This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

A New Distributed Algorithm for Even Coverage and Improved Lifetime in a Sensor Network

Xiaoyu Chu and Harish Sethu
Department of ECE, Drexel University
3141 Chestnut Street
Philadelphia, PA 19104-2875, USA.
{ xiaoyu.chu, sethu }@drexel.edu

***Abstract*—In "area-sensing" applications of sensor networks, such as surveillance or target tracking, each sensor node has a sensing radius within which it can monitor events. Coverage problems in sensor networks have largely focused on such applications, where the goal of good coverage is one of ensuring that each point in the region of interest is within the sensing radius of at least one node. On the other hand, in "spot-sensing" applications, each node makes a measurement (such as temperature or humidity) at the precise location of the node and there is no concept of a sensing radius. In this paper, we introduce a new coverage problem that is more meaningful to spot-sensing applications. In such cases, good coverage usually implies *even* coverage across points in the region. We borrow from the field of economics and adapt a well-accepted measure of inequality, the Gini index, to develop a metric for the evenness of coverage by a sensor network. Based on mathematical results on the expected distances between neighboring nodes, we present a new distributed algorithm, called EvenCover, for each node to determine if and when it should sleep or sense. We prove that the expected Gini index is $1 - 1/\sqrt{2} \approx 0.293$ when the spatial distribution of sensing nodes is given by a Poisson random process. On the other hand, when the sensing nodes are perfectly evenly distributed, we show that the Gini index has a lower bound of 0.2. These two results serve as points of reference to evaluate the coverage achieved by the EvenCover algorithm. We present a thorough simulation-based comparison of EvenCover against other distributed algorithms showing that it achieves better evenness and significantly increased lifetime. In addition, we discover that evenness of coverage permits a graceful degradation of the network as nodes exhaust their energy resources.**

I. Introduction

Networks of inexpensive low-power sensor nodes may be deployed to sense, process and gather information in a region of interest for a variety of purposes including surveillance, target tracking, pollution studies and wildlife monitoring [1]. The lifetime of these sensor nodes, and therefore of the sensor network, depends critically on the energy consumed in the operations required for the intended application. To conserve energy, a sensor node can choose to go into a *sleep* mode in which its sensing and communication modules are turned off. Sensor nodes can be designed to execute a routine at specific intervals of time to make a choice on whether they should be in the *sense* mode (also referred in this paper interchangeably as the *active* mode) or the sleep mode until the next time they execute the same routine. A key constraint on the decision process in the executed routine is to achieve good coverage of the region of interest. In some applications, however, hand-placing sensors or manually activating sensors at pre-selected locations is not feasible because it is prohibitively labor-intensive or the environment is unreachable or hazardous. This calls for distributed algorithms for the coverage problem executed by all the sensor nodes to automate the process of determining which sensor nodes should go active and which ones should go into a sleep mode. The sensor area coverage problem is a well-researched one with many distributed approaches discussed in the literature [2]–[14].

As far as the coverage problem is concerned, we argue that sensor network applications may be divided into two classes: those in which each sensor node makes observations of a designated area around it (area-sensing applications) and those in which each node makes measurements of physical phenomena at precisely the spot where it is located (spot-sensing applications). Area-sensing applications include enemy surveillance, target tracking and intrusion detection in which sensor nodes monitor a "sensing area" using vision, sound, seismic-acoustic energy, infrared energy, or magnetic field changes. In these applications, each node has a local region within which it can detect events and make relevant observations. On the other hand, spot-sensing applications involve each sensor node making measurements of physical phenomena such as temperature, humidity and environmental pollution at the exact location of the node. In these kinds of applications, there is no concept of a "sensing area" because the measurements made are of physical properties at the location of the node itself.

Past research on the sensor area coverage problem has largely assumed a system model that is more appropriate for area-sensing applications [9]–[14]. The system model used is one where each sensor node has a pre-defined sensing range. A location in the region of interest, say point p, is said to be covered by the network if it is within the sensing range of at least one sensor node that is active. A region is considered covered if all points within it are covered. This definition of coverage, while useful as a first approximation of the intuitive notion of coverage in area-sensing applications, is not a valid one for spot-sensing applications where the coverage at a point cannot be easily captured in a binary manner as either covered or not covered. In the absence of a concept of "sensing area",

978-1-4244-3513-5/09/$25.00 ©2009 IEEE

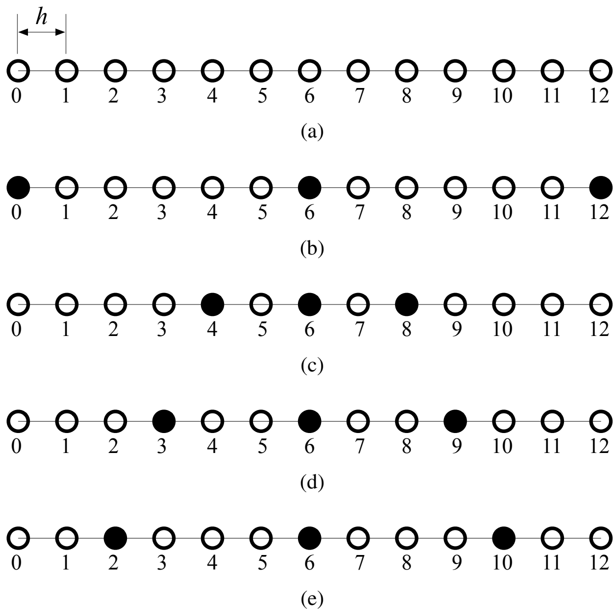


Fig. 1: A toy example illustrating the choice of a metric for evenness of coverage: (a) with no nodes in active mode; (b)–(d) with different sets of three nodes in active mode, each achieving a different level of evenness of coverage; (e) with the set of three nodes in active mode that yields the best evenness of coverage.

the quality of coverage at a point also cannot be accurately captured as the number of active nodes within whose "sensing area" the point belongs. Some researchers have improved upon the area-sensing model by defining a point as either covered or uncovered with a probability that is a function of the distance to the nearest sensor node [13]. However, even this model ends up imposing a binary either-or assessment of coverage at a point. Instead, we need a metric that describes the quality of representation that each point in the region enjoys in the spot-sensing measurements made by the active sensor nodes around it, best captured as a continuous function of the distances to the nearest set of active sensor nodes around the point. To a spot-sensing application, the value of the coverage achieved by a network can then be measured by the evenness of coverage across all points in the region of interest.

