Ad Hoc & Sensor Wireless Networks, Vol. 1, pp. 89–124
Reprints available directly from the publisher
Photocopying permitted by license only

Maintaining Sensing Coverage and Connectivity in Large Sensor Networks

Honghai Zhang and Jennifer C. Hou

Department of Computer Science,
University of Illinois at Urbana Champaign, Urbana, IL 61801, USA.
E-mail contact: jhou@cs.uiuc.edu

In this paper, we address the issues of maintaining sensing coverage and connectivity by keeping a minimum number of sensor nodes in the active mode in wireless sensor networks. We investigate the relationship between coverage and connectivity by solving the following two sub-problems. First, we prove that if the radio range is at least twice the sensing range, complete coverage of a convex area implies connectivity among the working set of nodes. Second, we derive, under the ideal case in which node density is sufficiently high, a set of optimality conditions under which a subset of working sensor nodes can be chosen for complete coverage.

Based on the optimality conditions, we then devise a decentralized density control algorithm, *Optimal Geographical Density Control* (OGDC), for density control in large scale sensor networks. The OGDC algorithm is fully localized and can maintain coverage as well as connectivity, regardless of the relationship between the radio range and the sensing range. *Ns-2* simulations show that OGDC outperforms existing density control algorithms [25, 26, 29] with respect to the number of working nodes needed and network lifetime (with up to 50% improvement), and achieves almost the same coverage as the algorithm with the best result.

1 INTRODUCTION

Recent technological advances have led to the emergence of pervasive networks of small, low-power devices that integrate sensors and actuators with limited on-board processing and wireless communication capabilities. These sensor networks open new vistas for many potential applications, such as battlefield surveillance, environment monitoring and biological detection [2,10,13,17].

Since most of the low-power devices have limited battery life and replacing batteries on tens of thousands of these devices is infeasible, it is well accepted that a sensor network should be deployed with high density (up to 20 nodes/m^3 [23]) in order to prolong the network lifetime. In such a high-density network with energy-constrained sensors, if all the sensor nodes operate in the active mode, an excessive amount of energy will be wasted, sensor data collected is likely to be highly correlated and redundant, and moreover, excessive packet collision may occur as a result of sensors intending to send packets simultaneously in the presence of certain triggering events. Hence it is neither necessary nor desirable to have all nodes simultaneously operate in the active mode.

One important issue that arises in such high-density sensor networks is *density control* — the function that controls the density of the working sensors to certain level [29]. Specifically, density control ensures only a subset of sensor nodes operates in the active mode, while fulfilling the following two requirements: (i) *coverage:* the area that can be monitored is not smaller than that which can be monitored by a full set of sensors; and (ii) *connectivity:* the sensor network remains connected so that the information collected by sensor nodes can be relayed back to data sinks or controllers. Under the assumption that an (acoustic or light) signal can be detected with certain minimum signal to noise ratio by a sensor node if the sensor is within a certain range of the signal source, the first issue essentially boils down to a coverage problem: assuming that each node can monitor a disk (the radius of which is called the *sensing range* of the sensor node) centered at the node on a two dimensional surface, what is the minimum set of nodes that can cover the entire area? Moreover, if the relationship between coverage and connectivity can be well characterized (e.g., under what condition coverage may imply connectivity and vice versa), the connectivity issue can be studied, in conjunction with the first. In addition to the above two requirements, it is desirable to choose a minimum set of working sensors in order to reduce power consumption and prolong network lifetime. Finally, due to the distributed nature of sensor networks, a practical density control algorithm should be not only distributed but also completely localized (i.e., relies on and makes use of local information only) [10].

In this paper, we address the issue of density control in an analytic framework, and based on the findings, propose a fully decentralized and localized algorithm, called *Optimal Geographical Density Control* (OGDC), in large scale sensor networks. Our goal is to maintain coverage as well as connectivity using a minimum number of sensor nodes. We investigate the relationship between coverage and connectivity by solving the following two sub-problems. First, we prove that under the assumption **(A1)** the radio range is at least twice the sensing range, a complete coverage of a convex area implies connectivity among the set of working nodes. Note that as indicated in Tables 2 and 3, **(A1)** holds for a wide spectrum of sensor devices that recently emerge. As a result, the proof allows us to focus only on the coverage problem, as complete coverage implies connectivity. Second, we explore, under the ideal case that the node density is sufficiently high, a set of optimality conditions under which a subset of working nodes can be chosen for complete coverage. Based on the optimality conditions, we then devise a decentralized and localized density control algorithm, OGDC. We also discuss the procedures taken by OGDC in the (infrequent) case that the radio range is smaller than twice the sensing range, thus allowing OGDC to be uniformly applied to all cases. We also perform *ns-2* simulations to validate OGDC and compare it against a hexagon-based GAF-like algorithm, the PEAS algorithm presented in [29] and the CCP protocol in [26].

Several researchers have addressed the same or similar issues, with the work reported in [11,25,26,28,29] coming closest to ours. (We will provide a detailed summary of existing work in Section 5.) However, the work reported in [28,29] does not ensure complete coverage. Although the work reported in [25] does attempt to solve the complete coverage problem, it requires a large number of nodes to operate in the active mode (even more than a simple algorithm based on the idea of GAF does [27]). On the other hand, the work in [11] assumes error-free channels and requires reliable broadcasting in a certain range, which is hard to implement in wireless environments. The very recent work by Wang *et al.* [26] contains a similar analysis on the relationship between coverage and connectivity, but does not derive optimal conditions for minimizing the number of working nodes as we do in this paper.

The rest of the paper is organized as follows. In Section 2 we investigate the relationship between coverage and connectivity. In Section 3 we derive the optimality conditions for complete coverage under the ideal case. Following that, we present in Section 4 the proposed density control algorithm, and give in Section 5 a detailed summary of existing works. Finally, we present our simulation study in Section 6 and conclude the paper in Section 7.

2 RELATIONSHIP BETWEEN COVERAGE AND CONNECTIVITY

In this section we investigate the relationship between coverage and connectivity. Specifically, we derive the necessary and sufficient condition under which coverage implies connectivity — the radio range is at least twice the sensing range. We assume the entire area is a convex set, and

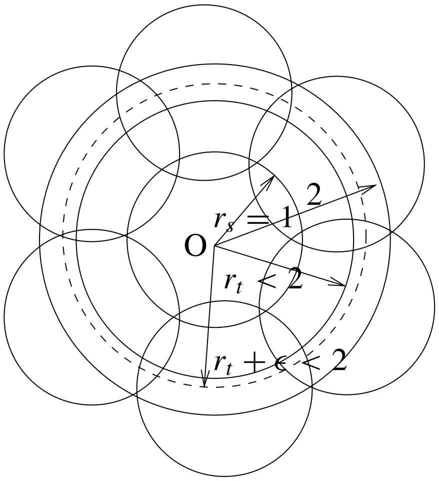


FIGURE 1
A scenario that demonstrates $r_t \geq 2r_s$ is a necessary condition that complete coverage ensures connectivity.

denote the sensing range and the radio transmission range as, respectively, r_s and r_t.

Lemma 1 *Assuming the monitored region is a convex set, the condition of* $r_t \geq 2r_s$ *is both necessary and sufficient to ensure that complete coverage of a convex region implies connectivity in an arbitrary network.*

Proof. We prove the necessary condition by devising a scenario in which coverage does not imply connectivity if $r_t < 2r_s$. In Figure 1, a sensor is located at O and has, respectively, a sensing radius r_s and a radio transmission radius $r_t < 2r_s$. Now we place a sufficient number of sensors on the circle centered at O with radius $r_t + \epsilon < 2r_s$ (where $\epsilon > 0$) such that they together cover the whole disk centered at O and with radius $r_t + \epsilon$. However, this network is not connected since the distance between node O and any other node is more than r_t.

Next we show that $r_t \geq 2r_s$ is also a sufficient condition to ensure that coverage implies connectivity. We prove this by contradiction. If a network is disconnected, there exists a pair of nodes between which no path exists. Let (S, D) be a pair of nodes with the minimum distance among all pairs of disconnected nodes (Fig. 2). Considering the circle whose center is on the line from node S to node D and the distance between its center and node S is r_s, we claim that there must exist some other node within or on the circle. Otherwise, since the number of nodes is finite in any finite area, we can move the circle along $\overline{SD}$ toward node D by a minimum distance ϵ in order to make the circle include another node. If we move the circle

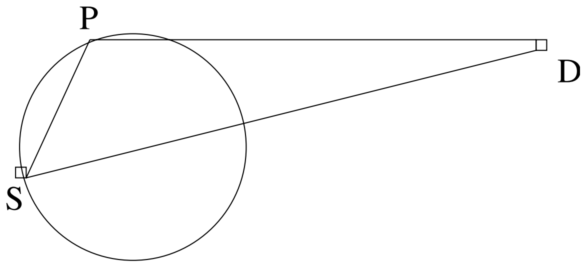


FIGURE 2
A scenario that demonstrates $r_t \geq 2r_s$ is the sufficient condition that complete coverage implies connectivity.

along $\overline{SD}$ toward node D by a distance $\epsilon/2$, there will be no node within or on the circle. That means the center of the circle is not covered by any node, which violates the condition of coverage. Let node P be such a node that lies within or on the circle (before it is moved). Nodes S and P are connected since their distance is less than $2r_s \leq r_t$. Hence nodes P and D must be disconnected; otherwise nodes S and D are connected. Since $\angle SPD > \pi/2 > \angle PSD$, we have $|\overline{SD}| > |\overline{PD}|$. This contradicts the assumption that nodes S and D have the minimum distance among all the pairs of disconnected nodes. □

Although the above derivation is made on a two dimensional surface, both the lemma and its proof apply to three dimensional space as well. An important implication of Lemma 1 is that if the radio range is at least twice the sensing range (which holds for a wide variety of applications), then complete coverage implies connectivity. That is, the problem of ensuring both coverage and connectivity can be reduced to that of ensuring coverage only. We will henceforth only consider the coverage problem in the analytical framework. Later in the course of designing our decentralized, localized algorithm, OGDC, we will consider the "extra" procedure taken to deal with the (rare) case that the radio ranges are smaller than twice the sensing ranges.

It is proved in [26] that if $r_t \geq 2r_s$, k-coverage implies k-connectivity of the entire network and $2k$-connectivity of the interior network on a convex area. However, we emphasize here that the condition $r_t \geq 2r_s$ is also necessary in the sense that if $r_t < 2r_s$, coverage does not imply connectivity in general.

