

Movement-Assisted Sensor Redeployment Scheme for Network Lifetime Increase

Yinying Yang and Mihaela Cardei
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
{yyang4@, mihaela@cse.}fau.edu

ABSTRACT

Sensor deployment in mobile sensor networks has received significant attention in recent years. Goals during sensor deployment include improving coverage, achieving load balance, and prolonging the network lifetime. To improve the initial deployment, one possible method is to use mobile sensors, thus allowing sensors to relocate. In this paper, we present a sensor deployment strategy in mobile sensor networks. Our goal is to improve coverage and prolong network lifetime through sensor relocation after the initial deployment. The problem in this paper is defined as Movement-assisted Sensor Positioning (MSP) for network lifetime increase problem. With the observation that the sensors closer to the sink tend to consume more energy than those farther away from the sink, we first compute the desired non-uniform sensor density in the monitored area to reduce the energy holes near the sink and to prolong network lifetime. Assuming that sensors can move only once, we then propose a centralized algorithm to relocate mobile sensors to satisfy the density requirement with minimum cost. We construct the virtual multi-flow graph and solve the sensor relocation problem using a maximum-flow minimum-cost algorithm. The improvement in network lifetime is proved both mathematically and by simulations.

Categories and Subject Descriptors

C.2.2 [Computer-Communication Networks]: Network Protocols.

General Terms

Algorithms, Design, Performance.

Keywords

Sensor deployment, sensor mobility, network lifetime, coverage, wireless sensor networks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MSWIM'07, October 22–26, 2007, Chania, Crete Island, Greece.
Copyright 2007 ACM 978-1-59593-851-0/07/0010 ...\$5.00.

1. INTRODUCTION

Sensors are used to monitor and control the physical environment. A Wireless Sensor Network (WSN) is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it [1], [4]. Sensor nodes measure various parameters of the environment and transmit data collected to one or more sinks, using hop-by-hop communication. Once a sink receives sensed data, it processes and forwards it to the users. In mobile sensor networks, sensors can self-propel via springs [2], wheels [5], or they can be attached to transporters, such as robots [5] and vehicles [9].

A large number of sensors can be distributed in mass by scattering them from airplanes, rockets, or missiles [1]. In that case, the initial deployment is difficult to control. However, good deployment is necessary to improve coverage, achieve load balance, and prolong the network lifetime. Recent research has focused on methods to improve the initial deployment. One possible method is deploying additional sensors after the initial deployment [8]. Another method ([2], [11], [13], [14]) is to use mobile sensors, thus allowing sensors to relocate.

In a WSN, sensors closer to the sink tend to consume more energy than those farther away from the sink [10]. This is mainly because, besides transmitting their own packets, they will also forward packets on behalf of other sensors that are located farther away. If sensors in the network are uniformly deployed (uniform density), then the sensors closer to the sink will deplete their energy resources first [10], resulting in holes in the WSN. This uneven energy consumption will reduce network lifetime. We propose to prolong network lifetime by adjusting the sensor density depending on the distance to the sink.

In this paper, we first compute the desired non-uniform sensor density in the monitored area in order to reduce the energy holes near the sink and to prolong network lifetime. Our objective is that sensors consume energy at the same rate, thus resulting in a balanced energy consumption in the whole network. Compared with uniform deployment, the network lifetime improvement after applying the non-uniform distributed deployment is proved mathematically. The problem in this paper is defined as the Movement-assisted Sensor Positioning (MSP) for network lifetime increase problem. We exploit a mobility model similar to the one proposed in [2]. Assuming that sensors can move only once, we propose a centralized scheme to relocate mobile sensors to satisfy the density requirement.

The whole monitored area is divided into several coronas, which should have different target densities after redeployment depending on the distance to the sink. Coronas closer to the sink have higher densities. The network is also divided into small square regions, each of which belongs to one corona. With the initial and target number of sensors in each region, the network is transformed into a virtual multi-flow graph having a set of vertices and edges. There are three vertices in each region. The reachable relationships between vertices form the edges. Edge capacities are set based on both the initial and the target number of sensors in a region. Additional cost is generated when there is movement and cost values are set for edges connecting two different regions. Thus the problem is translated to a maximum-flow minimum-cost problem, which is solved using the Edmonds-Karp's algorithm.

Simulation results indicate that our solution effectively prolongs the network lifetime. The network lifetime is increased if sensors in the network have more choices on their movement distance. Another benefit provided by our sensor redeployment mechanism is an improvement in the coverage of the network.

The rest of the paper is organized as follows. In section 2 we present related works. In section 3 we show the improvement in network lifetime when using non-uniform sensor densities. We continue in section 4 with the definition of the movement assisted sensors positioning problem and our solution for sensor relocation using a maximum-flow minimum-cost algorithm. Section 5 presents simulation results and section 6 concludes our paper.

2. RELATED WORKS

There are recent research works focusing on improving initial deployment of a WSN by making use of sensors' mobility [2], [11], [13]. Some research works (e.g. [2]) provide centralized sensor deployment mechanisms, while others (e.g. [11], [13]) present distributed protocols.

In [2], Chellappan *et al.* study the flip-based deployment mechanism to achieve the maximum coverage. They assume the sensor can only flip once, and divide the whole network into multiple square regions. The centralized algorithm maximizes the number of regions that are covered by at least one sensor node with the minimum moving cost.

