRS, LAS, and PAAS.

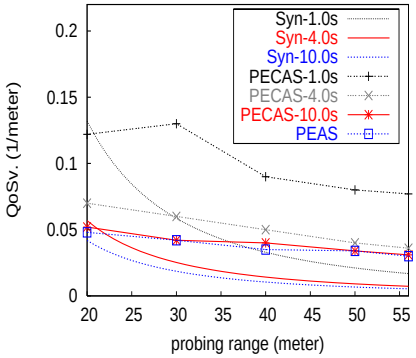


Fig. 10. $QoSv$ under the synchronized schedule compared to that in [7].

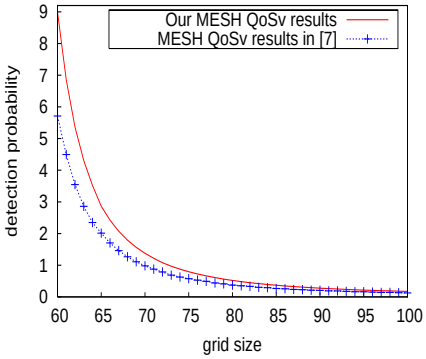


Fig. 11. $QoSv$ of the Mesh protocol compared to that in [7].

time duration in the three curves in [7] is 1.0sec, 4.0sec, and 10.0sec, respectively. This duration is H in the random schedule and the synchronized schedule. On the other hand, the $QoSv$ is the reciprocal of the ASD, i.e., $QoSv = 1/ASD$.

With these parameter settings, we plot the corresponding $QoSv$ under the random and the synchronized sensing schedules as well as the PECAS curves in [7]. A larger probing range r or a larger working time duration results in a smaller $QoSv$. We find that the random sensing schedule has a better $QoSv$ than PECAS for the reason that a small node density d incurs a small chance of nodes being close to each other. On the other hand, the synchronized sensing schedule has close $QoSv$ results to those of the PECAS protocol, as shown in Fig. 10. For example, when the working time duration is 1 second, the $QoSv$ difference between the synchronized sensing schedule and the PECAS schedule is less than 0.1/meter, which is relatively small. This is because, in PECAS, once a node goes into sleep, several nodes around it wake up, which is similar to the scenario where nodes all wake up simultaneously in the synchronized schedule.

2) Analysis of the Mesh Protocol: In the Mesh protocol [7], a planned distribution method is used to achieve the soft deployment where the object detection quality can be satisfied with deterministic guarantee. Every sensor node is assumed to know its geometric location. Only nodes at planned locations remain active so that all active nodes forms a planned pattern of 2-D mesh where the active nodes on the field form a set of horizontal and vertical solid lines. The distance between

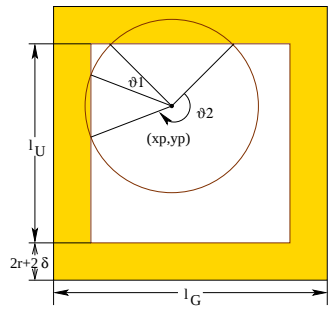


Fig. 12. Uncovered square of one grid in the Mesh protocol.

each adjacent horizontal or vertical lines is L_G. For the i-th horizontal line, sensors whose x-coordinates in the range $[i \cdot L_G - \delta, i \cdot L_G + \delta]$ remain active. Same is true for nodes around the j-th vertical line. Let the sensing range of each sensor be r, then each this horizontal or vertical line forms a stripe of a covered area of width $2r + 2\delta$. Each uncovered area in this sensor deployment is a square with side length of l_u, $l_u = l_G - 2r - 2\delta$, as shown in Fig. 12.

If the node density is high, for a randomly chosen point, its probability of not being covered by any active sensor is $Pr_{uc} = \frac{(\lfloor \frac{L}{L_G} \rfloor)^2 (L_G - 2r - 2\delta)^2}{L^2}$. As shown in Fig. 12, for the point with coordinate (x_p, y_p) in the uncovered square, we draw a disk centered at it with a radius of vt_a. Denote $\xi = \frac{L_u}{2}$. Suppose there are $2m$ intersecting points between this disk and the four border lines, then the circle is divided into arcs

by the intersecting points, interleavingly inside and outside the uncovered square. Let the angles of these arcs inside be $\theta_1(x_p, y_p), \ldots, \theta_m(x_p, y_p)$. By definition, the average DP at the point (x_p, y_p) is $\frac{\sum_{i=1}^{m} \theta_i}{2\pi}$. We integrate the average DP of a point over the whole uncovered square to obtain the DP in t_a:

$$DP_{mesh} = \frac{(\lfloor \frac{L}{L_G} \rfloor)^2 \int_{-\xi}^{\xi} \mathrm{d}x_p \int_{-\xi}^{\xi} (\frac{\sum_{i=1}^{m} \theta_i(x_p, y_p)}{2\pi}) \mathrm{d}y_p}{L^2}.$$