3 OPTIMAL SENSING COVERAGE IN THE IDEAL CASE

Recall that two requirements are implied in density control: first, the subset should completely cover the region R. Specifically, given that the coverage area of a sensor node is a disk centered at itself, we define a *crossing* as an intersection point of two circles (boundaries of disks) or that of a circle and the boundary of region R. A crossing is said to be *covered* if it is an *interior point* of a third disk. The following lemma from [12] pages 59 and 181 provides a sufficient condition for complete coverage. This condition is also necessary if we assume that the circle boundaries of any three disks do not intersect at a point. The assumption is reasonable as the probability of the circle boundaries of three disks intersecting at a point is zero, if all sensors are randomly placed in a region with uniform distribution. Lemma 2 serves as an important theoretical base for our distributed density control algorithm in the next section.

Lemma 2 *Suppose the size of a disk is sufficiently smaller than that of a convex region R. If one or more disks are placed within the region R, and at least one of those disks intersect another disk, and all crossings in the region R are covered, then R is completely covered.*

The second requirement is that the set of working sensors should consume as little power as possible so as to prolong the network lifetime. If each sensor consumes the same amount of power when it is active and has the same sensing range, the requirement of minimizing power consumption boils down to that of minimizing the number of working sensors. On the other hand, if sensors have different sensing ranges (e.g., using different levels of power to sense), a minimum number of working sensors does not necessarily imply minimum power consumption.

To derive conditions under which the second requirement is fulfilled, we first define the *overlap* at a point x as the number of sensors whose sensing ranges can cover the point minus $I_R(x)$, where

$$I_R(x) = \begin{cases} 1 & \text{if } x \in R, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The overlap of sensing areas of all the sensors is then the integral of overlaps of the points over the area covered by all the sensors. In general, the larger the overlap of the sensing areas, the more amount of redundant data will be generated and more power will be consumed. On the other hand, an adequate degree of redundancy may be needed to gather accurate, high-fidelity data in some cases. Although our focus in this paper is to ensure that every point is covered by at least one sensor, we will discuss how to extend our work to ensure k-coverage (i.e., every point is covered by at least k sensors) in Section 6.

We claim that overlap is a better index for measuring power consumption than the number of working sensors for two reasons. First, although the number of working sensors is not directly related to power consumption in the case that sensors have different sensing ranges, the measure of overlap still is, i.e., a larger value of overlap implies more data redundancy and power consumption. Second, as will be proved in the following lemma, minimizing the overlap value is equivalent to minimizing the number of working sensors in the case that all sensors have the same sensing ranges (i.e., the coverage disks of all sensors have the same radius r).

Lemma 3 *If all sensor nodes (i) completely cover a region R and (ii) have the same sensing range, then minimizing the number of working nodes is equivalent to minimizing the overlap of sensing areas of all the working nodes.*

Proof. See Appendix 1.

Lemma 3 is important as it relates the total number of working sensor nodes to the overlapping areas between working nodes. Since the latter is more easily measured from a local point of view, this greatly simplifies the task of designing a decentralized and localized density control algorithm, which will become clear later.

3.1 Properties under the Ideal Case

With Lemmas 2–3, we are now in a position to discuss how to minimize the overlap of sensing areas of all the sensor nodes. Our discussion is built upon the following assumptions:

(**A1**) The sensor density is high enough that a sensor can be found at any desirable point.

(**A2**) The region R is large enough as compared to the sensing range of each sensor node so that the boundary effects can be ignored.

Assumption (**A2**) is usually valid. Although (**A1**) may not hold in practice, as will be shown in Section 4, the result derived under (**A1**) still provides insightful guidance in designing the distributed algorithm.

By Lemma 2, in order to totally cover the region R, some sensors must be placed inside region R and their coverage areas intersect one another. If two disks A and B intersect, at least one more disk is needed to cover their crossing points. Consider, for example, in Figure 3, disk C is used to cover the crossing point O of disks A and B. In order to minimize the overlap while covering the crossing point O (and its vicinity not covered by disks A and B), disk C should also intersect disks A and B at the point O; otherwise, one can always move disk C away from disks A and B to reduce the overlap.

Given that two disks A and B intersect, we now investigate the number of disks needed, and their relative locations, in order to cover a crossing point O of disks A and B and at the same time minimize the overlap. Take the case of three disks (Fig. 3) as an example. Let $\angle PAO = \angle PBO \overset{\triangle}{=} \alpha_1$,

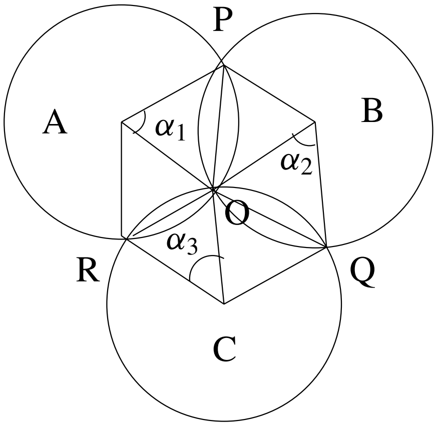


FIGURE 3
An example that demonstrates how to minimize the overlap while covering the crossing point O.

$\angle OBQ = \angle OCQ \overset{\triangle}{=} \alpha_2$, and $\angle OCR = \angle OAR \overset{\triangle}{=} \alpha_3$. We consider two cases: (i) $\alpha_1, \alpha_2, \alpha_3$ are all variables; and (ii) α_1 is a constant but α_2 and α_3 are variables. Case (i) corresponds to the case where we can choose all the node locations, while case (ii) corresponds to the case where two nodes (A and B) are already fixed and we need to choose the position of a third node C to minimize the overlap. Both of the above two cases can be extended to the general situation in which $k-2$ additional disks are placed to cover one crossing point of the first two disks (that are placed on a two-dimensional plane), and α_i, $1 \le i \le k$, can be defined accordingly. Again, the boundaries of all disks should intersect at point O in order to reduce the overlap. In the following discussion we assume for simplicity that the sensing range $r = 1$. Note, however, that the results still hold when $r \neq 1$.

Case 1: α_i**,** $1 \le i \le k$**, are all variables.** We first prove the following Lemma.

Lemma 4

$$\sum_{i=1}^{k} \alpha_i = (k-2)\pi, \tag{2}$$

Proof. See Appendix 2.

Now the overlap between the i^{th} and $(i \bmod k) + 1^{th}$ disks (which are called *adjacent disks*) is $(\alpha_i - \sin\alpha_i)$, $1 \le i \le k$. If we ignore the overlap caused by *non-adjacent* disks, then the total overlap is $L = \sum_{i=1}^{k}(\alpha_i - \sin\alpha_i)$. The coverage problem can be formulated as

Problem 1

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{k}(\alpha_i - \sin\alpha_i) \\ &\text{subject to} \sum_{i=1}^{k}\alpha_i = (k-2)\pi. \end{aligned} \tag{3}$$

The Lagrangian multiplier method can be used to solve the above optimization problem. The solution is $\alpha_i = (k-2)\pi/k, i = 1, 2, \cdots, k$ and the resulting minimum overlap using k disks to cover the crossing point O is

$$L(k) = (k-2)\pi - k\sin(\frac{(k-2)\pi}{k}) = (k-2)\pi - k\sin(\frac{2\pi}{k}).$$

Note that the overlap per disk

$$\frac{L(k)}{k} = \pi - \frac{2\pi}{k} - sin(\frac{2\pi}{k}) \tag{4}$$

monotonically increases with k when $k \geq 3$. Moreover when $k = 3$ (which means that we use one disk to cover the crossing point), the optimal solution is $\alpha_i = \pi/3$ and there is no overlap between *non-adjacent* disks. When $k > 3$, the overlap per disk is always higher than that in the case of $k = 3$, even if we ignore the overlaps between *non-adjacent* disks. This implies that using one disk to cover the crossing point and its vicinity is optimal in the sense of minimizing the overlap. Moreover, the centers of the three disks should form a equilateral triangle with edge $\sqrt{3}$. We state the above result in the following theorem.

Theorem 1 *To cover one crossing point of two disks with the minimum overlap, only one disk should be used and the centers of the three disk should form a equilateral triangle with side length $\sqrt{3}r$, where r is the radius of the disks.*

Case 2: α_1 is a constant, while α_i, $2 \leq i \leq k$, are variables. In this case the problem can still be formulated as in Problem 1, except that α_1 is fixed. The Lagrangian multiplier method can again be used to solve the problem, and the optimal solution is $\alpha_i = ((k-2)\pi - \alpha_1)/(k-1)$, $2 \leq i \leq k$. Again a similar conclusion can be drawn that using one disk to cover the crossing point gives the minimum overlap. We state the result in the following theorem.

Theorem 2 *To cover one crossing point of two disks whose locations are fixed (i.e., α_1 is fixed in Fig. 3), only one disk should be used and $\alpha_2 = \alpha_3 = (\pi - \alpha_1)/2$.*

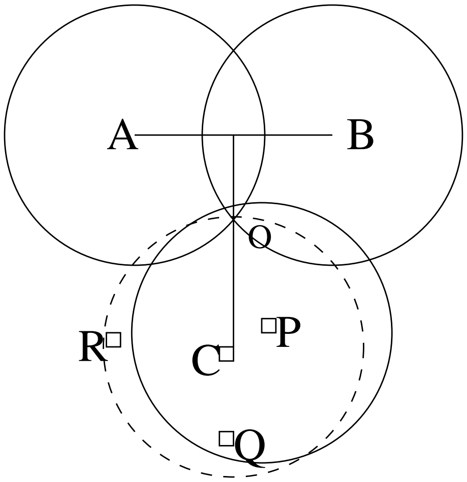


FIGURE 4
Although C is the optimal place to cover the crossing O of A, B, there is no sensor node there. The node closest to C, P, is selected to cover the crossing O.

In summary, to cover a large region R with the minimum overlap, one should ensure (i) at least one pair of disks intersects; (ii) the crossing points of any pair of disks are covered by a third disk; (iii) if the locations of any three sensor nodes are adjustable, then as stated in Theorem 1 the three nodes should form an equilateral triangle with side length $\sqrt{3}r$. If the locations of two sensor nodes A and B are already fixed, then as stated in Theorem 2 the third sensor node should be placed on the line that is perpendicular to the line connecting nodes A and B and have a distance r to the intersection of the two circles (e.g., the optimal point in Fig. 4 is C). These conditions are optimal for the coverage problem in the ideal case in which assumptions (**A1**) and (**A2**) hold.