In this paper, we formulate a new coverage problem appropriate for spot-sensing applications where we care most about the evenness of the quality of coverage across all points in the region of interest. We present distributed algorithms to solve the problem for any desired spatial granularity of coverage and show that improving the evenness of coverage has the additional benefit of significantly improved network lifetime.

A. Problem Statement

Consider n sensor nodes distributed within a certain region of interest, denoted by R. Let $G' = (V', E')$ denote the graph of these sensor nodes where each node $u \in V'$ represents a sensor node and each edge $(u, v) \in E'$ represents the fact that nodes u and v are neighbors and can communicate directly with each other. Assume that the graph G' is the result of an effective topology control algorithm. (Topology control is the problem of determining the transmission ranges of nodes in order that nodes may transmit at a lower power than the maximum to reduce energy consumption and the likelihood of interference [15].) Let $d(u, v)$ denote the Euclidean distance between sensor nodes represented by vertices u and v. The Euclidean distances between nodes are computable if the nodes are all fitted with low-power GPS receivers, or through location estimation techniques if only a subset of nodes are equipped with GPS receivers [16], or by estimating distances based on exchanging transmission and reception powers [17].

Let z denote the desired *coverage resolution*, in number of active nodes per unit area, determined based on the spatial granularity with which the physical phenomenon of interest should be sensed. Let $G = (V, E)$ denote the subgraph of G' such that $v \in V$ iff vertex v represents an active node (as opposed to one in the sleep mode). The problem now is one of determining G in a distributed manner so that we achieve evenness of coverage at the desired coverage resolution. In this paper, we do not require that G be a connected graph (retrieval of data from sensor nodes for spot-sensing applications is more often done through mobile gateways; connectivity is more important in area-sensing applications which track and report events [18]).

We now complete the problem statement by providing a metric for the evenness of coverage, which we illustrate in the following with a toy example. Consider a region of interest that is a single line (as opposed to the usual two- or three-dimensional regions of interest one encounters in real situations) within which are located thirteen sensor nodes as shown in Fig. 1a. Let h denote distance between any two neighboring nodes. Suppose that, based on the desired coverage resolution, we determine that we have to activate exactly three out of these thirteen nodes (we denote active nodes by black circles and sleeping nodes by empty circles). The question is: which set of three nodes yields the best spatial uniformity of coverage? At this juncture, it is important to mention that evenness of coverage is more about the points in the region of interest and less about the distances between neighboring active nodes. Therefore, metrics used traditionally for uniformity, such as the ones based on the average distance between an active node and its nearest active neighbor [19], do not serve the purpose here for spot-sensing applications (as opposed to those based on distances between points in the region to the respective nearest active node).

Now, if we choose our metric as the average distance of points to the nearest active node, we cannot distinguish between the assignments in Figs. 1b and 1c because in both cases the metric will yield an average of $3h/2$. Similarly, if we instead choose our metric based on the worst-case distance of a point to the nearest active node, the assignments in Figs. 1b and 1d will both yield $3h$ and we will not be able to distinguish between the two. Thus, neither the worst-case distance nor the average distance of a point to the nearest active node can serve as a good metric for the evenness of coverage. Meanwhile, the reader's intuition may suggest that the assignment of Fig. 1e offers the most uniform coverage. In the following, we present a formal expression of this intuitive notion of uniform coverage borrowing from the experience of economists in measuring inequality.

The field of economics has a long history of measuring inequality and a vast body of literature on the topic [20], [21]. We use a popular and well-accepted metric in economics, the Gini index, based on the relative mean difference between the quantities being compared (in our case, the quantities are distances of points to their respective nearest active sensor nodes). Consider m quantities, $g_1 \leq g_2 \leq \cdots \leq g_m$. The mean difference between these quantities is:

$$\Delta = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |g_i - g_j|$$

The relative mean difference is the mean difference divided by the mean, $\overline{g}$. The Gini index is defined as one-half of the relative mean difference, i.e.,

$$\text{Gini index} = \frac{\Delta}{2\overline{g}} = \frac{1}{2\overline{g}m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} |g_i - g_j|$$

Let p denote a point in the region of interest R. Let $d_p(v)$ denote the distance of node v from point p. Let $n_p(G)$ denote the nearest node in G (the set of active nodes) from point p. Adapting the Gini index to the context of our problem poses one issue: the number of quantities we have is infinite because there are an infinite number of points in any region of interest. Therefore, instead of using summations as above, we consider expected values in defining evenness. Let $\overline{d_R(G)}$ denote the expected distance of a point in region R to the nearest node in G. Define p and q as two arbitrary random points in the region. We define the evenness, $U(G, R)$ of graph G in the region R as:

$$U(G, R) = \frac{E[|d_p(n_p(G)) - d_q(n_q(G))|]}{2\overline{d_R(G)}} \tag{1}$$

where $p, q \in R$. The smaller the value of the above quantity, the better the spatial uniformity, i.e., the evenness of coverage.

Given G', the problem can now be stated as one of developing a distributed algorithm to find G such that $U(G, R)$ is minimized.