Wu and Yang present a scan-based distributed protocol in [13], called SMART. Their goal is to get uniform distribution via sensors relocation. The monitored network is divided into a 2-D mesh of clusters. For clusters on the same row, the first scan computes the total number of sensors in these clusters, the second scan forwards the average number of sensors in one cluster to each of them, and then each cluster determines give/take sensors to/from other clusters independently. The procedure is repeated for each row and then for each column. If there are holes in the network, they first plant seeds in holes followed by the row-wise and column-wise scan to achieve uniform distribution.

In [11], Wang *et al.* present three algorithms VEC, VOR, and Minimax. The Voronoi diagram is used in the paper. To maximize the coverage, in VEC, sensors that are too close to each other will be pushed away by the virtual force. The border of the network can also push sensors away. In VOR, when a node senses the coverage hole, it will be moved

towards the farthest vertex of the polygon in the Voronoi graph. Minimax algorithm is similar to VOR, the virtual force will pull sensors to sparser area, but the target location differs. It finds all the circles passing any two or three vertices in the polygon it belongs to. The target location for the sensor is the center of the circle with the minimum radius covering all the vertices of the polygon.

Similar to [2], [11] and [13], we make use of the mobility of sensors to relocate them after the initial deployment. However, their goals are mainly about improving the coverage and achieving load balance. Our paper presents a mechanism similar to that in [2] but with a different objective, which is to meet non-uniform density requirements in different coronas and maximize the network lifetime.

3. USING NON-UNIFORM SENSOR DISTRIBUTION TO IMPROVE NETWORK LIFETIME

Our study on the improvement in network lifetime using non-uniform sensor distribution is motivated by the study of the energy consumption in the uniform sensor deployment presented in [10]. This paper considers that sensors reporting data to a sink are uniformly distributed in a disk centered on the sink, which is partitioned in coronas as represented in Figure 1 (a). A sensor in corona C_i will transmit its own messages and will forward messages on behalf of sensors in corona C_{i+1} . A message relayed by a sensor in corona C_i will be forwarded by sensor nodes in coronas C_{i-1} , C_{i-2} , and so on until it reaches corona C_1 from where it will be transmitted to the sink. Note that the width of a corona is chosen so that a message will be forwarded by only one sensor in each corona.

Assuming that each sensor is equally likely to be the source of a path to the sink, the work [10] shows that to minimize the total energy spent on sending a message from a corona to the sink, all the coronas must have the same width. This triggers uneven energy depletion, with sensors in the first corona being the first to die due to energy depletion. Consequently this may result in network partitioning, with other sensors being unable to report their data to the sink.

Motivated by this result, our first objective is to compute the improvement in network lifetime when using a non-uniform sensor deployment.

3.1 Balanced Energy Consumption with Non-uniform Sensor Density

In modeling our problem, we divide the monitored area into a grid of regions, where each region is an $R \times R$ square. Then, we divide the area in coronas, as represented in Figures 1 (b) and (c). Let d be the width of each corona. For each region, take l to be the smallest distance between a point in the region and the sink. Then the region is part of the corona C_i for $i = \lceil l/d \rceil$. In this case the division in coronas is not circular, but it follows the regions' contour. When the region granularity is very small, $R \rightarrow 0$, then the division in coronas is similar to the one in Figure 1 (a), where coronas are circular.

We assume that the energy consumption is proportional to the number of messages transmitted and that sensors are uniformly deployed in the same corona. Intuitively, to balance energy consumption, we will deploy the fewest sensors

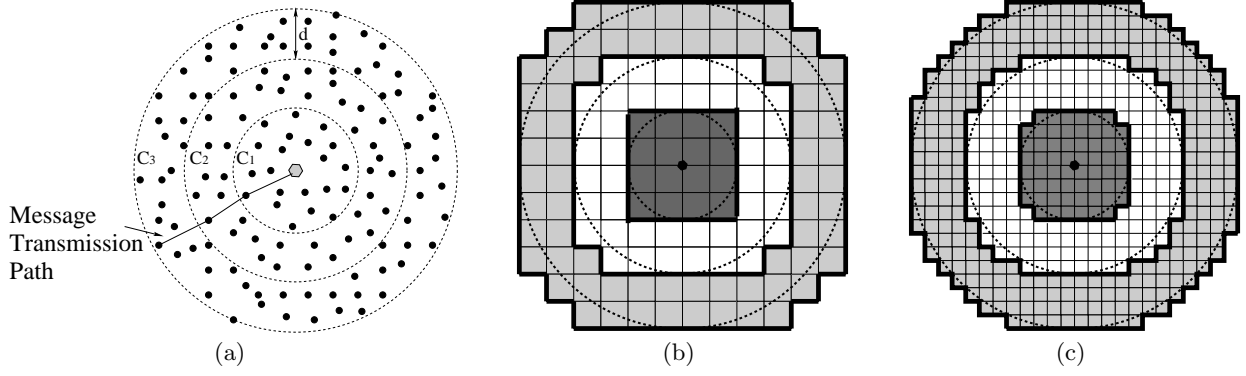


Figure 1: Sensing model using coronas concentric to the sink. (a) Division of monitored area in coronas, (b) Monitored area is divided into $R \times R$ square regions. Coronas are formed along these regions. (c) Division in coronas for smaller grid granularity R .

in the last corona C_n and the largest number of sensors closest to the sink, which is corona C_1 . We define ρ_i to be the sensor density in the corona C_i , thus $\rho_1 \geq \rho_2 \geq \dots \geq \rho_n$.