As mentioned above, the notion of overlap can be extended to the heterogeneous case in which sensors have different sensing ranges. Moreover, Theorem 1 and 2 can be generalized to the heterogeneous case. For ease of discussion, we consider the case of using only one extra disk to cover the crossing point O.

Theorem 3 *Assuming that different nodes have different sensing ranges, to cover one crossing point O of two disks with the minimum overlap, the three disks should be placed such that* $\overline{OP} = \overline{OQ} = \overline{OR}$. *If disk A and B are already fixed, disk C should be placed such that* $\overline{OR} = \overline{OQ}$.

Proof. We only prove the first part of the theorem where the location of all three disks can change. To prove the second part when node A and B are fixed we only need to take the variable x_1 below as a fixed value.

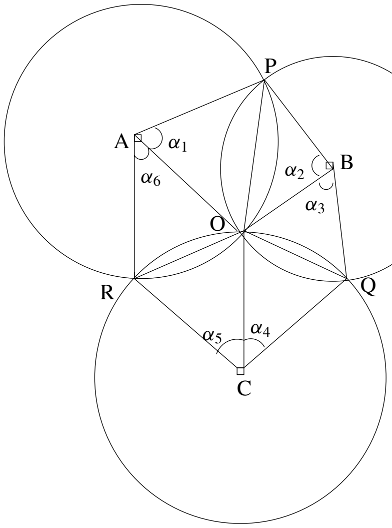


FIGURE 5
Minimizing the overlap while covering the crossing point O when each node has different sensing range.

Refer to Fig. 5. Let r_1, r_2 and r_3 denote the radii of disks A, B, and C, let $x_1 = \overline{OP}/2, x_2 = \overline{OQ}/2, x_3 = \overline{OR}/2$, and let $\alpha_1 = \angle OAP, \alpha_2 = \angle OBP, \alpha_3 = \angle OBQ, \alpha_4 = \angle OCQ, \alpha_5 = \angle OCR, \alpha_6 = \angle OAR$.Notice that if $r_1 = r_2 = r_3$, then $\alpha_1 = \alpha_2, \alpha_3 = \alpha_4, \alpha_5 = \alpha_6$. The angles $\alpha_i, 1 \leq i \leq 6$, can be expressed as

$$\begin{aligned}
&\alpha_1 = 2\arcsin(x_1/r_1), \alpha_2 = 2\arcsin(x_1/r_2),\\
&\alpha_3 = 2\arcsin(x_2/r_2), \alpha_4 = 2\arcsin(x_2/r_3),\\
&\alpha_5 = 2\arcsin(x_3/r_3),\ \alpha_6 = 2\arcsin(x_3/r_1).
\end{aligned} \tag{5}$$

and the total overlap can be written as

$$\frac{1}{2}\sum_{i=1}^{6} r_i^2(\alpha_i - \sin\alpha_i). \tag{6}$$

Now the problem is to minimize Eq. (6) subject to the same constraint as in Lemma 4:

$$\sum_{i=1}^{6} \alpha_i = 2\pi. \tag{7}$$

Now we apply Lagrangian multiplier theorem with the Lagrangian function

$$L = \frac{1}{2}\sum_{i=1}^{6}(r_i^2(\alpha_i - \sin\alpha_i) + \lambda(\sum_{i=1}^{6}\alpha_i - 2\pi). \tag{8}$$

Note that the variables α_i's are not independent, e.g., both α_1 and α_2 depend on x_1. Hence we have to apply the Lagrangian multiplier theorem on the independent variables x_i's and regard α_i as $\alpha_i(x_j)$ where x_j is one of the x_k's that α_i depends on. First we apply the first order necessary condition on x_1.

$$\begin{aligned}\frac{\partial L}{\partial x_1} &= \sum_{i=1}^{6}\frac{\partial L}{\partial \alpha_i}\cdot\frac{\partial \alpha_i}{\partial x_1}\\ &= (2x_1^2 + \lambda)\left(\frac{1}{\sqrt{r_1^2 - x_1^2}} + \frac{1}{\sqrt{r_2^2 - x_1^2}}\right) = 0\end{aligned} \tag{9}$$

If $x^* = (x_1^*, x_2^*, x_3^*)$ and λ^* satisfy the first order Lagrangian necessary condition, we have $2x_1^{*2} = -\lambda^*$. Applying the same necessary condition on x_2 and x_3 renders $2x_2^{*2} = 2x_3^{*2} = -\lambda^*$. Thus $x_1^* = x_2^* = x_3^*$ satisfies the first order necessary conditions. To show it also satisfies the second order sufficient conditions, it suffices to verify that

$$\frac{\partial L^2(x^*, \lambda^*)}{\partial x_i \partial x_j} = 0 \text{ for } i \neq j, \tag{10}$$

and

$$\frac{\partial L^2(x^*, \lambda^*)}{\partial x_i^2} > 0 \text{ for all } i \tag{11}$$

to show the Hessian matrix of the Lagrangian is positive definite. That is, (x_1^*, x_2^*, x_3^*) is a local minimum. Since there is only one local minimum, it is also a global minimum. Hence (x_1^*, x_2^*, x_3^*) minimizes the Eq. (6) subject to constraint Eq. (7), and $\overline{OP} = \overline{OQ} = \overline{OR}$ minimizes the overlap. □

4 OPTIMAL GEOGRAPHICAL DENSITY CONTROL ALGORITHM

In this section, we propose a completely localized density control algorithm, called OGDC, that makes use of the optimal conditions derived in Section 3. Note that as it may not be possible to locate sensor nodes in any desirable position (i.e., assumption (**A1**) may not hold), OGDC attempts to select as working nodes the sensor nodes that are closest to the optimal locations. We first give an overview of OGDC and then delve into the detailed operations. We also discuss its possible extension and some limitations.

4.1 Overview

OGDC is devised under the following assumptions:

(**B1**) Each node is aware of its own position. This assumption is not impractical, as many research efforts have been made to address the localization problem [9,18,21].

(**B2**) For clarity of algorithm discussion, we assume the radio range is at least twice the sensing range, and will relax this assumption in Section 4.3.

(**B3**) For clarity of algorithm discussion, we assume all sensor nodes are time synchronized, and will relax this assumption in Section 4.4.

At any time, a node is in one of the three states: "UNDECIDED," "ON," and "OFF." Time is divided into rounds. Each round has two phases: the *node selection phase* and the *steady state phase*. At the beginning of the node selection phase, all the nodes wake up, set their states to "UNDECIDED," and carry out the operation of selecting working nodes. By the end of this phase, all the nodes change their states to either "ON" or "OFF". In the steady state phase, all nodes keep their states fixed until the beginning of the next round. The length of each round is so chosen that it is much larger than that of the node selection phase but much smaller than the average sensor lifetime. Our simulation results show that the time it takes to execute the node selection operation for networks of size up to 1000 nodes in an area of $50 \times 50\text{m}^2$ (with timer values appropriately set) is usually well below 1 second and most nodes can decide their states (either "ON" or "OFF") in less than 0.2 second from the time instant when at least one node volunteers to be a starting node. The interval for each round is usually set to approximately hundreds of seconds, and the overhead of density control is small ($\overset{<}{\sim}$ 1%).

The node selection phase in each round commences when one or more sensor nodes volunteer to be starting nodes. For example, suppose node A volunteers to be a starting node in Fig. 4. Then one of its neighbors with an (approximate) distance of $\sqrt{3}r$, say node B, will be "selected" to be a working node. To cover the crossing point of disks A and B, the node whose position is closest to the optimal position C (e.g., node P in Fig. 4) will then be selected, in compliance with Theorem 2, to become a working node. The process continues until all the nodes change their states to either "ON" or "OFF," and the set of nodes with state "ON" forms the working set. As a node probabilistically volunteers itself to be a starting node (with a probability that is related to its remaining power) in each round, the set of working sensor nodes is not likely to be the same in each round, thus ensuring uniform (and minimum) power consumption across the network, as well as complete coverage and connectivity. In what follows, we give the detailed description of OGDC.

4.2 Detailed Description of OGDC

Selection of the starting node. At the beginning of node election phase, every node is powered on with the "UNDECIDED" state. A node volunteers to be a starting node with probability p if its power exceeds a pre-determined threshold P_t. The power threshold P_t is related to the length of the round and in general is set to a value so as to ensure with high probability the sensor can remain powered on until the end of the round.

If a sensor node volunteers, it sets a backoff timer of τ_1 seconds, where τ_1 is uniformly distributed in $[0, T_d]$. When the timer expires, the node changes its state to "ON", and broadcasts a power-on message. If a node hears other power-on messages before its timer expires, it cancels its timer and does not become a starting node. The power-on message sent by the starting node contains (i) the position of the sender and (ii) the direction α along which the second working node should be located. This direction is randomly generated from a uniform distribution in $[0, 2\pi]$. Non-starting node may also send power-on message. In this case, the direction field in the power-on message is set to -1 to indicate the sender is a non-starting node.

The use of backoff timers avoids the possibility of multiple neighboring nodes volunteering themselves to be the starting nodes in a round. The selection of T_d is a tradeoff between the performance and the latency. Using a large value of T_d can reduce the number of starting nodes in the network and possibly reduce the level of overlap. However, with fewer starting nodes, it will take a longer time to complete the operations of selecting working nodes. In our simulation, we select T_d to be about 1.5 times of the transmission time of a power-on packet.

If the node does not volunteer itself to be a starting node, it sets a timer of T_s seconds. When the timer T_s[1] expires, it repeats the above volunteering process with p doubled until its value reaches 1. The timer is canceled whenever the state of a node is changed to "ON" or "OFF" in response to other power-on messages. T_s should be set to a sufficiently large value such that if there exists at least one node whose power level qualifies it to be a starting node, the operation of selecting working nodes can be completed in an early stage of each round. The value of p is initially set to p_0. We will discuss how to determine the value of p_0 in Section 4.4.

[1] With a little abuse of symbols, we will use T_s to refer both the timer and the value of the timer. This applies to other timers.