B. Contributions and Organization

We propose a new coverage problem described in the previous subsection specifically for spot-sensing applications in sensor networks. Section II discusses work in sensor networks as well as in other fields which seek to solve similar underlying mathematical problems. Section III-A derives results on the expected distances between neighboring active nodes. Section III-B uses the results of Section III-A to develop a distributed algorithm, called EvenCover, for even coverage of points in the region of interest.

Section IV presents several simulation results on the performance of EvenCover and some other representative algorithms. The plots in this section show that the EvenCover algorithm achieves very good evenness in comparison to other algorithms. In fact, EvenCover comes closest to the best possible evenness (we prove a bound on the metric of evenness, the Gini index, in the Appendix). Quite importantly, we show that EvenCover also achieves a better network lifetime than other distributed coverage algorithms. Most importantly, we show that the EvenCover algorithm degrades gracefully, preserving the evenness as nodes exhaust their energy resources and die.

Section V concludes the paper with brief remarks on the EvenCover algorithm and future work.

II. Related Work

The problem of designating the mode of a sensor node as either active or sleeping is related (though not identical) to the 2-color instance of some versions of the distributed graph coloring problem [22], [23], in which each node takes on one of two colors with the goal to minimize the number of neighbors of the same color as itself. While the algorithms in this body of work will generally improve the spatial uniformity of active nodes, they do not consider the distances between the nodes in their computations and therefore, are limited in their application to the problem under consideration. We show this later in Section IV by simulating the *Flip* algorithm, an adaptation of the algorithm in [24], in which each node begins with randomly assigning itself one of two modes, and then, at random intervals of time, switches to the mode that best approximates the active node ratio in its neighborhood.

Spatially uniform distribution based on distances is more explicitly considered in another body of work related to the problem of facility location [25], [26]. The problem involves the location of facilities in an environment (such as emergency services in a city) given some constraints and an objective function. In the field of networking, related problems have been solved in the context of content distribution networks where one has to replicate resources in multiple locations (servers on a network) to boost performance by minimizing delay from users to the nearest resource or by achieving load balancing on the network [27]–[30]. A variety of techniques, including graph-theoretic approaches, heuristic algorithms and dynamic programming, have been employed in these works to arrive at a solution. Ko and Rubenstein developed the first distributed algorithm for the placement of replicated resources, best described as a solution to the distributed graph coloring problem, by having each node continually change its color in a greedy manner to maximize its own distance to a node of the same color [31]. This work, which considers the distance between two nodes as that along the communication path and *not as the geographical distance*, cannot be directly applied to the problem of uniform coverage considered in this paper. A further reason this body of work does not directly apply here is that they only consider the relationships between nodes and not between the nodes and the points in the region of interest.

Points in the region of interest are most explicitly considered in the set of works that propose coverage algorithms for sensor networks based on assuming a sensing area for each node [5], [9]–[14]. The goal is usually to ensure that each point in the region of interest is within the sensing area of at least k active sensor nodes. Distributed algorithms to achieve k-coverage also happen to improve the evenness of coverage although they do not specifically attempt it. Better evenness is achieved because, on average, this distributes the active nodes across the entire region of interest (in Section IV, we will compare the evenness and lifetime of a representative member of this class of algorithms with the one proposed in this paper.)

III. The EvenCover Algorithm

The design of a distributed algorithm for the problem stated in Section I-A requires that a node make an estimate of the quality of coverage in its local area in comparison to the desired level of coverage to reliably determine if it should sleep or go active. A node, therefore, needs to know the expected distances to the nearest active neighbors in a random spatial distribution of active nodes and compare it against the actual distance. While the distance to the nearest active node is important, it alone does not yield sufficient information to estimate the quality of coverage within the local region. Our distributed algorithm, therefore, employs the actual and the expected distances to the k-th nearest active node, for k from 1 to the number of neighbors within the communication range of the node.

A. Mathematical Foundations

Consider active sensor nodes randomly distributed in the region of interest, R, given by a Poisson process of rate z active nodes per unit area. Therefore, the probability that we will have k nodes within some area S is given by:

$$P(k,S) = \frac{(zS)^k e^{-zS}}{k!} \tag{2}$$

In the following, we assume that the region of interest is large enough to ignore boundary issues.

Theorem 1: Given a Poisson distribution of active nodes with z active nodes per unit area, the expected distance of an active node to its k-th nearest active neighbor is:

$$\left(\frac{k}{2^{2k}\sqrt{z}}\right)\binom{2k}{k}$$

Proof: From Eqn. (2), the probability that there are $k-1$ active nodes within a radius of r is given by:

$$P(k-1,\pi r^2) = \frac{(z\pi r^2)^{k-1} e^{-z\pi r^2}}{(k-1)!} \tag{3}$$

Consider a ring of radius r of infinitesimal area equal to $2\pi r dr$. Using Eqn. (2) again and noting that $2\pi rzdr \to 0$ implies $e^{-2\pi rzdr} \to 1 - 2\pi rzdr$, the probability that there is exactly one active node on this ring is given by:

$$P(1,2\pi rdr) = \frac{(2\pi rzdr)^1 e^{-2\pi rz\,dr}}{1!} \approx 2\pi rz\,dr \tag{4}$$

Let X_k be a random variable indicating the distance between an active node and its k-th nearest active neighbor for $k \geq 1$. Using Eqns. (3) and (4), we get:

$$\begin{aligned} E[X_k] &= \int_0^\infty rP(k-1,\pi r^2)P(1,2\pi rdr) \\ &= \int_0^\infty r\left(\frac{(z\pi r^2)^{k-1}e^{-z\pi r^2}}{(k-1)!}\right)2\pi rzdr \\ &= \frac{2}{(k-1)!}f(k) \end{aligned} \tag{5}$$

where $f(k)$ is given by:

$$f(k) = \int_0^\infty (z\pi r^2)^k e^{-z\pi r^2} dr \tag{6}$$

Using routine calculus employing integration by substitution, $f(k)$ can be simplified recursively as:

$$\begin{aligned} f(k) &= \frac{2k-1}{2}\int_0^\infty (z\pi r^2)^{k-1} e^{-z\pi r^2} dr \\ &= \frac{2k-1}{2} f(k-1) = \frac{(2k)!}{k!2^{2k-1}} f(1) \\ &= \frac{(2k)!}{k!\,2^{2k+1}}\sqrt{\frac{1}{z}} \end{aligned} \tag{7}$$