Our objective is to compute the sensor density for each corona so that all sensors deplete their energy at the same rate. This will balance the energy consumption. For this, we require that each sensor transmits the same number of messages in a specific time interval T . We model the data gathering as a periodic data reporting process, where one data message is generated from each unit of area during each time T . Based on this assumption, a sensor in corona C_n will generate $1/\rho_n$ messages in each time interval T . Also note that sensors in C_n do not forward messages on behalf of other sensors.

We compute now the number of messages transmitted by a sensor in corona C_i in time T . Such a sensor will generate $1/\rho_i$ messages with its own measurement results and will participate in forwarding of messages generated in coronas $C_{i+1}, C_{i+2}, \dots, C_n$.

Let us denote N_i the number of sensors in corona C_i and A_i the area of corona C_i . The total number of sensors is denoted as N , where $N = \sum_{i=1}^n N_i$. Parameter A represents the total monitored area, where $A = \sum_{i=1}^n A_i$. The number of messages generated in any corona C_i is N_i/ρ_i . A sensor in corona C_i participates in forwarding messages generated by sensors in coronas $C_{i+1}, C_{i+2}, \dots, C_n$, thus the number of forwarded messages in each time interval is:

$$\begin{aligned} \frac{\frac{N_{i+1}}{\rho_{i+1}} + \frac{N_{i+2}}{\rho_{i+2}} + \dots + \frac{N_n}{\rho_n}}{N_i} &= \frac{A_{i+1} + A_{i+2} + \dots + A_n}{A_i \cdot \rho_i} \\ &= \frac{A - (A_1 + A_2 + \dots + A_{i-1})}{A_i \cdot \rho_i} - \frac{1}{\rho_i} \end{aligned}$$

The total number of messages $TotalNum_i$ transmitted by a node in corona i is:

$$\begin{aligned} TotalNum_i &= \frac{1}{\rho_i} + \frac{A - (A_1 + A_2 + \dots + A_{i-1})}{A_i \cdot \rho_i} - \frac{1}{\rho_i} \\ &= \frac{A - (A_1 + A_2 + \dots + A_{i-1})}{A_i \cdot \rho_i} \end{aligned}$$

To balance the energy consumption, we require sensors in different coronas to consume the same energy, which means all sensors will send the same number of messages in time T , thus $TotalNum_n = TotalNum_i$. It follows that:

$$\begin{aligned} \frac{A - (A_1 + A_2 + \dots + A_{i-1})}{A_i \cdot \rho_i} &= \frac{1}{\rho_n} \\ \Rightarrow \rho_i &= \rho_n \cdot \frac{A - (A_1 + A_2 + \dots + A_{i-1})}{A_i} \end{aligned} \quad (1)$$

According to formula (1), the density of every corona can be determined based on ρ_n . Also, the value ρ_n can be computed as a value of the total number of sensors N and the coronas areas:

$$\begin{aligned} \rho_1 \cdot A_1 + \rho_2 \cdot A_2 + \dots + \rho_n \cdot A_n &= N \\ \rho_n \cdot \frac{A}{A_1} \cdot A_1 + \rho_n \cdot \frac{A - A_1}{A_2} \cdot A_2 + \dots \\ + \rho_n \cdot \frac{A - A_1 - A_2 - \dots - A_{n-1}}{A_n} \cdot A_n &= N \\ \Rightarrow \rho_n \cdot (n \cdot A - (n-1) \cdot A_1 - (n-2) \cdot A_2 \dots - A_{n-1}) &= N \\ \Rightarrow \rho_n &= \frac{N}{n \cdot A - (n-1) \cdot A_1 - (n-2) \cdot A_2 \dots - A_{n-1}} \\ \Rightarrow \rho_n &= \frac{N}{A_1 + 2 \cdot A_2 \dots + n \cdot A_n} = \frac{N}{\sum_{i=1}^n i \cdot A_i} \end{aligned} \quad (2)$$

3.2 Comparison in Network Lifetime for Non-uniform and Uniform Densities

In this section, we compare the improvement in network lifetime for circular coronas, see Figure 1 (a). First, we take the case of *non-uniform* sensor densities when $R \rightarrow 0$. In this case, the area of corona i is:

$$A_i = \pi(i \cdot d)^2 - \pi((i-1) \cdot d)^2 = \pi \cdot d^2(2 \cdot i - 1) \quad (3)$$

According to formula (1) we get:

$$\rho_i = \rho_n \cdot \frac{\pi \cdot (n \cdot d)^2 - \pi \cdot ((i-1) \cdot d)^2}{\pi \cdot d^2 \cdot (2 \cdot i - 1)} = \rho_n \cdot \frac{n^2 - (i-1)^2}{2 \cdot i - 1} \quad (4)$$

Let $A = \sum_{i=1}^n A_i = \pi(nd)^2$. Then, according to formulas (2) and (3), we obtain:

$$\rho_n = \frac{N}{A} \cdot \frac{6n}{4n^2 + 3n - 1} \Rightarrow TotalNum_n = \frac{A}{N} \cdot \frac{4n^2 + 3n - 1}{6n} \quad (5)$$

Let us now compute the number of messages transmitted by a sensor assuming a *uniform* sensor distribution ρ . Following the approach in section 3.1, we compute the total number of messages sent by a sensor in corona C_1 . As proved in [10], sensors in the first corona will die first, limiting the network lifetime.

$$TotalNum'_1 = \frac{1}{\rho} + \frac{N_2}{\rho} + \frac{N_3}{\rho} + \dots + \frac{N_n}{\rho}$$

After simple computations, we get:

$$TotalNum'_1 = \frac{A}{N} \cdot n^2 \quad (6)$$

Combining (5) and (6), the improvement in network lifetime obtained by deploying a non-uniform sensor distribution is:

$$\frac{TotalNum'_1}{TotalNum_n} = \frac{6n^3}{4n^2 + 3n - 1} \geq n \quad (7)$$

Equation (7) shows that by using a non-uniform sensor distribution we obtain a significant improvement in network lifetime, of at least n times.