We use the same parameter settings as that in [7], which are listed as follows: $l = 400m$, $R = 20m$, $v = 10m/s$, l_G varies from $60m$ to $100m$, $l_U = l_G - 10m$, and $2\delta = 10m$. The DP here is a cdf function of the variable t_a. We integrate the $(1 - cdf)$ function over the time span of $[0, \infty)$ to obtain the ASD, then $QoSv_{mesh} = \frac{1}{\int_0^{\infty} DP_{mesh} \mathrm{d}t_a}$. The comparison between our results and the results in [7] is illustrated in Fig. 11. The close match of the two curves validates the correctness of both analyses.

V. Worst-Case Object Detection Quality Analysis of Wave Sensing Protocols

In practice, many applications demand stringent requirements on worst-case object detection quality. For example, an object must be detected in 10 seconds with certainty. However, worst-case object detection quality metrics, such as sufficient phase and worst-case stealth distance, are not bounded in the random and synchronized schedules shown in Section IV. Given an observation duration, an object can escape detection; it can also travel an infinite distance even though its average stealth distance is small. Therefore, we need to design sensing scheduling protocols that achieve a bounded sufficient phase and worst-case stealth distance while minimizing energy consumption of the system. In this section, we present and analyze three wave sensing schedules including the line wave, the stripe wave, and the distributed wave sensing scheduling protocols, and evaluate the performance of the wave sensing protocols via extensive simulations.

In the design and analysis of the wave sensing scheduling protocols, we assume that the sensing field is completely covered when all sensors on the field are active. The main idea behind these protocols is as follows. When the distance between any two nodes is less than their sensing diameter $2R$, their sensing ranges intersect and form a connected region. If currently-active nodes make up a connected stripe with two ends on opposite borders of the field, the stripe divides the field into two regions. Under such a circumstance, in a sufficiently long time duration, an object can be detected when it crosses this stripe. For any specified continuous curve with two ends on opposite borders of the field, it is always possible to find a set of nodes whose sensing ranges completely cover this curve under the assumption that the field is completely covered when all the nodes wake up. To further reduce the detection time, we allow the curve to move so that every geographical point on the field can be scanned at least once in one wave scanning period, without leaving any sensing hole in this scanning period. Here a *sensing hole* is a continuous area that is not covered by any sensors in one wave scanning period. We define the curve (line) to be covered as the *active curve (line)*, and define the union of the sensing ranges of all active sensors that cover the active curve (line) as the *hot region*.

Our design goals are: (1) the hot region should contain no sensing hole in one wave scanning period, (2) the hot region should be as thin as possible in order to reduce network energy consumption; and (3) the active curve should move repeatedly, so that the object can be detected rapidly and energy consumption variance among nodes is small.

A. Line Wave Protocol Design

In this protocol, we make two assumptions. First, we assume that every node on the field has a timer that is well synchronized with others. The global timer synchronization techniques of [14] can be used in this protocol. Second, we assume that every node is aware of its own geographical location on the field through some localization techniques. Many previously proposed localization algorithms (e.g., [19], [26]) are practical, effective, and extremely cheap, and have been deployed in multiple real sensor network projects.

1) Line Wave Protocol Description: In the line wave protocol, the active curve is a straight line, as shown in Fig. 13. This protocol is specified as follows.

1. At system startup time, all nodes synchronize their timers, and obtain their geographical coordinates. There are two active lines on the two opposite borders of the field moving towards the center. All nodes are informed of the initial positions, the settling time, and the advancing distance (ad) of the active lines. Note that $ad < 2R$.
2. Every node computes current positions of the active lines based on its timer and the information of the active lines it obtained. Then, it calculates if its sensing range intersects the active lines. If there is an intersection, this node wakes up.
3. After the active lines have stayed at their current positions for their settling time, they move forward with a distance ad towards the field center. When they reach the center, they go back to the field borders. Step 2 repeats.