Combining Eqns. (5) and (7), we get:

$$E[X_k] = \left(\frac{k}{2^{2k}\sqrt{z}}\right)\binom{2k}{k} \tag{8}$$

■

B. EvenCover

Let r be the communication radius of the nodes (i.e., all nodes within distance r are considered neighbors). Let D denote the *active node density*, the fraction of nodes in the region of interest that are active. As before, let N denote the total number of nodes in the region, A the area of the region and z the desired coverage resolution. Note that z is a property of the application and not of the sensor network used by the application, while D describes the state of the sensor network. Let $D_z = zA/N$ denote the desired active node density.

The pseudo-code for the algorithm is shown in Fig. 2. Each node can compute the quality of coverage (QoC) at the point where it is located based on a comparison between the expected and actual distances to its nearest neighbors. A node should stay in its current mode or switch to a different mode depending on whether or not the action taken helps bring the QoC computed by it closer to what one might expect in a random spatial distribution of sensing nodes. On the other hand, if the QoC is worse than expected, the node should go into the sense mode if not already in the sense mode. Let QoC(i) denote the QoC computed by node i. The issue now becomes one of computing the QoC, which we define as:

$$QoC(i) = \delta(i) + \sum_{k=1}^{H(i)} \frac{E[X_k(i)]}{X_k(i)} \tag{9}$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

```
01: Initialization:
02:     Turn on sense mode with probability D_z.

03: EvenCover Algorithm (executes in a loop):
04:     do:
05:         if node is active:
06:             QoC = 1
07:         else:
08:             QoC = 0
09:         Wait for a length of time equal to random(0, T)
10:         Compile list of active neighbors
11:         for 1 ≤ k ≤ number of active neighbors:
12:             X_k = distance to k-th nearest active neighbor
13:             QoC = QoC + E[X_k]/X_k
14:         if ( QoC > zπr^2):
15:             if ( QoC −zπr^2 ≥ 0.5 ):
16:                 Set node to sleep mode
17:             else:
18:                 Set node to sleep mode with
                        probability (QoC −zπr^2)
19:         else:
20:             if ( zπr^2− QoC ≥ 0.5 ):
21:                 Set node to active mode
22:             else:
23:                 Set node to active mode with
                        probability (zπr^2− QoC)
24:     while true
```

Fig. 2: The EvenCover Algorithm

where $H(i)$ is the number of active neighbors of node i, $X_k(i)$ is the distance from node i to its k-th nearest active neighbor, and $\delta(i)$ is given as:

$$\delta = \begin{cases} 1, & \text{if node } i \text{ is in sense (active) mode,} \\ 0, & \text{if node } i \text{ is in sleep mode.} \end{cases} \tag{10}$$

Knowing the desired coverage resolution, z, the expected number of active neighbors in a circular region of radius r is readily seen to be $z\pi r^2$. If QoC(i) computed as above exceeds $z\pi r^2$ by 0.5 or more and the node is in active mode, turning it to the sleep mode will bring the local active node density closer to that corresponding to the desired coverage resolution. Similarly, if $z\pi r^2$ exceeds QoC(i) by 0.5 or more and node i is in sleep mode, turning it to the active mode will also bring the local active node density closer to that corresponding to the desired coverage resolution. If QoC(i) exceeds $z\pi r^2$ by less than 0.5, the algorithm sets the node to sleep mode with probability equal to QoC$(i) - z\pi r^2$ (this does not necessarily bring the local coverage resolution closer to the desired value but is an attempt to bring the overall active node density of the network closer to that corresponding to the desired coverage resolution). Similarly, if $z\pi r^2$ exceeds QoC(i) by less than 0.5, the node is set to active mode with probability $z\pi r^2 -$QoC(i).

Let a time interval, T, be a parameter used by the algorithm. The algorithm to turn on and off occurs at random instants of time on different nodes. In the beginning, each node independently selects a random starting instant in the range $(0, T)$ and executes the algorithm involving the computation of the QoC to determine if the node should turn on or off. Next, the node will wait for a random length of time, also in the range $(0, T)$, to execute the algorithm again. A node will continue to perform the algorithm and go through sleep/sense cycles until it runs out of battery or stops at any specific time defined by the administrative entity for the sensor network.

IV. Simulation Results

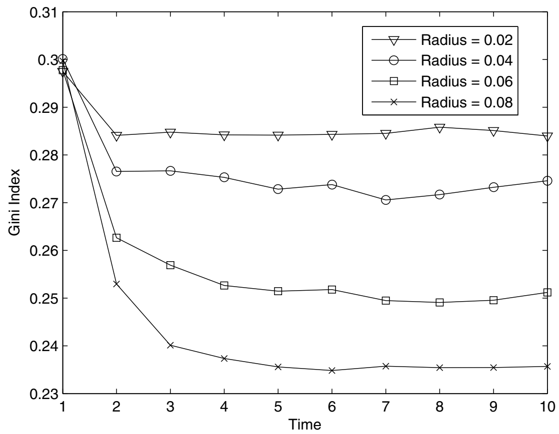
Our simulation experiments use 400 sensor nodes located in a square region of unit area with a spatial distribution given by a Poisson process (each point in the region is equally likely to have a node). We choose T as equal to 10 units of time. The coverage resolution, the corresponding density of active nodes and the communication radius used in the experiments are described as we discuss each of the simulation experiments in the following subsections.