4. MSP PROBLEM DEFINITION AND SOLUTION

Motivated by our study in section 3, our objective is to relocate (or move) sensors to balance energy consumption. We consider that sensors are initially randomly deployed to monitor an area of interest and that the sink is located in the center of the area. We also assume that sensors can move using the flip-based mobility model introduced in [2]. The objective is to determine an optimal movement (flip) plan for the sensors to achieve sensor densities according to the formulas in section 3.1.

4.1 MSP Problem Definition

The *mobility model* considered in this paper is flip-based, where a sensor can move from its current location to a new location when triggered by an appropriate signal. Such movements can be determined [2] by propellers powered by fuels, coiled springs unwinding during flips, etc. Similar to [2], we consider two movement models. In the first model, a sensor can flip only once, with a constant distance F . In the second model, a sensor can choose to move to different distances but it is allowed to flip exactly once. The moving distance is $d_m = k \times \delta$, where k is the flip choice, which is an integer greater than zero. δ is the moving granularity. We assume δ always equals to region size R . When flip choice $k = 1$, sensors are able to flip the distance δ away, which means they can only flip to the closest neighbor regions. If $k = 2$, sensors can choose to flip a distance δ or $2 \cdot \delta$. Also, we must satisfy $d_m \leq F$.

The problem that we address in this paper is the **Movement-assisted Sensors Positioning for network lifetime increase (MSP)**: *Given a sensor network with N sensors randomly deployed for periodical monitoring of an area A centered to a sink, use a flip-based mobility model to reposition sensors to maximize sensor network lifetime.*

Based on our previous discussion, our goal is to move sensors to obtain higher densities for the areas closer to the sink. In order to model the sensor movement, we divide the

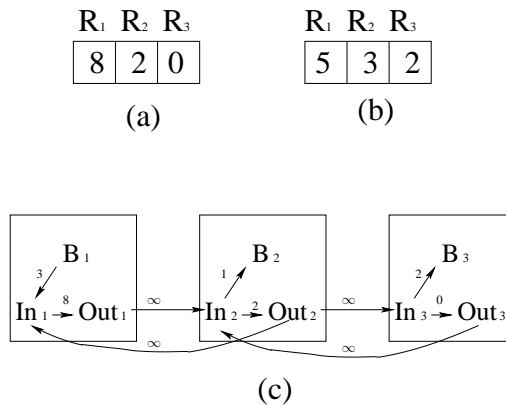


Figure 2: An example of a multi-flow graph. (a) Initial number of sensors in regions R_1 , R_2 , and R_3 , (b) Target number of sensors in regions R_1 , R_2 , and R_3 , (c) An example of the multi-flow graph for the initial deployment, in which B_1 is the source, B_2 and B_3 are holes.

monitored area A in squared regions and coronas, as represented in Figures 1 (b) and (c). R_s is the sensing range and R_c represents the transmission range.

First, we divide the area A into a grid with $R \times R$ squares. To guarantee the coverage and connectivity, we select R such that if a sensor is located in a square region, then every point inside the region is covered by the sensor, which means $R_s \geq R\sqrt{2}$. In addition, a sensor can directly communicate with sensors in the adjacent regions. The coverage and connectivity conditions are satisfied if $R \leq \min\{R_s/\sqrt{2}, R_c/\sqrt{5}\}$. Based on the flip movement model, a sensor will be able to flip to its neighbor regions: left, right, top, and bottom. This incurs an additional condition that $R \leq F$.

Second, we divide the area A in coronas, as represented in Figures 1 (b) and (c). The coronas are defined along the square regions' borders. We select the corona width d such that any node in corona C_i can directly reach a sensor in corona C_{i-1} . This is satisfied if $R_c \geq d + R \cdot \sqrt{2}$. This condition is needed for data forwarding, such that a message forwarded on its path to the sink is relayed by one sensor in each corona. To find which corona a region belongs to, take l , the smallest distance between a point in the region and the sink. The region is part of the corona C_i for $i = \lceil l/d \rceil$.

Thus, after the initial deployment, the WSN can be represented similar to Figure 3 (a), which is an example showing the upper right quarter of Figure 1 (b). The values in the square regions are the initial sensor densities. Knowing the total number of sensors N and the area of each corona, we are able to compute the target density values for each corona using formulas (1) and (2). Thus, depending on which corona each sensor belongs to, the target density of each region is determined, as shown in Figure 3 (b).