Note that a sleeping node periodically wakes up to receive new messages addressed to it. Sensing tasks are distributed to all nodes, thus, energy consumption variance among nodes is kept small.

2) Bounded Sufficient Phase and Worst-Case x-Axis Stealth Distance: We define the *x-axis stealth distance* as the distance a moving object travels on the x-axis before it is detected. A *handoff* is defined as the process when active lines advance to their new positions, all nodes covering the new lines wake up, and those nodes covering old lines only go to sleep.

Theorem 6: In the line wave protocol, the sufficient phase of any moving object is bounded by $2P$, where P is the wave scanning period. In other words, the moving object can always be detected in $2P$.

Proof: Consider the handoff process of an active line in the line wave protocol. Suppose this line moves from left to right. We denote the old active line as ol, and denote the new active line as nl. Suppose there is a sensing hole H in the sensing range union. Consider a point $p \in H$. When all the nodes on

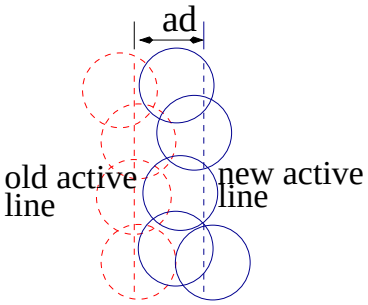


Fig. 13. Line wave protocol illustration.

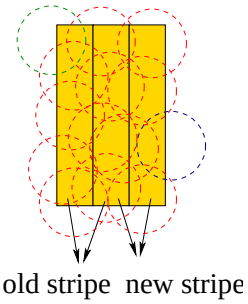


Fig. 14. The sensing stripe handoff in the stripe wave protocol.

the field wake up, the field can be completely covered. Thus, there must exist a sensor s that can cover p when it wakes up. Denote the circle of s's sensing range as C. We know that C either intersects ol or nl, or both, because the diameter of C is $2R$, while the distance between ol and nl is less than $2R$. Therefore, there is no sensing hole in the hot regions of ol and nl.

We next prove an object can be detected in $2P$. Starting from any time point, it takes at most P time for the active lines to return to the boundaries of the field. After another P, the hot regions scan the sensing field without leaving any sensing hole, thus, the object is detected. □

Lemma 3: The worst-case x-axis stealth distance is less than $2vP$. If the object moves along a straight line with an even speed v, the worst-case x-axis stealth distance is less than L.

Proof: According to Theorem 6, the object is detected in $2P$ time. The distance that the object travels in $2P$ is $2vP$.

Suppose that the object travels along a straight line, and it takes $2P$ to detect this object. In the first P, when the object is behind one of the active lines and is chasing that line, it can travel at most $\frac{L}{2}$ on the x-axis without being detected. In the second P, the object is between the two active lines, the maximum x-axis distance it can travel is $\frac{L}{2}$. Therefore, the object can travel at most L on the x-axis before being detected. □

B. Stripe Wave Protocol Design

One restriction of the line wave protocol is the precision requirement on node coordinates. To relax this constraint, we design a stripe wave protocol. In this protocol, stripes, instead of lines, are covered by active sensors, as shown in Fig. 14. When the stripe width is larger than the required coordinate precision, object detection quality can be achieved.

In the stripe wave protocol, nodes wake up if their sensing ranges intersect active stripes. The width of active stripes is twice of their advancing distance. In this way, there is an overlap between the old stripe and the new stripe. All the other procedures remain the same as those of the line wave protocol.

1) Sufficient Phase and Worst-Case x-Axis Stealth Distance of Stripe Wave Protocols: If one active stripe stays in a place for the same amount of time, and advances the same distance in the same direction as the active lines of the line wave protocol. Then for any point p on the field, if p is covered in the line protocol, it is also covered in the stripe protocol. From Theorem 6 and Lemma 3, we can have the following two corollaries.

Corollary 2: In the stripe wave protocol, the sufficient phase of a moving object is at most $2P$, where P is the wave scanning period. In other words, the moving object can always be detected in a duration of $2P$.

Corollary 3: The worst-case x-axis stealth distance is less than $2vP$. If the object moves along a straight line with an even speed, then the worst-case x-axis stealth distance is less than L.