Each data point reported in the figures in this section represents an average of 20 different randomly generated network graphs. The 95% confidence interval is within $\pm 1\%$ for all the data points reported in the graphs.

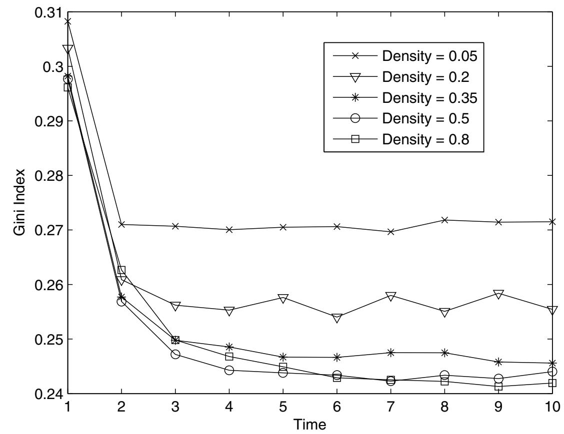
In the Appendix, we prove two results on the Gini index of a distribution of active nodes in a region. Theorem 2 proves that a lower bound on the Gini index is 0.2 (we believe this to be a loose bound). Theorem 3 proves that when the active nodes are located in the region with a spatial distribution given by a Poisson process, the Gini index is $1 - 1/\sqrt{2}$. This value of the Gini index is achieved when Poisson-distributed nodes decide to randomly set themselves to active mode with probability equal to D_z, the expected active node density when the desired coverage resolution is z. In the average case, such a straightforward coverage technique may yield fair results but when portions of the region are sparsely or densely covered, it fails to achieve the appropriate coverage resolution. Certainly, any distributed algorithm that seeks to achieve good evenness should be able to achieve a Gini index lower than $1 - 1/\sqrt{2} = 0.293$ after a short period of execution time. Therefore, one should expect that the Gini indices would reduce to something between 0.2 and 0.293 after a few duty cycles of the algorithm at each node.

A. Convergence

Fig. 3a shows the convergence of the EvenCover algorithm toward the lower bound on the Gini index for different values of the communication radius while the desired coverage resolution corresponds to an active node density equal to 0.5 (i.e., given 400 nodes in a unit area in our simulations, the desired coverage resolution, z, is 200 active nodes per unit area). Fig. 3a plots the Gini index as the algorithm continues to execute for a length of time equal to $10T$. Note that the Gini index as the algorithm begins to execute is close to its expected value of $1 - 1/\sqrt{2}$ since the algorithm begins with a random assignment of the sense/sleep mode to each sensor node. Note that the index reduces rapidly as early as $2T$ (about 2 executions of the algorithm, i.e., two rounds in the loop between lines 04–24 in Fig. 2). Even though the



(a) Gini index achieved by EvenCover plotted against time from 0 to $10T$ for different values of the communication radius. In these experiments, the desired coverage resolution used corresponds to an active node density of 0.5.



(b) Gini index achieved by EvenCover plotted against time from 0 to $10T$ for different values of the desired active node density. In these experiments, the communication radius used is 0.07.

Fig. 3: Plots showing the convergence of EvenCover as the algorithm executes and improves the evenness of coverage.

EvenCover algorithm does not require that all nodes use the same communication radius, we assume so in this experiment in order to gain insight into the performance of the algorithm as a function of the communication radius. As one might expect, the algorithm increases in its effectiveness as the communication radius increases. When the communication radius is small, most sensor nodes have little information about the quality of coverage in their local region and thus, end up making a less reliable judgement. When the communication radius is large, sensor nodes have more neighbors and are able to collect significantly more relevant information about the quality of coverage in their neighborhood, thus improving the performance of the algorithm.

In our second set of experiments on the convergence properties of EvenCover, we keep the communication radius constant and vary the desired active node density from 0.05 to 0.8. Given 400 nodes, most topology control algorithms that rely entirely on local estimation of distances between neighbors end up with 4–6 neighbors [32], [33]. Therefore, we use a communication radius of 0.07 units corresponding to an average of about five neighbors within the radius. The result is plotted in Fig. 3b for an interval of time up to $10T$. Once again, the algorithm appears to converge rapidly within time $2T$. Note that the algorithm performs better at higher densities because, with a larger proportion of active nodes, the evenness of coverage can be more easily fine-tuned by the algorithm.

B. Comparative analysis

We report results for the following distributed coverage algorithms:

- The EvenCover algorithm
- Sponsored Coverage: For a representative coverage protocol that assumes an area-sensing application, we choose the well-cited Sponsored Coverage protocol [34]. In this protocol, a node decides to go into the sleep mode if its entire designated sensing area is also covered by its neighbors. To avoid situations in which each of two neighbors expects a certain spot to be covered by the other, the protocol implements a random time for which each node delays its decision.
- Flip: In this protocol, each node counts the fraction of its neighbors (including itself) that are active and sets itself into either the sleep mode or the active mode depending on whether or not this fraction is larger or smaller than the desired active node density. We use this algorithm as representative of coverage strategies based on distributed algorithms for graph coloring that do not use the distance between the nodes in their computations [23].

From the point of view of the application, a fair comparison between these algorithms is achieved only when the communication radius assumed is the same and the resulting active node density is also the same at the end of the execution of the algorithm. The communication radius is easily set to the same value for each of the different algorithms. For the active node density, we note the density generated by the algorithms that do not take the density as an input parameter, and then we use the corresponding coverage resolution (z) as inputs into the EvenCover algorithm. The Gini index achieved by the algorithms are compared after the algorithms execute for a time period of $20T$. Fig. 4 plots the performance of these algorithms. As before, it is easy to see that the Gini index improves with larger communication radius since each node is able to collect more information about it.