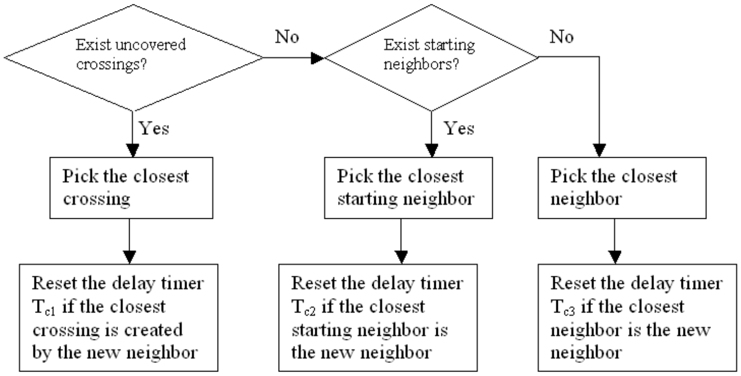

Exist uncovered crossings?
No
Exist starting neighbors?
No
Yes
Yes
Pick the closest crossing
Pick the closest starting neighbor
Pick the closest neighbor
Reset the delay timer T$_{c1}$ if the closest crossing is created by the new neighbor
Reset the delay timer T$_{c2}$ if the closest starting neighbor is the new neighbor
Reset the delay timer T$_{c3}$ if the closest neighbor is the new neighbor

FIGURE 6
The procedure taken when a node receives a power-on message

Actions taken when a node receives a power-on message. When a sensor node receives a power-on message, if the node is already "ON", or it is more than $2r_s$ away from the sender node, it ignores the message; otherwise it adds this sender to its neighbor list, and checks whether or not all its neighbors' coverage disks completely cover its own coverage disk. If so, the node sets its state to "OFF" and turns itself off. Otherwise, it enters one of the following three cases (as depicted in Fig. 6): i) there exists uncovered crossing that is created by its working neighbors and falls in the node's coverage disk; ii) the condition in (i) is not satisfied and at least one neighbor is a starting node; iii) neither (i) nor (ii) satisfies. A node can determine if a neighbor is a starting-node from the direction field of the power-on message sent by that neighbor (a positive value indicates a starting node and vise versa).

In case (i), the node first finds the closest uncovered crossing that falls in its coverage disk. If the closest uncovered crossing is created by the new neighbor (that sends the latest power-on message to the node), the node will cancel existing timer (T_{c1}, T_{c2} or T_{c3}) (if any) and (re-)set a timer of value T_{c1}. Otherwise, the node retains the existing timer. The rationale behind how the value of T_{c1} is calculated is illustrated in Figure 7: let O denote the closest uncovered crossing point, A, B the two corresponding sender nodes, C the optimal location of a third sensor node used to cover the crossing point O, R the location of the receiver node, d the distance between the receiver node and the crossing point O, and $\Delta\alpha$ the angle between $\overline{OC}$ and $\overline{OR}$. The value of T_{c1} is set as

$$T_{c1} = t_0(c((r_s - d)^2 + (d\Delta\alpha)^2) + u), \tag{12}$$

where t_0 is the time it takes to send a power-on message, c is a constant

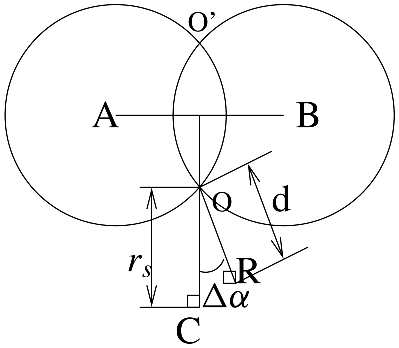


FIGURE 7
A scenario that demonstrates how the value T_{c1} is set (in case (i)).

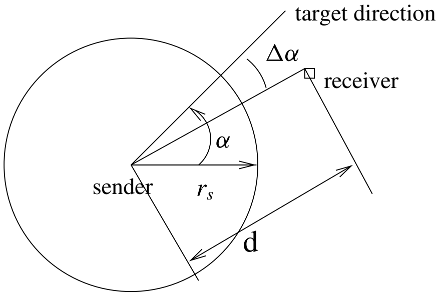


FIGURE 8
A scenario that demonstrates how the value T_{c2} is set (in case (ii)).

that determines the backoff scale and is set to $10/r_s^2$ in our simulation study, u is a random number drawn from the uniform distribution on [0, 1]. T_{c1} includes two terms: a deterministic term $c((r_s - d)^2 + (d\Delta\alpha)^2)$ and a random term (u). If the receiver is right in the direction α and its distance to the crossing is r_s, the deterministic term is 0; otherwise, $c((r_s - d)^2 + (d\Delta\alpha)^2)$ roughly represents the deviation from the optimal position and a delay is introduced in proportion of this deviation. The random term is introduced to break ties in the case that there exist nodes whose locations yield the same value of the deterministic term.

In case (ii), the node finds the closest starting neighbor. If the closest starting neighbor is the new neighbor, the node cancels the existing backoff timer (T_{c1}, T_{c2} or T_{c3}, if any) and (re-)sets a backoff timer of value T_{c2}. Otherwise, the node retains the existing timer. The value T_{c2} is set as (Fig. 8)

$$T_{c2} = t_0(c((\sqrt{3}r_s - d)^2 + (d\Delta\alpha)^2) + u), \tag{13}$$

where t_0, c, u are the same as those in Eq. (12), d is the distance from the sender to the receiver, $\Delta\alpha$ is the angle between α and the direction from the sender to the receiver.

In case (iii), the node finds the closest neighbor. If the closest neighbor is the new neighbor, the node cancels the existing backoff timer (T_{c1}, T_{c2} or T_{c3}, if any) and (re-)sets a backoff timer of value T_{c3}, which is much greater than that of the average values of T_{c1}, T_{c2} but much less than the value of T_s. Otherwise, it retains the existing timer. This is because when a node receives only power-on messages from non-starting neighbors, it expects to receive another power-on message and the coverage areas of the two senders will overlap.

In any of the above three cases, when the backoff timer expires, the node sets its state to "ON" and broadcasts a power-on message with the direction field α set to -1 (indicating a message generated by a non-starting node).

4.3 Extension to the case of insufficient transmission ranges

Now we extend OGDC to ensure both connectivity and coverage when the radio range is smaller than twice the sensing range. The only issue we need to address is to determine when a node should sleep. A sufficient condition that a node can sleep is that (**C1**) its coverage area is completely covered, and (**C2**) its working neighbors are all connected without it. It is difficult to test in a decentralized manner whether or not the second condition holds, because a node is only aware of its existing working neighbors (from whom it has received power-on messages). As a result we relax the second condition as "its *existing* working neighbors are all connected without it." If two neighbor nodes are within the transmission range of each other, they are necessarily connected. This can be determined by each node under assumption **(B1)**. Moreover, if a starting node propagates a power-on message (possibly via multiple hops) to two workings nodes, clearly they are connected. Hence, two existing working neighbors are connected if either (i) they are within the transmission range of each other, or (ii) they receive power-on messages originated from some common starting node.

Specifically, we associate each starting node with a unique id, called *netid*, and all nodes receiving power-on messages originated from the same starting node share its *netid*. A node may have multiple *netid*s (which are arranged into a *netid* list), if it receives power-on messages originated from more than one starting node. When a node decides to stay awake, it puts its *netid* list into the power-on message it sends. Each time a node A receives a power-on message from another node B, node A merges node B's *netid* list into its own. Moreover, each node divides its working neighbors into different groups based on their *netid*s. Specifically, each group initially contains working neighbors that share the same *netid*. When a node receives a power-on message, it will first update the groups as follows: if the message contains more than one *netid*, the node will merge all the groups which contain a *netid* in the list of the newly received messages. If the new neighbor is directly connected with another neighbor (as they are within the transmission range of each other) but with non-overlapping *netid* lists, the node will also merge all the groups which contain a *netid* in either of the two *netid* lists. At the end of the group-merging process, the node then decides if it can go to sleep: if there is only one group left and the node's coverage area is covered by the group, the node can go to sleep.

Efficient implementation of the *netid* list. Significant overhead may be incurred in power-on messages, if the *netid* list is long. Fortunately, a simple calculation shows that the probability that there are at most 3 starting nodes conditioning on that there exist starting nodes is more than 97%, if each node volunteers to be a starting node with the probability $1/N$, where N is the number of nodes in the sensor network. One way of efficiently implementing the *netid* list is as follows. A bitmap with the maximum size of k is used. To put a *netid* into the list, the "real *netid*" (which could be the starting node id) is first hashed into an integer j from 1 to k, and then the jth bit in the bitmap is set. The probability of having hash collision is small for reasonably large k. We choose $k = 8$ in our simulation.

4.4 Discussion

After describing the operations of OGDC, we are now in the position to elaborate on several implementation and parameter tuning issues:

Setting of the initial volunteering probability, p_0. Recall that p_0 is the initial probability that a node volunteers itself to be a starting node. In the case that the region to be covered is not large, it is desirable that at one time only one node determines to be a starting node. To this end, we set $p_0 = 1/N$, where N is the total number of sensor nodes in the network, as this maximizes the probability that exactly one sensor node volunteers itself as a starting node. On the other hand, if the region to be covered is large, it is desirable to have multiple sensor nodes volunteer themselves at one time. In this case, we set $p_0 = k/N$ as this maximizes the probability that exactly k nodes volunteer themselves. We argue that the number of sensor nodes, N, or at least its order is known at the time of network deployment. Even if this is not the case, as the value of p is doubled every time the T_s timer expires, the value of p_0 does not have a significant impact on the performance.

Guidelines of OGDC parameter tuning. OGDC has several tunable parameters. We have briefly described how to set the value of each parameter

TABLE 1
Parameter values used in the simulation study.

Parameter	Function	Value Used
r_s	sensing range	10 m
round time	period for executing OGDC	1000 s
P_t	power threshold for volunteering to be a working node	the level that allows a node to be idle for 900 seconds
T_d	maximum timer value used in volunteering to be a starting node	10 ms
T_s	maximum timer value used in re-initiating the process of volunteering to be a starting node	1 s
T_{c3}	timer value used when a node only receives power-on messages from non-starting neighbors and the coverage disks of those neighbors do not intersect in the node's coverage disk	200 ms
t_0	the time it takes to send a power-on packet	6.8 ms
c	constant used in Eqs. (12) and (13)	$10/r_s^2$
channel capacity		40K bps

when it is introduced for the first time. Now we outline the set of guidelines for parameter tuning. Table 1 lists the parameters, their functions, and their values used in our simulation study.