Our *MSP problem* can be reformulated as follows: *given a grid with $R \times R$ square regions, the initial and the target number of sensors of each region, and the flip-based mobility model, determine an optimal sensor movement plan to maximize the number of regions that achieve their target number of sensors, while simultaneously minimizing the total number of flips used.*

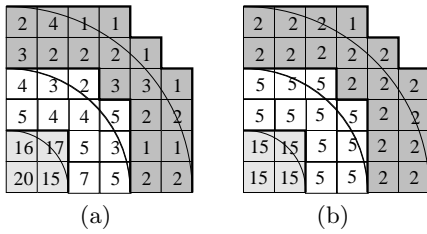


Figure 3: An example showing a quarter of the monitored area from Figure 1 (b). (a) Initial sensor deployment in each region, (b) Target number of sensors in each region.

4.2 Solution to the MSP Problem

To solve the MSP problem, we follow the framework in [2] using a generalization for the case when the desired number of sensors in each region is variable. According to the MSP problem, let us assume that after the initial deployment there are $s_1, s_2, s_3, \dots, s_m$ sensors in region $R_1, R_2, R_3, \dots, R_m$, and that the target number of sensors for each region is $t_1, t_2, t_3, \dots, t_m$.

The proposed scheme is centralized and can be executed as follows. Sensors in each region form a cluster. Sensors in the same cluster select one cluster-head based on a pre-defined priority (e.g. the energy level). Cluster-heads are in charge of communication with the sink and organizing the movement inside their clusters. Cluster-heads determine the number of sensors in their cluster and send the information to the sink along with the region coordinates. When the sink receives the global information, it computes the multi-flow graph, determines the movement plan for each cluster, and then feeds the results back to cluster-heads. Based on the feedback information, the cluster-head informs sensors in its cluster on their flipping movements by broadcasting a message including their moving distances and directions.

The scheme has two steps. In the first step, we construct a directed multi-flow graph $G = (V, E)$ for the initial deployment and in the second step we compute the movement plan using the multi-flow graph. This movement plan shows the way in which sensors will move (flip) to other regions for energy balancing.

In the first step, to form the multi-flow graph, three vertices are added for each region R_i :

1. B_i : Base vertex of the region, which keeps track of the number of sensors in the region R_i .
2. In_i : Vertex that keeps track of the number of sensors that moves from other regions.
3. Out_i : Vertex that keeps track of the number of sensors that can move to other regions.

Edges in E are added inside regions and between regions. Each edge has a capacity representing the maximum number of sensors that can be transmitted along this edge. Edges inside each region R_i are added as follows (see Figure 2):

1. Add an edge (In_i, Out_i) from In_i to Out_i with capacity s_i .
2. If $s_i \geq t_i$, then add an edge (B_i, In_i) from B_i to In_i with capacity $s_i - t_i$. If $s_i < t_i$ then add an edge (In_i, B_i) from In_i to B_i with capacity $t_i - s_i$.

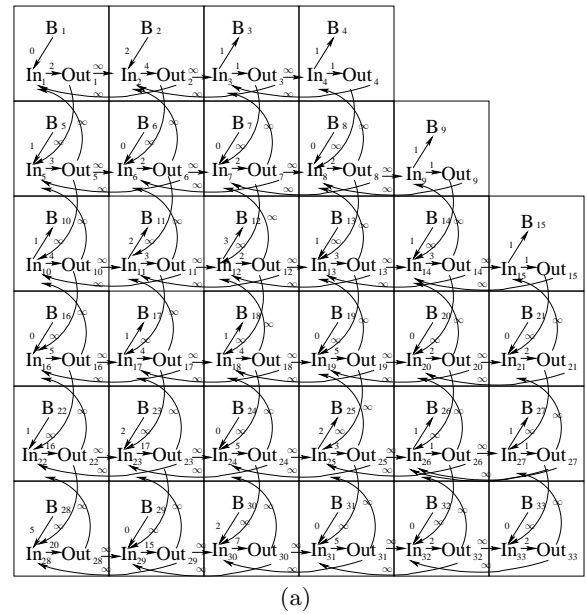


Figure 4: Multi-flow graph for the initial deployment from Figure 3.

The interpretation of the capacity value and the selection of the **source** and **hole** vertices are explained next. The capacity values between In_i and Out_i represents the number of sensors that can leave region R_i . When $s_i > t_i$, then the vertex B_i is a **source** since it has more sensors than the desired number, so it can transfer sensors to other regions. The capacity of the edge (B_i, In_i) shows the number of sensors that can be transferred to other regions. If $s_i < t_i$, then B_i is a **hole** since it has to assimilate more sensors to achieve the desired number of sensors; see Figure 2. The capacity of the edge (In_i, B_i) shows the number of sensors that need to be assimilated to reach the desired number of sensors.

Besides the edges inside regions, additional edges are added between two regions reachable from each other. A region can reach k regions on each direction: right, left, top, and bottom. Thus, a region will reach other $4 \times k$ regions. If two regions R_i and R_j are reachable from each other, then two additional edges are added to the graph as follows:

1. Add an edge (Out_i, In_j) from Out_i to In_j with capacity ∞ .
2. Add an edge (Out_j, In_i) from Out_j to In_i with capacity ∞ .

In order to determine the movement plan generated by movements between different regions, we use cost values defined as follows. For all edges inside regions, the cost is zero, since no sensor movement is involved. Edges between regions have a cost of one, since a flow on this edge represents a sensor's flipping from one region to the other.

1. $\text{cost}(Out_i, In_j) = \text{cost}(Out_j, In_i) = 1$.
2. cost on other edges = 0.