Fig. 4 shows that the EvenCover algorithm achieves better evenness. The Sponsored Coverage algorithm seeks full, but not necessarily even, coverage of points in the region. As a result, in parts of the region with a dense cluster of nodes, the Sponsored Coverage algorithm will unnecessarily turn on larger numbers of nodes if the communication radius is

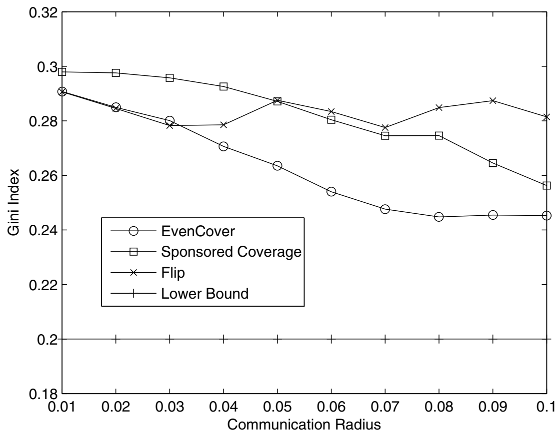


Fig. 4: The Gini Index achieved after time $20T$ by different coverage algorithms for different values of the communication radius. For each communication radius, the active node density achieved by different algorithms is the same in this figure.

small. When the communication radius is large, the Sponsored Coverage algorithm and EvenCover perform somewhat more similarly. The Flip algorithm, on the other hand, does not consider distances between nodes and therefore, is far from being able to achieve even coverage.

C. Network Lifetime

In this section, we compare the network lifetime of EvenCover with different algorithms. In our simulation experiments, we begin with each node allocated a certain amount of energy which is expended in the following three ways:

- On/off broadcast transmission, used to broadcast the node's new status (on/off) to its neighbors.
- Data transmission, used to transmit the sensed data to the data sinks.
- Reception, for receiving data and control information from neighbors.

In this set of experiments, we use an additional coverage algorithm, labeled *Random* in the graphs, to compare against EvenCover. In the *Random* assignment, each node decides to go active with probability z, the desired active node density. The EvenCover algorithm does not use the concept of *rounds* while the other algorithms do. Therefore, in order to present a fair comparison, we use a slightly modified version of our algorithm so that each node is scheduled to work in rounds. At the beginning of each round, each node randomly picks a start time between 0 and T, and makes its decision based on the information received so far. If the node chooses to turn on, then it will broadcast its decision to its neighbors. Each of its neighbors will receive the decision and remember it for its own reference. If the node chooses to be turned off, then it will not make any broadcast attempts. A node is considered dead if it has consumed all the power allocated and alive, otherwise. Once the node is dead, it cannot be turned on again nor can it broadcast or receive signals. The power consumption model is adapted from [34], with the same assumption that each node is allocated 2J of energy and the data signal is a 2000-bit report message. The transmission energy consumption and the reception energy consumption is calculated in the following ways:

$$E_{T_x}(d) = E_{elec} + \varepsilon_{friss-amp} d^2 \tag{11}$$

$$E_{R_x} = E_{elec} \tag{12}$$

where $E_{T_x}(d)$ is the energy consumed in transmitting the signal to an area of radius d, E_{elec} is the energy consumed for the radio electronics, $\varepsilon_{friss-amp}$ is for the power amplifier and E_{R_x} is the energy consumed in receiving the signal. Radio parameters are set as $E_{elec} = 50nJ/bit$ and $\varepsilon_{friss-amp} = 10pJ/bit/m^2$. The energy consumed in data transmission is set to be thirty times that consumed in on/off broadcast transmission.

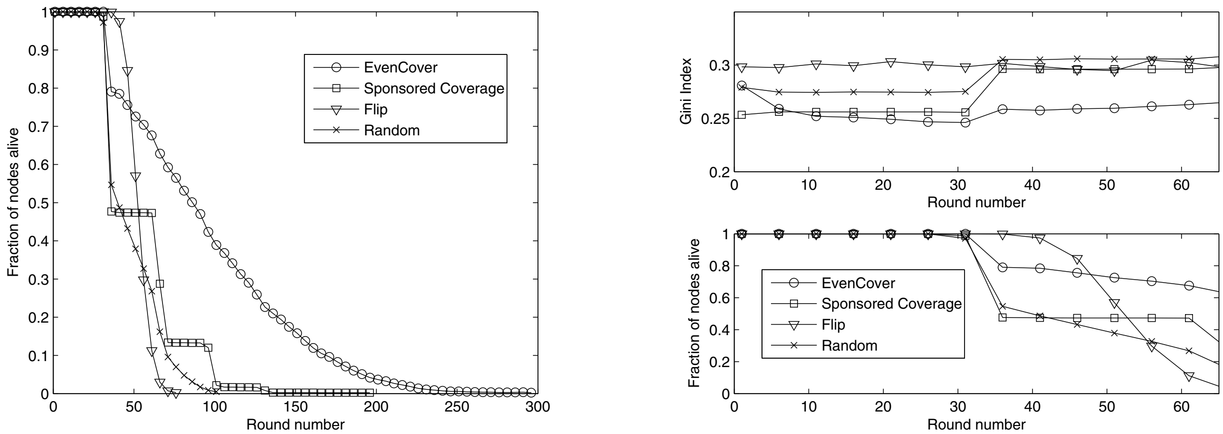
Fig. 5a reports the fraction of nodes alive as time progresses to indicate the network lifetime (for example, one may define network lifetime as the time until 50% of the nodes are dead). Figs. 5a and 5b demonstrate that the EvenCover algorithm significantly improves the lifetime of the network, degrades gracefully, and at the same time, achieves better uniformity of coverage even as nodes die. Since the Sponsored Coverage algorithm seeks complete coverage of all points in the region, some sensor nodes may have to constantly stay active while its neighbors are constantly in sleep mode. The goal of full coverage, therefore, contributes to a reduced lifetime while the goal of even coverage, appropriate for spot-sensing applications, results in an improved lifetime. The combination of striving for evenness and the use of random chance to place nodes in specific modes (as in lines 18 and 23 in Fig. 2) leads to the difference between the lifetime achieved by EvenCover and that achieved by other algorithms. In addition to improved lifetime, Fig. 5b shows that the EvenCover algorithm successfully preserves the evenness, as nodes in the network die, in comparison to other algorithms. This indicates that EvenCover achieves a more graceful degradation of the sensor network as the battery power in the nodes are exhausted.