Most timing related parameters such as T_d, T_s and c should be set according to the transmission time of a power-on message t_0. As a rule of thumb, the T_d timer used to suppress surplus starting nodes should be in the same order of t_0. The T_s timer should be set to approximately two orders of magnitude larger than t_0 to allow the density control process to be completed before the T_s timer fires, if there exist some starting nodes in the network. T_{c3} should be chosen much larger than the average value of T_{c1}, T_{c2} and much smaller than T_s. The constant c should be chosen such that T_{c1} and T_{c2} are approximately one order of magnitude larger than t_0 on average to avoid packet collision. The round time should be set to a value that is approximately one order of magnitude less than that of the lifetime of a single sensor.

The value of P_t is dependent on the application requirement. If the application requires continuous, complete coverage, P_t should be set to a value such that a sensor can remain active for at least the duration of a round time. If intermittent, incomplete coverage in each round is acceptable, P_t can be set to a value that is less than the power required to keep the sensor active for the entire round time.

It is worth mentioning that we follow the above guidelines to tune parameters in our simulation and the simulation results are quite satisfactory.

Moreover, the performance of OGDC is not particularly susceptible to parameter settings as long as the above guidelines are followed.

Time synchronization. For simplicity of algorithm discussion, we have assumed that all nodes are time synchronized (assumption (**B3**)). This assumption can be relaxed as follows. In the first round we designate a sensor node to be the starting node. When the starting node sends a power-on message, it includes in its power-on message a duration δT after which the receivers should wake up for the next round. When a non-starting node broadcasts a power-on message, it reduces the value of δT by the time elapsed since it receives the last power-on message and includes the new value of δT in its power-on message. In this fashion, all the nodes get "synchronized" with the starting node and will all wake up at the beginning of the next round.

If the monitored region is so large that it is not acceptable to have one starting node in a round, we can synchronize a few nodes before deployment, distribute them evenly in the entire region, and designate them to be the starting nodes in the first round. Then we can similarly synchronize other nodes with the starting nodes in the first round as above. In fact it is not unreasonable to assume that multiple synchronized nodes with overlapping coverage areas can serve as reference points of other nodes ([4, 5]). To overcome the small clock drifting over the network lifetime, when a node wakes up, it needs to wait for a short time ($\geq$ the maximum clock drifting) before it starts to send any message.

What if no other sensor nodes volunteer. It may occur that the power of a node is less than the threshold power P_t and yet no power-on message is received even after the node sets the value of p to 1. This indicates that all the nodes do not have sufficient power and cannot volunteer themselves to be starting nodes. In this case, the node resets its power threshold P_t to 0 and restarts the density control process.

What if message loss occurs. If a packet sent from a neighbor is lost for any reason (transmission errors or collisions), a node is simply not aware of the existence of that neighbor. For example, if a starting node's power-on message is lost at all receivers (which occurs with a low probability), all other nodes will repeat the process of electing a starting node. As a result, the number of working nodes may increase. In spite of the performance degradation, OGDC is still robust in the sense that the algorithm is still operational and produces a set of working nodes to be powered on. If the sensor network is deployed in an environment where transmission error occur frequently, a node can be instrumented to send multiple power-on messages (with random delays) to increase the probability that its neighbor(s) receive the power-on message. This is a subject of future investigation.

5 RELATED WORK

Minimizing energy consumption and prolonging the system lifetime has been a major design objective for wireless ad hoc networks. GAF [27] assumes the availability of GPS and conserves energy by dividing a region into rectangular grids, ensuring that the maximum distance between any pair of nodes in adjacent grids is within the transmission range of each other, and electing a leader in each grid to stay awake and relay packets (while putting all the other nodes into sleep). The leader election scheme in each grid takes into account of battery usage at each node. SPAN [7], on the other hand, decides if a node should be working or sleeping based on connectivity among its neighbors. Both algorithms need to perform local neighborhood discovery.

The key differences between wireless ad hoc networks and sensor networks are two folds from the perspective of power saving: First, algorithms used for wireless ad hoc networks do not address the issue of sensing coverage. Second, although reducing power consumption is a common design objective, algorithms used for wireless ad hoc networks often aim to maximize the life time of each individual node, while those used for sensor networks aim to maximize the time interval of continuously performing some (monitoring) functions. As long as the coverage and connectivity is maintained, a sensor network is considered to function well even if some sensors die much earlier than others.

Several centralized and distributed algorithms have been proposed for sensing coverage in sensor networks [6, 11, 24–26, 28, 29]. Slijepcevic *et al.* [24] address the problem of finding the maximal number of covers in a sensor network, where a cover is defined as a set of nodes that can completely cover the monitored area. They prove the NP completeness of this problem, and provide a centralized heuristic solution. They show that the proposed algorithm approaches the upper bound of the solution under most cases. It is, however, not clear how to implement the solution algorithm in a distributed manner.

Cerpa and Estrin [6] present ASCENT, to automatically configure sensor network topologies. In ASCENT, each node measures the number of active neighbors and the per-link data loss rate through data traffic. Based on these two values, it decides whether to sleep or keep awake. ASCENT does not consider the issue of completely covering the monitored region either.

Tian *et al.* [25] devise an algorithm that ensures complete coverage using the concept of “sponsored area.” Whenever a sensor node receives a packet from one of its working neighbors, it calculates its sponsored area (defined as the maximal sector covered by the neighbor). If the union of all the sponsored areas of a sensor node covers the coverage disk of the node, the node turns itself off. As will be shown in Section 6, this approach may be less efficient than a hexagon based GAF-like algorithm. Moreover, the authors only address the coverage problem without investigating the connectivity problem.

Ye *et al.* [28, 29] present PEAS, a distributed, probing-based density control algorithm for robust sensing coverage. In this work, a subset of nodes operate in the active mode to maintain coverage while others are put into sleep. A sleeping node wakes up occasionally to check if there exist working nodes in its vicinity. If no working nodes are within its probing range, it starts to operate in the active mode; otherwise, it sleeps again. The probing range can be adjusted to achieve different levels of coverage redundancy. The algorithm guarantees that the distance between any pair of working nodes is at least the probing range, but does not ensure that the coverage area of a sleeping node is completely covered by working nodes, i.e., it does not guarantee complete coverage.

Gupta *et al.* [11] devise both a centralized and a distributed algorithm to find a subset of nodes that ensure both coverage and connectivity. The centralized algorithm guarantees that the size of the formed subset is within $O(\log n)$ factor of the optimal size, where n is the network size. However, the distributed algorithm is heuristic-based and does not guarantee the $O(\log n)$ factor. It is also difficult to implement the distributed algorithm because it requires each node to reliably broadcast messages to all the nodes within $2r$ hops, where r is the maximum number of hops between any two nodes whose sensing regions intersect. In fact, the value of r has to be found out.

It has recently come to our attention that Wang *et al.* [26] have investigated the same problem and come closest to ours. In particular, they also observe that coverage infers connectivity if the radio range is at least twice the sensing range ($r_t \geq 2r_s$), and that if all the crossing points inside a region (or disk) are covered then the region (or disk) is covered. In their Coverage and Configuration Protocol (CCP), each node collects neighboring information and then use this as an eligibility rule to decide if a node can sleep. In the case of radio range is less than twice the sensing range, they combine their protocol with SPAN [7] to form a connected covering set.

The major differences between the work reported in [26] and ours lie, however in that (i) in our work, we intend to find the *minimum* number of sensors that maintain coverage and connectivity. We first transform the problem of minimizing the number of working nodes into that of minimizing overlap, and then derive the optimal conditions for minimizing overlap. We have also extended derivation of the optimal condition to accommodate the case of non-uniform sensing ranges; (ii) our OGDC algorithm is based on the above optimization analysis and is hence theoretically founded. As will be shown in Section 6, OGDC requires less working nodes to maintain coverage and connectivity; and (iii) we show the condition $r_t \geq 2r_s$ is also necessary for complete coverage to imply connectivity in an arbitrary network.

It should also be noted that the work reported in [15, 18] gives a totally different definition on coverage. Coverage in these pieces of work is defined as finding a path through a sensor network, given the location of all sensors. Two coverage problems are studied: the best coverage problem attempts to find the path that minimizes the maximal distance of all points to their closest sensors, while the worst coverage problem attempts to find the path which maximizes the minimum distance of all points on the path to their closest sensors. In particular, Meguerdichian *et al.* [18] present centralized algorithms for both the best and worst coverage problems, and Li *et al.* [15] give localized algorithms for both problems. Another related problem is to deploy a minimum number of base stations in cellular networks so as to cover the maximal area. The work reported in [16, 19] approaches this problem via devising centralized numerical methods.

6 PERFORMANCE EVALUATION

6.1 Simulation Environment Setup

To validate and evaluate the proposed design of OGDC, we have implemented it in *ns-2* [1] with the CMU wireless extension, and conducted a simulation study in a $50 \times 50\text{m}^2$ region where up to 1000 sensors are uniformly randomly distributed. Each data point reported below is an average of 20 simulation runs unless specified.

Schemes for comparison. In addition to evaluating OGDC, we also evaluate the performance of the PEAS algorithm proposed by Ye *et al.* [29], the CCP algorithm by Wang *et al.* [26] and a hexagon-based GAF-like algorithm. The former two algorithms have been introduced in Section 5. The latter (hexagon-based GAF-like) algorithm is built upon GAF [27] and operates as follows. The entire region is divided into square grids and one node is selected to be awake in each grid. To maintain coverage, the grid size must be less than or equal to $r_s/\sqrt{2}$. Thus, for a large area with size $l \times l$, it requires $\frac{2l^2}{r_s^2}$ nodes to operate in the active mode to ensure complete coverage. As pointed out by [14], hexagonal grids are more "homogeneous" than square grids and thus offer more scaling benefits, e.g., the number of working nodes is significantly smaller. To maintain coverage in hexagonal grids, the side length of each hexagon is at most $r_s/2$, and it requires $\frac{8l^2}{3\sqrt{3}r_s^2} \approx \frac{1.54l^2}{r_s^2}$ working nodes to completely cover a large area with size $l \times l$. As will be discussed below, the hexagon-based GAF-like algorithm performs better than the "sponsored area" algorithm [25], and hence the latter is not included in the comparison.