C. Distributed Wave Protocol

We further design a distributed wave protocol. Compared with the line wave and stripe wave protocols, the distributed wave protocol has the following advantages: (1) it is completely distributed, and only involves local communications on each node, and (2) it does not need global timer synchronization among nodes.

1) Hot Regions and Wave Fronts: In this distributed wave protocol, there are two continuous active curves with two ends on two opposite borders of the field. These two curves scan the sensing field periodically, so that every point can be covered at least once during one wave scanning period. A set of sensors wake up to cover these curves. We define the *hot region* of an active curve as the union of sensing ranges of active sensors covering this curve, and define the *wave front* of the curve as the boundary of its hot region in its moving direction. Fig. 15 illustrates the wave front of a hot region moving to the right. Since the active curve is continuous, the wave front is continuous as well. For an active curve scanning the field from left to right, its wave front also moves from left to right.

2) Active Curves Move Forward: Here we describe how an active curve moves forward in our distributed wave protocol. Consider an active sensor s that has part of its sensing circle on the wave front of the active curve. As shown in Fig. 16, we define the wave front curve of s as the part of its sensing circle on the wave front. Before s goes to sleep, it finds all nodes whose sensing ranges intersect its wave front curve to wake up. After those sensors become active, part of their sensing circles become part of the new wave front. For example, in Fig. 16, before s goes to sleep, it finds node t and node o to wake up because the sensing ranges of t and o intersect the

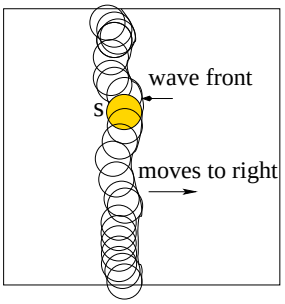


Fig. 15. The wave front moves in the distributed wave protocol.

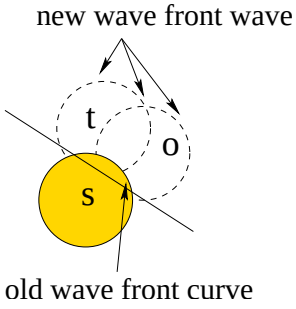


Fig. 16. An active sensor activates a set of nodes to cover its wave front curve.

wave front curve of s. In this way, the wave front of an active curve always moves forward, eventually it reaches the center vertical line of the field. The same process repeats afterwards.

Note that a sensing hole is a set of continuous points on the field that have not been covered in one scanning period. We have the following theorem.

Theorem 7: In the distributed wave protocol, the wave front of an active curve can scan the whole sensing field in an finite time without leaving any sensing hole.

Proof: Consider an active sensor s that has part of its sensing circle on the wave front. We claim that s can always find a set of sensors that have not waked up in current scanning period to cover s's wave front curve.

Let $P(t)$ be the set of points on the field that have been sensed between time 0 and t in current scanning period. Then we have $P(t) \subset P(t+\Delta t)$, where Δt is a time increment. In other words, the wave front always moves forward and does not go back. For any point p on the field, $p \in P(t) \Rightarrow p \in P(t+\Delta t)$. This implies that if a point $p \in P(t)$, then p is behind the wave front at time $t+\Delta t$. Therefore, the sensors that had already waked up and gone back to sleep in the past cannot cover points on the current wave front. Since any point on the field is within the sensing range of some sensor, there must exist a set of sensors that can cover s's wave front. Thus, we can find sensors that had not waked up to cover s's front wave curve at time $t+\Delta t$. On the other hand, according to the design of this distributed wave protocol, the wave front is continuous with two ends on the opposite borders of the field. Therefore, no sensing hole will be created in this distributed protocol. □

Lemma 4: In one scanning period, every node on the field wakes up exactly once, and consumes the same amount of energy given that they stay awake for the same amount of time.

Proof: We assume that no two sensor nodes are located at the same geographical coordinates. We only consider one of the active curves, since the proof can be applied to the other curve due to the symmetry. When a node goes to sleep, it always activates those nodes ahead of the wave front to cover its wave front curve. We use induction to prove that the nodes behind the wave have already waked up once.

- Base. At system startup time, the active curve is on a side border of the square. Only a set of nodes wake up to cover this curve, and all other nodes have not waked up yet.
- Induction step. Suppose at time t, all nodes behind the wave front have waked up once and only once. Consider the next earliest moment that one active node on the wave front goes to sleep. It activates all nodes that can partly cover its wave front curve. Because these newly-activated sensors are ahead of the wave front, they were in sleeping mode before time t, and have just waked up at t.