V. Concluding Remarks

Coverage problems, in recent research literature, have largely focused on area-sensing applications. In this paper, we turn our attention to spot-sensing applications and introduce a new coverage problem with the goal of evenness in coverage. To the best of the knowledge of the authors, this is the first work that specifically targets evenness of coverage in sensor networks. We constructed a theoretical foundation, based on which we developed a distributed algorithm called EvenCover. Using simulation results, we have shown that EvenCover achieves very good evenness and a longer network lifetime than other distributed coverage algorithms.

Coverage algorithms designed for area-sensing applications use the sensing radius as the only input parameter to decide which nodes should go active and which should go to sleep. A given sensing radius implies a specific target density of

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.



(a) The fraction of nodes that are alive for each algorithm.

(b) The Gini index of the graph after each round in comparison with the fraction of nodes that are alive.

Fig. 5: Plots showing the network lifetime (the fraction of nodes that are alive is used as an indication of network lifetime) and the Gini Index achieved as the network nodes die. The communication radius used is $r = 0.07$.

active nodes, and vice versa. If the target density is low, it implies a large sensing radius and therefore, for most coverage algorithms, a large communication radius and high energy costs. Coverage algorithms designed for area-sensing applications, therefore, cannot be readily adapted for spot-sensing applications, especially at lower values of the desired density of active nodes. The EvenCover algorithm, however, works well for spot-sensing applications at all active node densities while also achieving a long lifetime. As a result, most importantly, the EvenCover algorithm allows a graceful degradation of the network as nodes die because it preserves evenness at all active node densities.

ACKNOWLEDGMENT

This work was supported in part by NSF Award CNS-0626548.

APPENDIX

Theorem 2: The lower bound on $U(G, R)$ is 0.2.

Proof: The lower bound of $U(G, R)$ is achieved when the sensing nodes are perfectly evenly distributed. Consider a distribution of nodes such that the region of interest is completely covered by circular areas of radius r with a sensing node at the center of each circle. Note that such a distribution of nodes is unrealizable but offers a way to derive a lower bound on $U(G, R)$, though not a tight one. Since each circle is identical to all others as far as distances of points to nearest nodes is concerned, $U(G, R)$ for the region within one circle is the $U(G, R)$ for the entire region. The average distance of a point from the center of the circle is:

$$\overline{d} = \int_0^r \frac{2\pi x}{\pi r^2} x dx = \frac{2r}{3} \tag{13}$$

Consider two random points within such a circle at distances x and y from the center of the circle.

$$E[|x-y|] = \int_0^r \frac{2\pi x}{\pi r^2} \int_0^r \frac{2\pi y}{\pi r^2} |x-y| dy dx = \frac{4r}{15} \tag{14}$$

Using Eqns. (13) and (14) in the definition of $U(G, R)$ given in Eqn. (1), we get:

$$U(G, R) = \left(\frac{1}{2}\right)\left(\frac{4r/15}{2r/3}\right) = \frac{1}{5}$$

■

Theorem 3: The expected value of $U(G, R)$ when the sensing nodes are distributed as per a Poisson process of rate z nodes per unit area is $1 - 1/\sqrt{2}$.

Proof: Let W denote a random variable indicating the distance of a random point from its nearest node. The probability density function of W is given by:

$$p_W(r) = 2\pi r e^{-\pi r^2 z}$$

Consider any two random points whose distances to their respective nearest nodes are x and y. Now,

$$\begin{aligned} E[|x-y|] &= \int_0^\infty p_W(x) \int_0^\infty p_W(y)|x-y|\, dy\, dx \\ &= \int_0^\infty p_W(x) \int_0^x p_W(y)(x-y)\, dy\, dx \\ &\quad + \int_0^\infty p_W(x) \int_x^\infty p_W(y)(y-x)\, dy\, dx \end{aligned}$$

Focusing first on the inner integrals and simplifying, we get the following two results:

$$\begin{aligned} \int_0^x p_W(y)(x-y)dy &= \int_0^x e^{-z\pi y^2} 2\pi y z(x-y) dy \\ &= x - \int_0^x e^{-z\pi y^2} dy \end{aligned} \tag{15}$$

$$\int_x^\infty p_W(y)(y-x)dy = \int_x^\infty e^{-z\pi y^2} 2\pi yz dy (y-x) dy = \int_x^\infty e^{-z\pi y^2} dy \quad (16)$$

Define $g(x)$ as follows:

$$g(x) = \int_0^x e^{-z\pi y^2} dy$$

Since $g(\infty) = \frac{1}{2}\sqrt{\frac{1}{z}}$,

$$\int_x^\infty e^{-z\pi y^2} dy = \frac{1}{2}\sqrt{\frac{1}{z}} - g(x)$$

Using (15) and (16), $E[|x-y|]$ may be expressed as:

$$\begin{aligned} & \int_0^\infty p_W(x)[x + \frac{1}{2}\sqrt{\frac{1}{z}} - 2g(x)]dx \\ \approx & \int_0^\infty e^{-z\pi x^2} 2\pi xz \left[x + \frac{1}{2}\sqrt{\frac{1}{z}} - 2g(x)\right] dx \\ = & \sqrt{\frac{1}{z}} + 2\int_0^\infty g(x) \frac{d}{dx}(e^{-z\pi x^2})\, dx \\ = & \sqrt{\frac{1}{z}} + 2e^{-z\pi x^2} g(x)|_0^\infty - \int_0^\infty 2e^{-z\pi x^2} \frac{d(g(x))}{dx}\, dx \end{aligned}$$