After we have built the multi-flow graph, the MSP problem reduces to finding the maximum flow in the network between sources and holes, with a minimum overall cost. To determine the minimum-cost maximum-flow, we use the Edmonds-Karp's algorithm in [3] and [6]. Then we determine the movement plan based on the flow edges between regions. There are other maximum-flow minimum-cost algorithms, and some of them have lower running time [7]. The process proposed in this section is summarized in the MSP-Algorithm.

Algorithm 1 MSP-Algorithm:

- 1: Form clusters within each region $R \times R$
 - 2: Each cluster-head sends to the sink: the number of sensors in its cluster and region coordinates
 - 3: The sink receives s_1, s_2, \dots, s_m values and computes the target number of sensors for each region t_1, t_2, \dots, t_m using formulas (1) and (2). The sink constructs the multi-flow graph $G(V, E)$ based on these information
 - 4: The sink uses Edmonds-Karp algorithm [3, 6] to determine sensors movement plan between regions
 - Initially, the flow f is set to zero
 - Repeatedly find a shortest augmenting path p in the residual network
 - Augment f along p by the minimum residual capacity along the path
 - When no augmenting paths exists, f is a minimum-cost maximum-flow
 - 5: The sink sends the sensor movement plan to the regions' cluster-heads
 - 6: Each cluster-head distributes the movement plan inside its cluster by broadcasting a message including the moving distances and directions of sensors in the cluster.
-

Let us now analyze the time complexity of our solution. Recall that we divide the monitored area into n coronas with the same width d , and then we divide it into $R \times R$ squares, so there are $O(\lceil (\frac{n-d}{R}) \rceil^2)$ squares in the monitored area. The time complexity of steps 1, 2, 5, and 6 in the **MSP-Algorithm** is $O(\lceil (\frac{n-d}{R}) \rceil^2)$. In step 3, the time complexity to compute the desired number of sensors for each region is $O(\lceil (\frac{n-d}{R}) \rceil^2)$. Denoting $|V|$ as the number of vertices and $|E|$ as the number of edges in the multi-flow graph, we have $|V| = O(\lceil (\frac{n-d}{R}) \rceil^2)$ and $|E| = O(\lceil \frac{d_m}{R} \rceil \cdot \lceil (\frac{n-d}{R}) \rceil^2)$ respectively, where d_m is the flip distance and $\lceil \frac{d_m}{R} \rceil$ denotes the number of reachable regions for each region. The time complexity to construct the multi-flow graph is $O(|V| + |E|)$. In step 4, Edmonds-Karp's algorithm takes $O(|V| \cdot |E|^2)$ [3, 6]. To summarize, the total time complexity for our solution is $O(|V| \cdot |E|^2) = O(\lceil \frac{d_m}{R} \rceil^2 \cdot \lceil (\frac{n-d}{R}) \rceil^6)$.

4.3 Example

In this section, we present an example with $N = 600$ sensors uniformly distributed in the monitored area. The upper right quarter of the monitored area is illustrated in Figure 3. We consider the following parameters: the radius of the monitored area is 6 units, corona width d is 2 units, and flip choice k is 1. The whole network is divided into 1×1 unit square regions. The area of each corona is $A_1 = 16$, $A_2 = 44$, and $A_3 = 72$.

According to formulas (1) and (2), we compute the den-

sities of each of the three coronas and obtain the following values: $\rho_3 = 1.9$ (approximated to $\rho_3 = 2$), $\rho_2 = 4.9$ (approximated to $\rho_2 = 5$), and $\rho_1 = 15.4$ (approximated to $\rho_1 = 15$). Figure 3 (a) shows the number of sensors in each region after the initial deployment and Figure 3 (b) shows the target number of sensors in each region. The multi-flow graph based on the original deployment is illustrated in Figure 4.

5. SIMULATION

In this section, we present the simulation results of our solution for the MSP-Algorithm. We study the network life improvement, coverage improvement, and the number of movements under different flip choices.

5.1 Simulation environment

Metrics used in the simulation include total coverage improvement, coverage improvement of holes, number of movements (NM), and network lifetime (NL). When a region contains at least the target number of sensors, it is defined as covered. Total coverage (TC) shows how many regions are covered, which is computed as $TC = \frac{N(RC)}{N(R)}$, where $N(RC)$ is the number of regions having at least the target number of sensors (covered) and $N(R)$ is the total number of regions.

Coverage improvement of holes is similar to the total coverage improvement. However, it focuses more on how much the holes' coverage is improved after redeployment. We use the following equation to compute the coverage for holes (CH): $CH = \frac{\sum_{i=1}^{N(H)} r_i}{N(H) t_i}$, where $N(H)$ is the number of holes in the initial deployment, r_i is the actual number of sensors in region R_i after redeployment using our solution, and t_i is the target number of sensors in region R_i . The number of movements represents the cost of the movement plan, which is defined as: $NM = \sum_{i=1}^N f_i$, where N is the total number of sensors and f_i is a Boolean value representing the number of flips of sensor i .

The network lifetime is defined as the number of rounds the network lasts before the first sensor runs out of energy. Each region generates one message every round. In our simulations, we account the energy consumed for message transmissions and consider that e energy is consumed per message, where $e = 1$ unit. Each sensor has total energy, $E = 1000$ units. The total number of rounds for each sensor i can be calculated as: $\frac{E_i}{M_i \cdot e}$, where M_i is the total number of messages node i transmits. The network lifetime is computed as the minimum number of rounds.