On the other hand, if a node has not waked up yet, its sensing range must intersect the wave front curve of some node m at some moment t', where t' is less than the scanning period. Therefore, it will be activated by node m at some moment. □

We directly obtain the following conclusion from Lemma 4.

Corollary 4: The scanning period of this distributed wave protocol is less than $wt \cdot n$, where wt is the active duration of nodes in one scanning period and n is the total number of nodes on the sensing field.

D. Evaluation of Wave Protocols

We conduct extensive simulation experiments to verify our analyses and to evaluate the performance of the wave protocols. We assess the average-case object detection quality based on the simulation results of DP and ASD.

	$d\uparrow$	$R\uparrow$	$v\uparrow$	$t_a\uparrow$	$wt\uparrow$
DP	$\rightarrow$	$\rightarrow$	$\uparrow$	$\uparrow$	$\downarrow$
ASD	$\rightarrow$	$\rightarrow$	$\uparrow$	$\rightarrow$	$\uparrow$

TABLE III
DP AND ASD CHANGE WHEN PARAMETERS INCREASE IN THE MODEL.

In our experiments, we generate a 200×200 grid field, and randomly place $d \times 40,000$ sensors on it. One constraint on these sensors is that when all of them are active, their sensing ranges should be able to cover the whole sensing field. A small object moves along a straight line towards a randomly selected direction with a constant speed v. We generate two active sensing lines or stripes at the two borders of the field moving towards the center periodically. We run each simulation for hundreds of times. We use the ratio of times of detection over the number of experiments to estimate DP, and use the average

non-detecting distance to estimate ASD. Since we have given upper bounds on the SP and the WSD in the protocol design part, we do not evaluate them in our experiments. Effects of system parameters on DP and ASD of the line wave and stripe wave protocols are listed in Table III.

1) Comparison of the Three Wave Protocols: Different from the line wave and stripe wave protocols, the wave scanning period of the distributed wave protocol depends on the geographical locations of the nodes. We compare the wave scanning period of these three protocols under the following parameter setting: $d = 0.3, R = 1.5, wt = 0.5$, and $v_w = 5.4$. We find that $P_{line} = 74.8$, $P_{stripe} = 75.3$, and $P_{dist} = 71.5$. This means the distributed wave protocol scans the field faster than the other two protocols at the cost of extra energy consumption.

To compare the DP, ASD, and energy consumption of different wave protocols, we use the same set of parameters except P for one simulation scenario. In the line wave and stripe wave protocols, ad is slightly less than $2R$. Note that $L = 200$, and $P = \frac{2L}{v_w}$.

- *DP and ASD Results*
  In all our experiments on DP, we restrict that $t_a < P$ to make sure that DP varies between 0 and 100%. Fig. 17 and 18 demonstrate that all three protocols have close DP and ASD results. However, the distributed wave protocol performs slightly different from the other two protocols, it has a higher DP and a lower ASD. When either v or t_a increases, DP increases too, which is shown in Fig. 17(a). Fig. 17(b) shows that a larger wt incurs a smaller DP. On the other hand, a larger v incurs a larger ASD, as shown in Fig. 18(a). Interestingly, the ASD increases linearly when node settling time wt increases, as we can observe from Fig. 18(b). This is because for a larger wt, it takes longer for an active line or stripe to scan the field than a smaller wt, thus, the object can travel a longer distance.
- *Node Energy Distribution Result*
  In our experiments, all nodes have the same amount of energy E at the beginning, and node energy consumption rate is $er = C \cdot R^3 \cdot \frac{2R}{v_w} / \frac{L}{2v_w} = \frac{4CR^4}{L}$, where C is constant being dependent on hardware design of the sensor nodes. We set $C = 0.00625$. We draw the node energy cumulative distribution in Fig. 18(c) to further show energy variance among nodes. For any curve point in this figure, its x value represents the node remaining energy, and its y value represents the number of nodes with energy less than the value specified by the x-axis. We observe that the remaining energy of most nodes is around the average node energy of the network. On the other hand, the node energy distribution of the distributed wave protocol has a narrower range than those of the other two wave protocols. For example, 90% nodes in the line wave protocol have node remaining energy in the range [408-413], and 70% nodes in the stripe wave protocol have node remaining energy in the range [404-409]. On the contrary, in the random sensing schedule, 70% nodes have node remaining energy in the range [411.5-412.5]
- *Comparison of Line Wave, Random, and Synchronized Schedules*
  In Section IV, we have formally studied the random sensing schedule and the synchronized sensing schedule. In Fig. 19 and 20, we compare the DP and ASD results of the line wave schedule, the random schedule, and the synchronized schedule, respectively, when varying v and fixing all other parameters. We observe that, with a small v, the line wave schedule and the random schedule have a larger DP than the synchronized sensing schedule; however, as v increases, the synchronized schedule begins to catch up the line wave schedule, and eventually outperforms it. Similarly, the line wave schedule has a smaller ASD for small v. When v increases, the ASD of the line wave schedule exceeds that of the random sensing schedule, and the synchronized schedule has a larger ASD than the other two schedules.