Simplifying further using routine calculus, we get:

$$E[|x-y|] = \sqrt{\frac{1}{z}}(1 - \sqrt{\frac{1}{2}}) \quad (17)$$

From Theorem 1, we know that the expected distance to the nearest node is $1/2\sqrt{z}$. Combining this with (17) in the definition of $U(G, R)$, we get:

$$U(G, R) = \frac{1}{2\frac{1}{2}\sqrt{\frac{1}{z}}}\sqrt{\frac{1}{z}}(1 - \sqrt{\frac{1}{2}}) = 1 - \sqrt{\frac{1}{2}}$$

■

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[2] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings of INFOCOM 2001*. IEEE, 2001, pp. 1380–1387.

[3] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 28–39.

[4] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mob. Netw. Appl.*, vol. 10, no. 4, pp. 519–528, 2005.

[5] X. Bai, D. Xuan, Z. Yun, T. H. Lai, and W. Jia, "Complete optimal deployment patterns for full-coverage and k-connectivity (k≤6) wireless sensor networks," in *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2008, pp. 401–410.

[6] B. Cărbunar, A. Grama, J. Vitek, and O. Cărbunar, "Redundancy and coverage detection in sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 94–128, 2006.

[7] R.-S. Chang and S.-H. Wang, "Deploying sensors for maximum coverage in sensor networks," in *IWCMC '07: Proceedings of the 2007 international conference on Wireless communications and mobile computing*. New York, NY, USA: ACM, 2007, pp. 220–224.

[8] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in wireless ad hoc sensor networks," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 753–763, June 2003.

[9] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized sensor area coverage with low communication overhead," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 661–672, May 2008.

[10] C.-F. Huang, L.-C. Lo, Y.-C. Tseng, and W.-T. Chen, "Decentralized energy-conserving and coverage-preserving protocols for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 182–187, 2006.

[11] C.-F. Huang, Y.-C. Tseng, and H.-L. Wu, "Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network," *ACM Trans. Sen. Netw.*, vol. 3, no. 1, p. 5, 2007.

[12] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: coverage, connectivity and diameter," in *Proceedings of INFOCOM 2003*. IEEE, 2003, pp. 1073–1083.

[13] Y.-C. Wang and Y.-C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.

[14] T. Yan, Y. Gu, T. He, and J. A. Stankovic, "Design and optimization of distributed sensing coverage in wireless sensor networks," *Trans. on Embedded Computing Sys.*, vol. 7, no. 3, pp. 1–40, 2008.

[15] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley and Sons, Ltd., 2006.

[16] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," in *IEEE INFOCOM*, 2003, pp. San Francisco, CA.

[17] K. Krizman, T. E. Biedka, and T. S. Rappaport, "Wireless position location: Fundamentals, implementation strategies, and source of error," in *Proc. IEEE Vehicular Technology Conf.*, 1997, pp. 919–923.

[18] B. Raman and K. Chebrolu, "Censor networks: A critique of "sensor networks" from a systems perspective," *Computer Communication Review*, vol. 38, no. 3, pp. 75–81, July 2008.

[19] P. J. Clark and F. C. Evans, "Distance to nearest neighbor as a measure of spatial relationships in populations," *Ecology*, vol. 35, no. 4, pp. 445–453, Oct. 1954.

[20] F. A. Cowell, *Measuring inequality: techniques for the social sciences*. John Wiley & Sons, New York, 1977.

[21] A. W. Marshall and I. Olkin, *Inequalities: theory of majorization and its applications*. Academic Press, New York, 1979.

[22] D. B. West, *Introduction to Graph Theory*. Prentice Hall, 2001.

[23] A. J. O'Donnell and H. Sethu, "On achieving software diversity for improved net- work security using distributed coloring algorithms," in *Proc. ACM Conf. on Comp. and Commun. Security*, 2004, pp. 121–131.

[24] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag, 2001.

[25] Z. Drezner and H. W. H. (eds.), *Facility Location: Applications and Theory*. Springer, 2004.

[26] S. H. Owen and M. S. Daskin, "Strategic facility location: A review," *European J. of Oper. Res.*, vol. 111, no. 3, pp. 423–447, Dec. 1998.

[27] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the placement of web server replicas," in *IEEE INFOCOM 2001*, 2001, pp. 1587–1596.

[28] B. Li, M. J. Golin, G. F. Ialiano, and X. Deng, "On the optimal placement of web proxies in the internet," in *IEEE INFOCOM*, 1999.

[29] S. Kim, M. Yoon, and Y. Shin, "Placement algorithm of web server replicas," in *Lecture Notes in Comp. Sci.*, vol. 3043, 2004, pp. 328–336.

[30] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," *Computer Communication*, vol. 25, no. 4, pp. 376–383, Mar. 2002.

[31] B.-J. Ko and D. Rubenstein, "Distributed self-stabilizing placement of replicated resources in emerging networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 476–487, Jun. 2005.

[32] H. Sethu and T. Gerety, "A distributed topology control algorithm in the presence of multipath propagation," 2007. [Online]. Available: http://arxiv.org/abs/0709.0961

[33] N. Li and J. C. Hou, "Localized topology control algorithms for heterogeneous wireless networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1313–1324, December 2005.

[34] D. Tian and N. D. Georganas, "A node scheduling scheme for energy conservation in large wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 271–290, 2004.