We conduct the simulation on a custom discrete event simulator, which could generate random initial sensor deployment. In the simulation, we used the following variable parameters:

- the diameter of the circle monitored area is 12, 16, and 20 units.
- the region size is 1 unit.
- the corona width is 2 units.
- the total number of sensors in the network varies from 200 to 3000.
- the flip choice, k , varies from 1 to 3.

All the tests are repeated 200 times. The collected data is averaged and reported in the following figures.

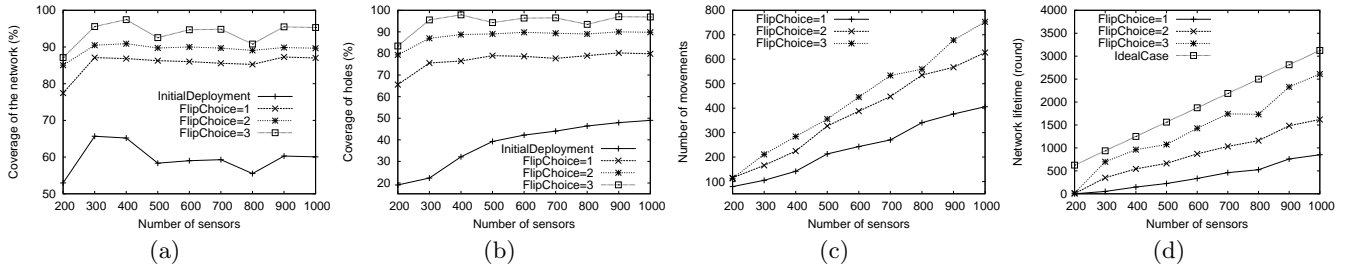


Figure 5: When area size is 12 units, corona width is 2 units, region size is 1 unit, (a) Total coverage improvement, (b) Coverage improvement of holes, (c) Number of movements, (d) Network lifetime.

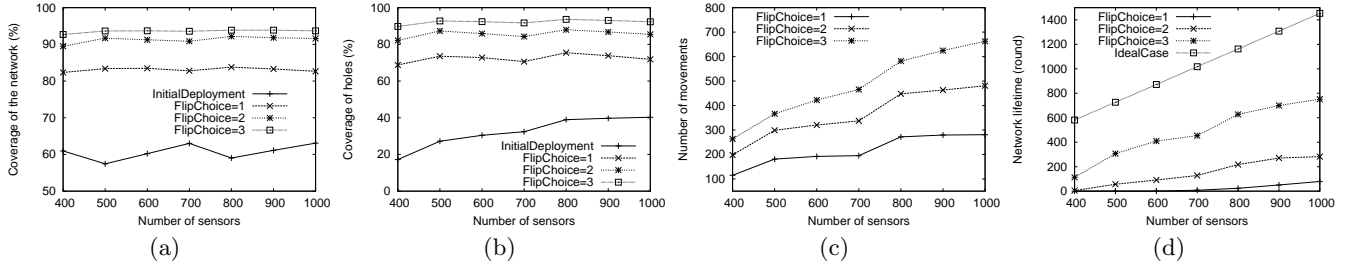


Figure 6: When area size is 16 units, corona width is 2 units, region size is 1 unit, (a) Total coverage improvement, (b) Coverage improvement of holes, (c) Number of movements, (d) Network lifetime.

5.2 Simulation results

In Figure 5, sensors are deployed in a circle area with the diameter 12 units, containing 3 coronas. Corona width is 2 units and the region size is 1 unit. Figures 5 (a) and (b) show the coverage improvement under different flip choices. Compared with the initial deployment, our solution with different flip choices can effectively improve the total coverage and the coverage of holes. The improvements are almost independent of the scale of the network in both figures. We also observe that with the increase of the flip choices, the coverage of the whole network and coverage of holes are increased. This is because when the flip choice is greater than one, sensors have more choices, and they can cover holes that are farther away.

There are two reasons why coverage improvement cannot reach 100% in Figures 5 (a) and (b). The main reason is that although some source regions still have redundant sensors, they cannot form a flow to holes because of the limitation of flip distance. So there may remain some holes after redeployment. The second reason is that the density we get from formulas (1) and (2) may be a decimal fraction instead of an integer, but when computing the target number for each region, we round this value to the closest integer. Therefore, in some cases, the network may lack sufficient sensor nodes to cover all regions.

Figure 5 (c) shows the number of movements under different flip choices. The number of movements represents the cost of the movement plan. The number of movements increases with the increase of k . The main reason is that when the sensors have more flip choices, they have more moving possibilities that can be efficiently used to cover holes.

When k increases, more links are added in the multi-flow graph, which means more redundant sensors in source regions are exploited to cover holes. Therefore, more flows can be formed, which improves the coverage and network lifetime at the cost of higher number of movements. A tradeoff between coverage improvement and the cost exists.

Figure 5 (d) presents the network lifetime under different flip choices. The first observation is that the more flip choices that sensors have, the closer the network lifetime gets to the ideal case (when all regions have the target number of sensors). This is due to the increase of the coverage improvement of holes with more flip choices. Second, the network lifetime also increases when the network scale increases. That is to say, denser networks will have longer network lifetime.

In Figure 6, the diameter of the area is increased to 16 units. The corona width and region size are the same as those in Figure 5, consequently the whole area contains one more corona. In this case, the test is conducted on a sparser environment. To guarantee that each region in the outermost corona contains at least one sensor, the number of sensors in the network begins at 400. The curves in Figure 6 have the similar trend to those in Figure 5. However, in general, when flip choice is 3, the sparse network gets worse coverage improvement and hence fewer movements and a shorter network lifetime.