Parameters used. We use the energy model in [29], where the power consumption ratio for transmitting, receiving (idling) and sleeping is 20:4:0.01. We define one unit of energy (power) as that required for a node to remain idle for 1 second. Each node has a sensing range of $r_s = 10$ meters, and a lifetime of 5000 seconds if it is idle all the time.

The tunable parameters in OGDC are set as follows: the round time is set to 1000 seconds, the power threshold P_t is set to the level that allows a node to be idle for 900 seconds, the timer values are set to, respectively, $T_d = 10$ ms, $T_s = 1$ s, and $T_e = T_s/5 = 200$ ms, t_0 is set to the time it takes to send a power-on packet, $6.8ms$ (the wireless communication capacity is 40Kbps, the packet size is 34 bytes). The constants used in Eqs. (12) and (13) are set, respectively, to $c = \frac{10}{r_s^2}$, and p_0 is set to $1/N$ where N is the total number of sensors. Table 1 summarizes all the parameter values used.

Although OGDC involves tuning of several parameters, we have found that its performance is rather insensitive to the parameter values, as long as they are set in compliance with the guidelines discussed in Section 4.4. The system parameters, such as the initial energy of a node, the radio transmission rate, and the energy consumption rate, are the same for all the nodes.

Performance metrics. The performance metrics of interest are (i) the percentage of coverage, i.e. the ratio of the covered area to the total area to be monitored; (ii) the number of working nodes required to provide the percentage of coverage in (i); and (iii) α-lifetime, defined as the total time during which at least α portion of the total area is covered by at least one node. The conventionally defined network lifetime is then 100%-lifetime. Note that the lifetime definition used in this paper is slightly different from that in [28], where the lifetime is defined as the time interval until which coverage falls below a pre-determined percentage and never comes back again.

In the first part of the simulation, we assume the transmission range is at least twice the sensing range (which is set to 20m) so that we can focus on coverage alone. In the second part, we simulate the cases in which the transmission range is smaller than twice the sensing range.

6.2 Simulation in the Cases of Sufficient Transmission Ranges

We measure coverage as follows: we divide the area into 50 × 50 square grids, and a grid is considered covered if the center of the grid is covered, and coverage is defined as the ratio of the number of grids that are covered by at least one sensor to the total number of grids. In the $50 \times 50\text{m}^2$ area, 45 hexagon cells are required to cover the entire area if the hexagon-based GAF-like algorithm is used (Fig 9). Hence, the hexagon-based algorithm ensures 100% coverage if at least 45 sensors operate in the active mode in each round, one for each cell. However, at least 47 nodes are required to

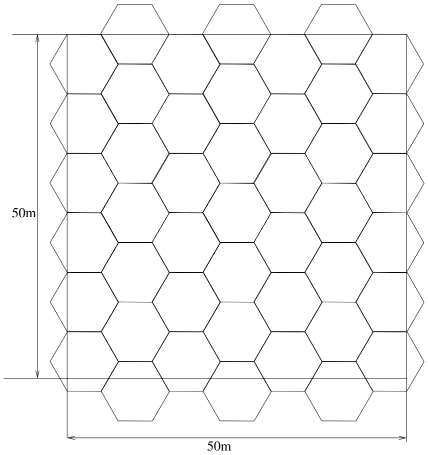
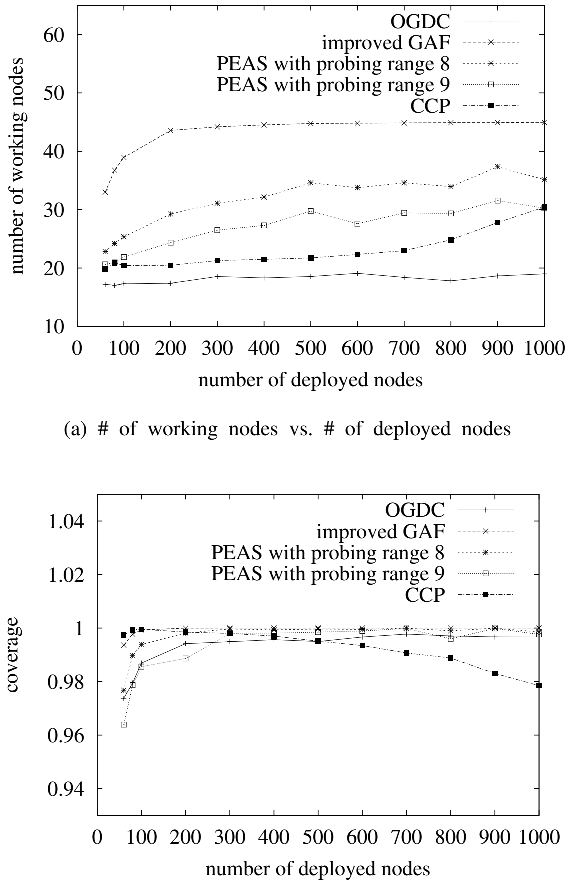


FIGURE 9
45 hexagons are required to cover a 50 × 50 m^2 area.

operate in the active mode under the "sponsored area" algorithm proposed in [25] to ensure the complete coverage. When the number of sensor nodes in the sensor network increases, the sponsored area algorithm requires more nodes to cover the entire area. As the sponsored area algorithm performs worse than the hexagon-based, GAF-like method, we do not include the sponsored area algorithm [25] in the following comparison.

Number of working nodes and coverage. Fig. 10 shows the number of working nodes and coverage versus the number of sensor nodes deployed in the network. Both metrics are measured after the density control process is completed. Under most cases, OGDC takes less than 1 second to perform density control in each round, while PEAS [29] and CCP [26] may take up to 100 seconds. As shown in Fig. 10, OGDC needs only half as many nodes to operate in the active mode as compared to the hexagon-based GAF-like algorithm, but achieves almost the same coverage (in most cases OGDC achieves more than 99.5% coverage). As the PEAS algorithm can control the number of working nodes by using different probing ranges, we tried two different probing ranges: 8m and 9m. (Using a probing range of 10m leads to insufficient coverage, the result of which is thus not reported here.) As shown in Fig. 10, using a smaller probing range results in more working nodes. With a probing range of 9m, the resulting coverage is less than that achieved by OGDC, while the number of working nodes is up to 50% more than that of OGDC. Moreover, the number of working nodes required under OGDC modestly increases with the number of sensor nodes deployed, while both PEAS and CCP incur a 50% increase in the number of working nodes, when the number of sensor nodes deployed in

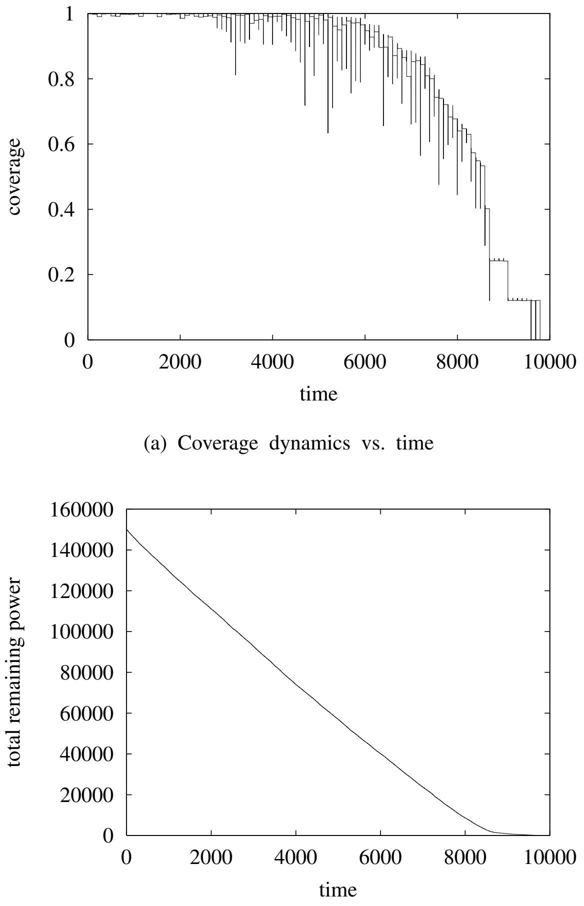


(a) # of working nodes vs. # of deployed nodes

(b) Coverage vs. # of deployed nodes

FIGURE 10
# of working nodes and coverage versus # of sensor nodes in a 50 × 50 m^2 area.

the network increases from 100 to 1000. We also observe that when the number of working nodes becomes very large, the coverage ratio of CCP actually decreases. This is because a large number of message exchanges are required in CCP to maintain neighborhood information. When the network density is high, packets incur collision more often and the neighborhood information may be inaccurate. In contrast, in OGDC each working node sends out at most one power-on message in each round, and as a result the packet collision problem is not so serious. The result of CCP reported here is a little different from that is reported in [26] because it assumes error-free channel conditions (no collisions, etc) in [26].



(a) Coverage dynamics vs. time

(b) Total remaining power vs. time

FIGURE 11
Dynamics of the sensing coverage and the total remaining power versus time under OGDC in a sensor network of 300 sensor nodes in a 50 × 50 m^2 area.

Fig. 11 shows the dynamics of coverage and total remaining power over the time in a typical simulation run for a sensor network of 300 sensor nodes in a 50 × 50 m^2 area. OGDC can provide over 95% coverage for appropriately 10 times of the lifetime of a single sensor node and the total power of the network decreases smoothly.