VI. Conclusion

Balancing object detection quality and network lifetime is a challenging task in sensor networks. Under partial coverage, we develop an analytical model for object detection applications, and mathematically study average-case object detection quality of the random and synchronized sensing scheduling protocols with respect to various network conditions. Aiming to achieve bounded worst-case object detection quality, we propose and analyze three wave sensing scheduling protocols, and formally prove the bounds on worst-case object detection quality of these protocols. Our proposed protocols and their analyses characterize the interactions among network parameters, average-case and worst-case object detection quality, and energy consumption of the protocols. Our analyses can help to plan a sensor network with average-case and worst-case object detection quality requirements and stringent node power budget, and can direct new sensing scheduling protocol design as well.

Acknowledgments

The authors would like to thank anonymous reviewers for their constructive comments, and thank William L. Bynum for reading the paper and his comments. Shansi Ren and Xiaodong Zhang are partially supported by the National Science Foundation under grants CNS-0098055, CNS-0405909, and CNS-0509054/0509061. Qun Li is supported in part by US National Science Foundation grant CCF-0514985. Some preliminary results of this work have been presented in [24], and [25].

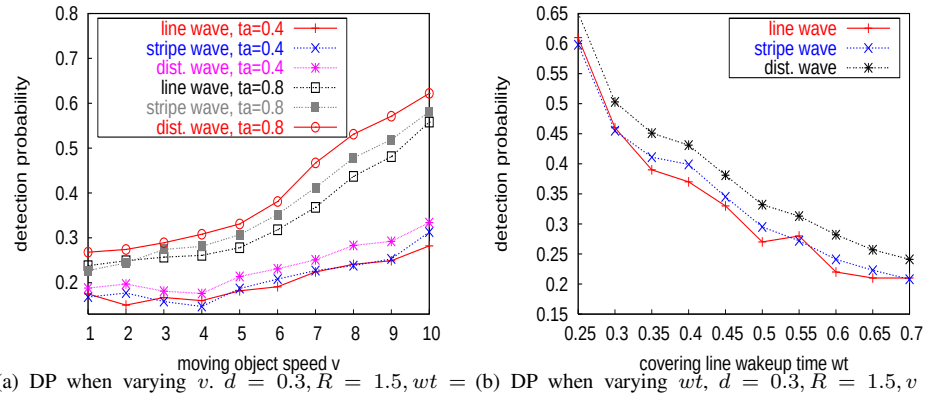
References

[1] Z. Abrams, A. Goel, and S. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of IPSN'04*, pages 424–432, Berkeley, CA, USA, April 2004.

[2] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus. Tracking a moving object with a binary sensor network. In *Proceedings of ACM SenSys'03*, pages 150–161, Los Angeles, CA, USA, November 2003.

[3] R. Brooks, P. Ramanathan, and A. Sayeed. Distributed target classification and tracking in sensor networks. *Proceedings of the IEEE*, 91(8):1163–1171, August 2003.