α-lifetime. Fig. 12 compares the α-lifetime achieved by OGDC, PEAS and CCP in a sensor network of 300 nodes, where α varies from 98% to 50%. For the PEAS algorithm we again tried two different probing ranges:

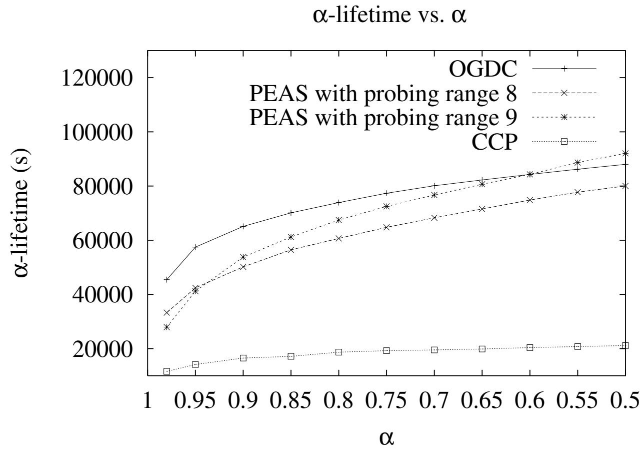
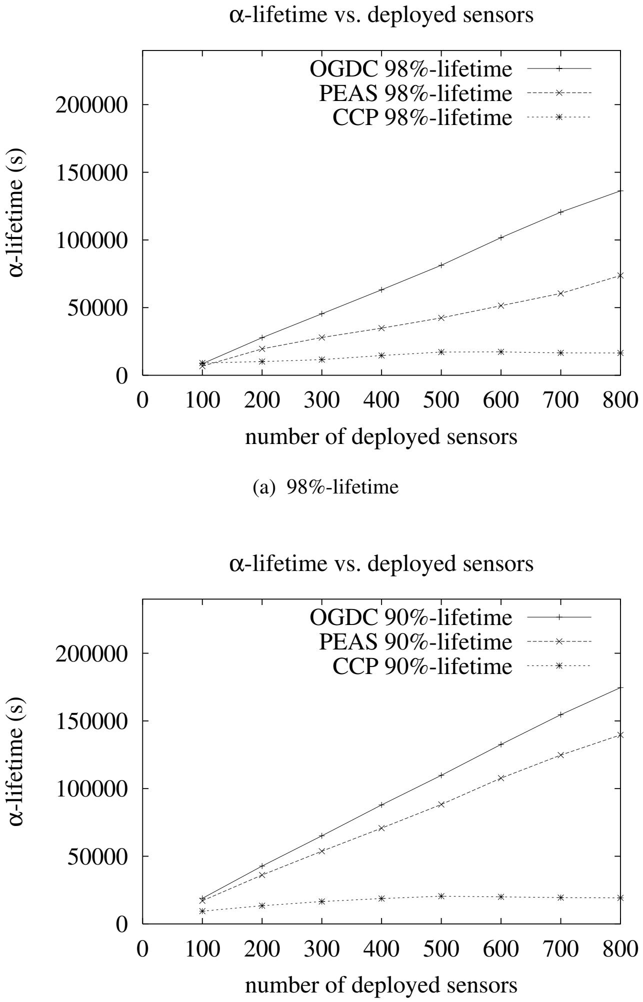


FIGURE 12
Comparison of α-lifetime versus α under OGDC, PEAS and CCP.

8m and 9m. As shown in Fig. 12, for a reasonably large α, the α-lifetime of PEAS is much shorter than that of OGDC. Only when α is less than 60%, the lifetime of PEAS using the probing range 9m is longer than that of OGDC. This is because with a relatively small probing range, PEAS requires an excessive number of nodes to operate simultaneously. Hence, its lifetime is consistently shorter than OGDC. On the other hand, with a large probing range of 9m, PEAS only guarantees that no two working nodes are in each other's probing range and does not ensure complete coverage. Moreover, when a node dies, it may take more than 100 seconds for another node to wake up to take its place. During that transition period the network is not completely covered. As a result, the low percentage lifetime is prolonged in PEAS. A nice property of OGDC is that during most of the lifetime, the monitored region is covered with a high percentage. It is clear that OGDC is preferred to PEAS no matter what probing range is used, unless the desired coverage percentage is very low (i.e. less than 60%). Although CCP uses less working nodes than PEAS in most cases, its lifetime is much shorter than both PEAS and OGDC. This is due to two reasons. First, CCP needs to periodically broadcast hello messages, the operation of which consumes energy. Second, in CCP when a node wakes up from the sleep mode it must stay awake and wait until it receives hello messages from sufficient number of neighbors that can cover its coverage region.

Fig. 13 shows the 98%-lifetime and 90%-lifetime under OGDC, CCP and PEAS with a probing range of 9m, when the number of sensor nodes deployed in a network varies from 100 to 800. The α-lifetime scales linearly



(a) 98%-lifetime

(b) 90%-lifetime

FIGURE 13
Comparison of α-lifetime versus number of sensor nodes under OGDC, PEAS (with probing range 9m) and CCP.

as the number of sensors deployed increases for both OGDC and PEAS algorithms. However, OGDC achieves nearly 100% more 98%-lifetime and 40% more 90%-lifetime than PEAS does. Again CCP achieves a much shorter lifetime than OGDC and PEAS.

For applications that require high levels of tracking accuracy and reliability, it may be desirable that each point is covered by multiple sensors. To this

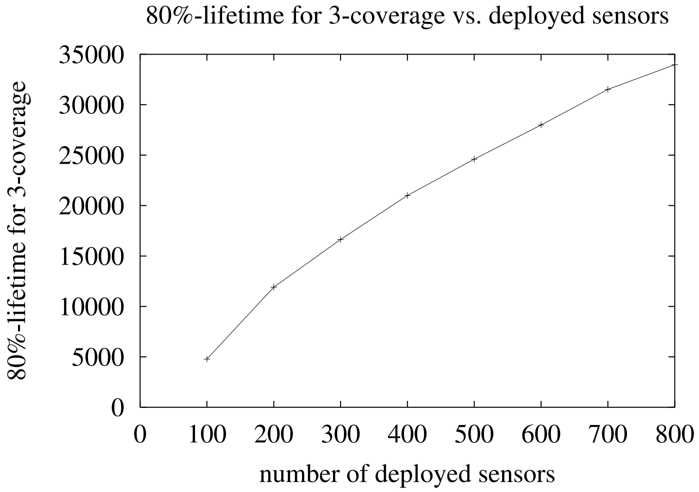
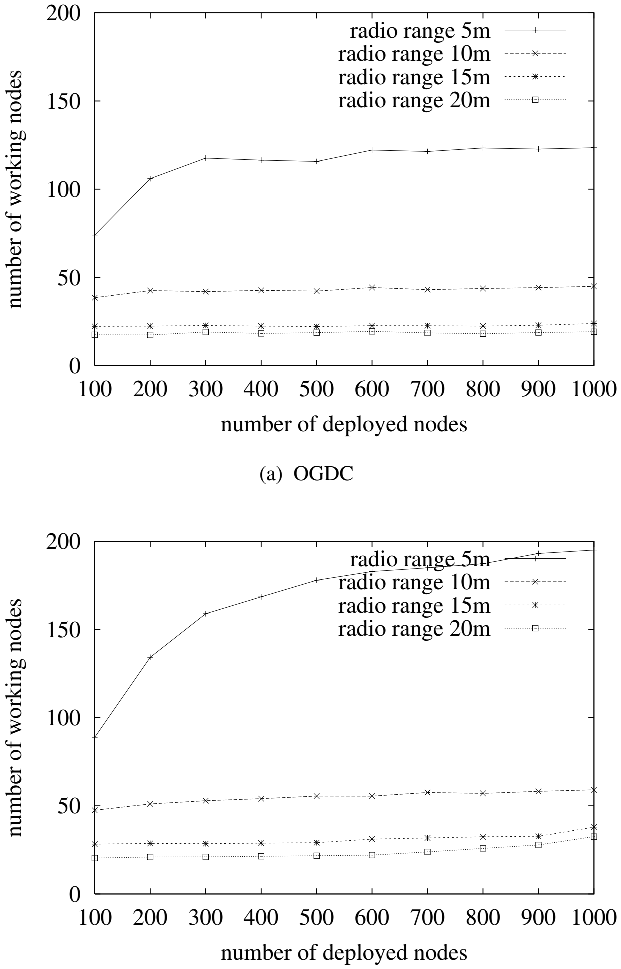


FIGURE 14
80%-lifetime with 3-coverage versus number of sensor nodes under OGDC.

end, we define k-coverage as that each point in an area is covered by at least k sensor nodes. OGDC can be readily extended to accommodate k-coverage as follows: a node is only turned off when each grid point in the node's coverage area is covered by at least k other nodes. Figure 14 shows the curve of 80%-lifetime with 3-coverage versus the number of sensor nodes. Again the 80%-lifetime linearly increases with the number of sensor nodes deployed in the network. A more in-depth study on k-coverage is a subject of our future research.

6.3 Simulation in the Cases of Insufficient Transmission Range

We now investigate the effect of small transmission ranges on coverage and connectivity. Since PEAS does not consider the connectivity issue, we only compare OGDC against CCP. Fig. 15 shows the number of working nodes versus the number of sensor nodes deployed with respect to different radio transmission ranges r_t under OGDC and CCP. OGDC uses a much smaller number of working nodes than CCP, especially when the radio range is small. Due to wireless channel errors, the sensor network may not always be connected in the case of small radio ranges, even if all the sensor nodes are powered on. Hence, instead of using the coverage of the network as the performance index, we measure the coverage of the largest connected component and plot the result in Fig. 16. The coverage of the largest connected component is very close to 1 under both algorithms, except in the cases that the number of sensor nodes deployed and the radio range are both small (e.g., $n = 100$ and $r_t = 5$). As a matter of fact, in the



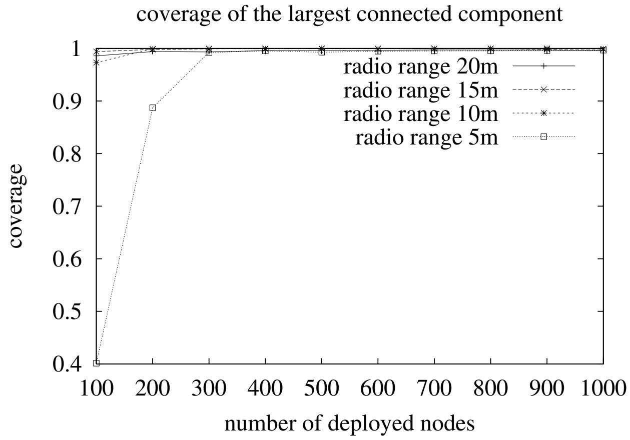
(a) OGDC

(b) CCP

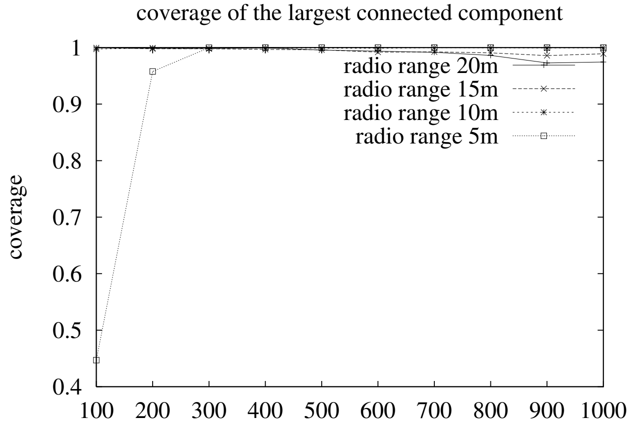
FIGURE 15
Number of working nodes versus number of sensor nodes deployed with respect to different radio ranges under OGDC and CCP (the sensing range is fixed at 10m).

case of $n = 100$ and $r_t = 5$, the sensor network with all the sensor nodes active is not connected, and has more than 18 connected components with a 45% coverage for the largest connected component in average.