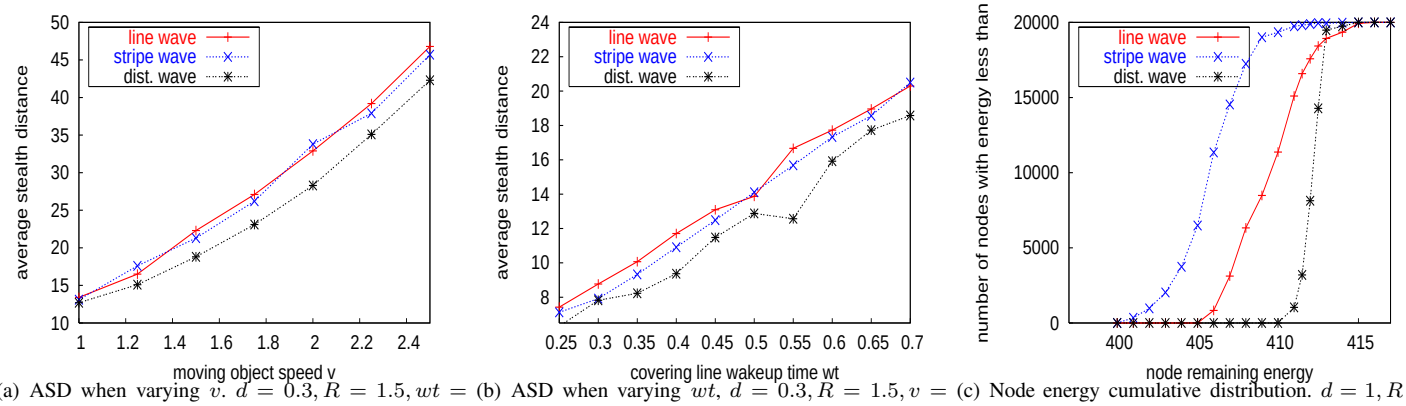
[4] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proceedings of IPSN'05*, pages 20–27, Los Angeles, CA, USA, April 2005.



(a) DP when varying v. $d = 0.3, R = 1.5, wt = 0.05, v_w = 50$.

(b) DP when varying wt, $d = 0.3, R = 1.5, v = 1, t_a = 8, v_w = 5$.

Fig. 17. DP of the wave protocols when varying different parameters.



(a) ASD when varying v. $d = 0.3, R = 1.5, wt = 0.5, v_w = 50$.

(b) ASD when varying wt, $d = 0.3, R = 1.5, v = 1, v_w = 5$.

(c) Node energy cumulative distribution. $d = 1, R = 1, wt = 1$.

Fig. 18. ASD and node energy distribution when varying different parameters.

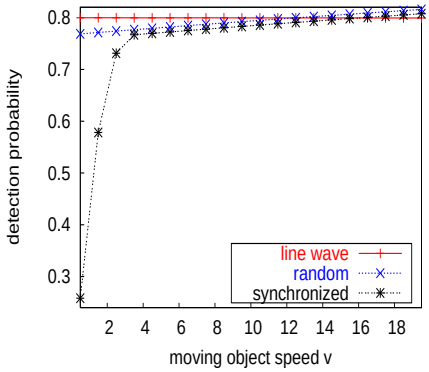


Fig. 19. DP when varying v. $d = 1, R = 0.75, t_a = 0.4, wt = 0.01, v_w = 2$.

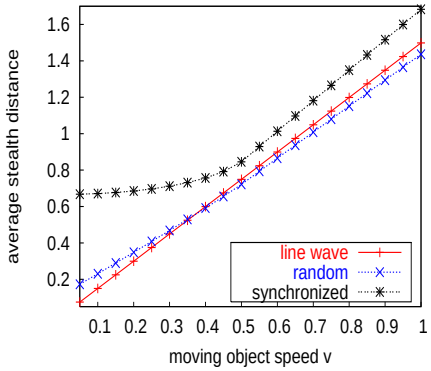


Fig. 20. ASD when varying v. $d = 1, R = 0.75, wt = 0.01, v_w = 2$.

[5] Q. Cao, T. Yan, J. Stankovic, and T. Abdelzaher. Analysis of target detection performance for wireless sensor networks. In *Proceedings of IEEE DCOSS'05*, pages 276–292, Marina del Rey, CA, USA, July 2005.

[6] P. B. Godfrey and D. Ratajczak. Naps: Scalable, robust topology management in wireless ad hoc networks. In *Proceedings of IPSN'04*, pages 443–451, Berkeley, CA, USA, April 2004.

[7] C. Gui and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *Proceedings of ACM MobiCom'04*, pages 129–143, Philadelphia, PA, USA, September 2004.

[8] R. Gupta and S. R. Das. Tracking moving targets in a smart sensor network. In *Proceedings of IEEE VTC Fall 2003 Symposium*, volume 5, pages 3035–3039, October 2003.

[9] P. Hall. *Introduction to the Theory of Coverage Processes*. John Wiley and Sons, 1988.

[10] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. An energy-efficient surveillance system using wireless sensor networks. In *Proceedings of USENIX MobiSys'04*, pages 270–283, Boston, MA, USA, June 2004.

[11] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *Proceedings of ACM/USENIX MobiSys'04*, pages 270–283, Boston, MA, USA, June 2004.

[12] C. Hsin and M. Liu. Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithms. In *Proceedings of IPSN'04*, pages 433–442, Berkeley, CA, USA, April 2004.

[13] D. Li, K. Wong, Y. Hu, and A. Sayeed. Detection, classification and

tracking of targets in distributed sensor networks. 19(2):17–30, March 2002.

[14] Q. Li and D. Rus. Global clock synchronization in sensor networks. In *Proceedings of IEEE INFOCOM'04*, pages 564–574, Hong Kong, China, March 2004.

[15] B. Liu and D. Towsley. A study on the coverage of large-scale sensor networks. In *Proceedings of IEEE MASS'04*, pages 475–483, Fort Lauderdale, FL, USA, October 2004.

[16] J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao. Distributed group management for track initiation and maintenance in target localization applications. In *Proceedings of IPSN'03*, pages 113–128, Palo Alto, CA, USA, April 2003.

[17] A. Mainwaring, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *ACM Workshop on Wireless Sensor Networks and Applications'02*, pages 88–97, Atlanta, GA, USA, September 2002.

[18] S. Megerian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Worst and best-case coverage in sensor networks. 4(1):84–92, January–Febuary 2005.

[19] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of ACM SenSys'04*, pages 50–61, Baltimore, MD, USA, November 2004.

[20] E. Onur, C. Ersoy, and H. Delic. How many sensors for an acceptable breach detection probability? *Computer Communications (Special Issue on Dependable Wireless Sensor Networks), In Press, Corrected Proof, available online July 14, 2005*.

[21] E. Onur, C. Ersoy, and H. Delic. Quality of deployment in surveillance wireless sensor networks. *International Journal of Wireless Information Networks (Special Issue on Ad Hoc Sensor Networks)*, 12(1):61–67, January 2005.

[22] S. Pattem, S. Poduri, and B. Krishnamachari. Energy-quality tradeoffs for target tracking in wireless sensor networks. In *Proceedings of IPSN'03*, pages 32–46, Palo Alto, CA, USA, April 2003.

[23] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, March 2002.

[24] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang. Analyzing object detection quality under probabilistic coverage in sensor networks. In *Proceedings of IWQoS'05*, pages 107–122, Passau, Germany, June 2005.

[25] S. Ren, Q. Li, H. Wang, and X. Zhang. Design and analysis of wave sensing scheduling protocols for object-tracking applications. In *Proceedings of IEEE DCOSS'05*, pages 228–243, Marina del Rey, CA, USA, July 2005.

[26] A. Savvides, C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of ACM MobiCom'01*, pages 166–179, Rome, Italy, July 2001.

[27] X. Sheng and Y. Hu. Energy based acoustic source localization. In *Proceedings of IPSN'03*, pages 285–300, Palo Alto, CA, USA, April 2003.

[28] Q. Wang, W. Chen, R. Zheng, K. Lee, and L. Sha. Acoustic target tracking using tiny wireless sensor devices. In *Proceedings of IPSN'03*, pages 642–657, Palo Alto, CA, USA, April 2003.

[29] T. Yan, T. He, and J. Stankovic. Differentiated surveillance for sensor networks. In *Proceedings of ACM SenSys'03*, pages 51–62, Los Angeles, CA, USA, November 2003.

[30] 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 IEEE ICDCS'03*, pages 28–37, Providence, RI, USA, May 2003.

[31] H. Zhang and J. Hou. On deriving the upper bound of α-lifetime for large sensor networks. In *Proceedings of ACM MobiHoc'04*, pages 121–132, Roppongi, Japan, May 2004.

[32] W. Zhang and G. Cao. Optimizing tree reconfiguration for mobile target tracking in sensor networks. In *Proceedings of IEEE INFOCOM 2004*, volume 4, pages 2434–2445, Hong Kong, China, March 2004.

[33] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, 19(2):61–72, March 2002.