In general we observe that as the radio range decreases, the coverage increases slightly and the number of nodes also increases. This is the cost for maintaining connectivity. However, the number of working nodes grows far less than the inverse of the square of the radio range.



(a) OGDC



(b) CCP

FIGURE 16
Coverage of the largest connected component versus the number of sensor nodes deployed with respect to different radio ranges under OGDC and CCP (the sensing range is fixed at 10m).

7 CONCLUSIONS AND FUTURE WORKS

In this paper we have investigated the issues of maintaining coverage and connectivity by keeping a minimum number of sensor nodes to operate in the active mode in wireless sensor networks. We begin with a discussion on the relationship between coverage and connectivity, and show that if the radio range is at least twice the sensing range, then complete coverage implies connectivity. Hence, if the condition holds, we only need to consider the coverage problem. Then, we derive, under the ideal case in which node density is sufficiently high, a set of optimality conditions under which a subset of working sensor nodes can be chosen for complete coverage. Based on the optimality conditions, we then devise a decentralized and localized density control algorithm, OGDC. OGDC is fully localized and can maintain coverage as well as connectivity, regardless of the relationship between the radio range and the sensing range. *Ns-2* simulations show that OGDC outperforms the PEAS algorithm [29], the CCP algorithm [26], the hexagon-based GAF-like algorithm, and the sponsor area algorithm [25] with respect to the number of working nodes needed and network lifetime (with up to 50% improvement), and achieves almost the same coverage as the best algorithm.

We have identified several avenues for future research. First, OGDC requires that each node knows its own location. However, we claim that this requirement can be relaxed to that each node knows its relative location to its neighbors. We are in the process of verifying this claim. Second, as mentioned in Section 6, we will look into the issue of *k*-coverage and its impact on fault tolerance. Finally, to better evaluate OGDC (or other density control algorithms), we will endeavor to derive the upper bound of the network lifetime when density control is in effect.

REFRENCES

[1] ns-2 network simulator, http://www.isi.edu/nsnam/ns.

[2] I F Akyildiz, W Su, Y Sankarasubramaniam and E Cayirci (Mar 2002) *Wireless Sensor Networks: A Survey, Computer Networks*.

[3] KEYENCE America, http://www.keyence.com/products/sensors.html.

[4] N Bulusu, J Heidemann and D Estrin (Oct 2000) GPS-less low cost outdoor localization for very small devices, *IEEE Personal Communications Magazine*, **7(5)**, 28–34.

[5] Nirupama Bulusu (2002) *Self-Configuring Localization Systems*, PhD thesis, University of California, Los Angeles.

[6] A Cerpa and D Estrin, Ascent: Adaptive self-configuring sensor networks topologies, In *Proc. of Infocom 2002*.

[7] B Chen, K Jamieson, H Balakrishnan and R MOrris (2001) Span: An energy-efficient operation in multihop wireless ad hoc networks, In *Proc. of ACM MobiCom'01*.

[8] Inc. Crossbow Technology, http://www.xbow.com/support/support_pdf_files/mts-mda_series_user_manual_revb.pdf.

[9] L Doherty, L El Ghaoui and K S J. Pister (Apr 2001) Convex position estimation in wireless sensor networks, In *Proc. of IEEE Infocom 2001*, Anchorage, AK.

[10] D Estrin, R Govindan, J S Heidemann, and S Kumar (Aug 1999) Next century challenges: Scalable coordination in sensor networks, In *Proc. of ACM MobiCom'99*, Washington.

[11] H Gupta, S Das and Q Gu, Connected sensor cover: Self-organization of sensor networks for efficient query execution. In *Proc. of Mobihoc 2003*.

[12] P Hall (1988) *Introduction to the Theory of Coverage Processes*.

[13] J M Kahn, R H Katz and K S J Pister (Aug 1999) Next century challenges: Mobile networking for "smart dust", In *Proc. of ACM MobiCom'99*.

[14] S Kandula and J C Hou (Aug 2002) Hierarchical clustering for data-centric sensor networks, Technical report, University of Illinois at Urbana, Champaign.

[15] X Li, P Wan and O Frieder (Apr 28 – May 2nd 2002) Coverage in wireless ad-hoc sensor networks, In *ICC 2002*, New York City.

[16] K Lieska, E Laitinen and J Lahteenmaki (Sep 1998) Radio coverage optimization with genetic algorithms. In *IEEE International Sysmpsium on Personal, Indoor and Mobile Radio Communications*, **1**, 318–322.

[17] A Mainwaring, J Polastre, R Szewczyk and D Culler (Aug 2002) Wireless sensor networks for habitat monitoring, In *First ACM International Workshop on Wireless Workshop in Wireless Sensor Networks and Applications (WSNA 2002)*.

[18] S Meguerdichian, F Koushanfar, M Potkonjak and M B Srivastava (2001) Coverage problems in wireless ad-hoc sensor networks, In *INFOCOM*, 1380–1387.

[19] A Molina, G E Athanasiadou and A R Nix (May 1999) The automatic location of base-station for optimized cellular coverage: a new combinatiorial approach, In *IEEE 49th Vehicular Technology Conference*, **1**, 606–610.

[20] Motes, http://www.xbow.com/products/wireless_sensor_networks.htm.

[21] A Savvides, C Han and M Strivastava (2001) Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of ACM MOBICOM'01*, 166–179, ACM Press.

[22] Infrarad Sensor, http://www.interq.or.jp/japan/se-inoue/e_pyro.htm.

[23] E Shih, S Cho, N Ickes, R Min, A Sinha, A Wang and A Chandrakasan (July 2001) Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, In *Proc. of ACM MobiCom'01*, Rome, Italy.

[24] S Slijepcevic and M Potkonjak (June 2001) Power efficient organization of wireless sensor networks, In *ICC 2001*, Helsinki, Finland.

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

[26] X Wang, G Xing, Y Zhang, C Lu, R Pless and C Gill (Nov 2003) Integrated coverage and connectivity configuration in wireless sensor networks, In *ACM Sensys'03*.

[27] Y Xu, J Heidemann and D Estrin (July 2001) Geography-informed energy conservation for ad hoc routing, In *Proc. of ACM MOBICOM'01*, Rome, Italy.

[28] F Ye, G Zhong, S Lu and L Zhang (2002) Energy efficient robust sensing coverage in large sensor networks, Technical report, UCLA.

[29] F Ye, G Zhong, S Lu and L Zhang (2003) Peas: A robust energy conserving protocol for long-lived sensor networks, In *The 23nd International Conference on Distributed Computing Systems (ICDCS)*.

APPENDIX 1

PROOF OF LEMMA 3

We prove the Lemma by showing that given the conditions stated in the lemma, the number of working sensor nodes and the overlap have a linear relationship with a positive slope.

Let the indicator function of a working node i, $I_i(x)$, be defined as

$$I_i(x) = \begin{cases} 1, & \text{if } x \text{ is within the coverage area of node } i, \\ 0, & \text{otherwise.} \end{cases}$$

Let R' be a region that contains R and the coverage areas of all sensor nodes. Then the coverage area of a sensor node i is a disk with the size $\int_{R'} I_i(x)dx \overset{\triangle}{=} |S_i|$, where $|S_i|$ denotes the size of the area S_i covered by sensor node i. By condition (ii), $|S_i| = |S|$ for all i. With the definition of $I_i(x)$, the overlap at point x can be written as

$$L(x) = \sum_{i=1}^{N} I_i(x) - I_R(x), \tag{14}$$

where N is the number of working nodes, and the overlap of sensing areas of all the sensor nodes, L, can be written as

$$\begin{aligned} L &= \int_{R'} L(x)dx \\ &= \int_{R'} (\sum_{i=1}^{N} I_i(x) - I_R(x))dx \\ &= \sum_{i=1}^{N} \int_{R'} I_i(x)dx - |R| \\ &= N|S| - |R|, \end{aligned} \tag{15}$$

where condition (i) is implied in the first equality and condition (ii) is implied in the fourth equality. Eq. (15) states that minimizing the number of working nodes N is equivalent to minimizing the overlap of sensing areas of all the sensor nodes L. □

APPENDIX 2

PROOF OF LEMMA 4

There are multiple coverage areas centered at C_i's and they all intersect at point O. We assume that the centers of these coverage areas are labeled as C_i, with the index i increasing clockwise. (Fig. 3 gives the case of $k = 3$, where $C_1 = A$, $C_2 = B$, and $C_3 = C$.) Now we have $\sum_{i=1}^{k} \angle C_i O C_{(i \mod k)+1} = 2\pi$ and $\angle C_i O C_{(i \mod k)+1} + \alpha_i = \pi$. From the above equations, we have $\sum_{i=1}^{k} \alpha_i = (k-2)\pi$. □

TABLE 2
Radio transmission range of Berkeley Motes [20]

Product	Transmission Range
MPR300*	30m
MPR400CB	150m
MPR410CB	300m
MPR420CB	300m
MPR500CA	150m
MPR510CA	300m
MPR520CA	300m

*MPR300 is the second generation sensors, while the rest are the third generation sensors.

TABLE 3
Sensing range of several typical sensors

Product	Sensing Range	Typical Applications
HMC1002 Magnetometer sensor [8]	5m	Detecting disturbance from automobiles
Reflective type photoelectric sensor [3]	1m	Detecting targets of virtually any material
Thrubeam type photoelectric sensor [3]	10m	Detecting targets of virtually any material
Pyroelectric infrared sensor (RE814S) [22]	30m	Detecting moving objects
Acoustic sensor on Berkeley Motes* [8]	$\sim$ 1m	Detecting acoustic sound sources

*This result is based on our own measurement on Berkeley motes [8].