In Figures 7 (a) and (b), we uniformly distribute sensors in the monitored area with the same average number of sensors per region. We compare the network lifetime among three networks with diameters 12 units, 16 units, and 20 units, each of which has 3, 4, and 5 coronas respectively. The

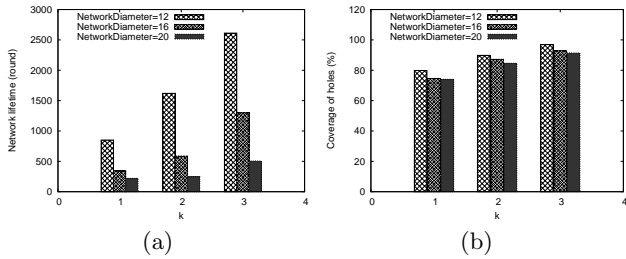


Figure 7: Sensors are uniformly distributed in the monitored area with the same average number of sensors per region (a) Comparison of the network lifetime, (b) Comparison of coverage improvement of holes.

corona width and region size are the same in all networks, but the total number of sensors differs from 1000 sensors (the case with 3 coronas) to 3000 sensors (the case with 5 coronas). The network containing 3 coronas has the longest network lifetime. This result is motivated by the observation that when there are more coronas, more data is forwarded to the sink, thus sensors deplete their energy resources faster. In addition, the messages generated by the outer coronas have to travel longer paths to the sink.

The simulation results can be summarized as follows:

1. With multiple flip choices, our solution effectively improves the total coverage and the coverage of holes. Improvements are almost independent of the scale of the network.
2. With multiple flip choices, the network lifetime is prolonged after redeployment. The more flip choices sensors have, the closer the network lifetime gets to the ideal case.
3. With the increase of the coverage, the movement cost is also increased.
4. Compared with dense networks, sparse networks have worse coverage improvement and shorter network lifetime.
5. When using the same initial deployment density, with the increase of the number of coronas in the network, the network life decreases.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we address the sensor redeployment to improve the coverage and to prolong the network lifetime, which is defined as the Movement-assisted Sensors Positioning (MSP) for network lifetime increase problem. With the observation that the sensors closer to the sink tend to consume more energy than those farther away from the sink, we first compute the desired non-uniform sensor density in the monitored area to reduce the energy holes near the sink. The improvement of network lifetime is proved mathematically. In our movement model, sensors have multiple flip choices on their moving distances but they can move only once. A centralized sensor redeployment strategy is presented so

that the density requirements can be satisfied with the minimum movement cost. We construct the virtual multi-flow graph for the initial deployment, and based on the graph we solve the sensor relocation problem using the maximum-flow minimum-cost algorithm. Simulation results show the solution can effectively prolong the network lifetime and improve the coverage of the total network and the coverage of holes.

In our future work, we plan to exploit distributed sensor deployment schemes for the MSP problem under the similar mobility model to prolong the network lifetime and improve the coverage.

Acknowledgments

This work was supported in part by NSF grants CCF 0545488 and CNS 0422762.

7. REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks: a survey, *Computer Networks*, 38(4), pp. 393-422, 2002.
- [2] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, Mobility limited flip-based sensor networks deployment, *IEEE Transactions of Parallel and Distributed Systems*, 18(2), pp. 199-211, 2007.
- [3] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*, MIT Press, 2001.
- [4] D. Culler, D. Estrin, and M. Srivastava, Overview of sensor networks, *IEEE Computer*, 37(8), pp. 41-49, 2004.
- [5] K. Dantu, M. H. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, Robomote: enabling mobility in sensor networks, *IPSN 2005*, pp. 404-409, 2005.
- [6] J. Edmonds and R. M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, *J. ACM*, 19(2), pp. 248-264, 1972.
- [7] A. V. Goldberg and R. Tarjan, Solving minimum-cost flow problems by successive approximation, *Proceedings of ACM Symposium on Theory of Computing (STOC)*, New York, 1987.
- [8] A. Howard, M. J. Mataric, and G. S. Sukhatme, An incremental self deployment algorithm for mobile sensor networks, *Autonomous Robots, Special Issue on Intelligent Embedded Systems*, Sep. 2002.
- [9] U. Lee, E.O. Magistretti, B.O. Zhou, M. Gerla, P. Bellavista, and A. Corradi, Efficient data harvesting in mobile sensor platforms, *PerCom Workshops*, pp. 352-356, 2006.
- [10] S. Olariu and I. Stojmenovic, Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting, *IEEE INFOCOM*, 2006.
- [11] G. Wang, G. Cao, and T. F. LaPorta, Movement-assisted sensor deployment, *IEEE Transactions on Mobile Computing*, 5(6), pp. 640-652, 2006.
- [12] G. Wang, G. Cao, T. F. La Porta, and W. Zhang, Sensor relocation in mobile sensor networks, *IEEE INFOCOM*, pp. 2302-2312, 2005.
- [13] J. Wu and S. Yang, SMART: a scan-based movement-assisted sensor deployment method in wireless sensor networks, *IEEE INFOCOM*, pp. 2313-2324, 2005.
- [14] Y. Zou and K. Chakrabarty, Sensor deployment and target localization based on virtual forces, *IEEE INFOCOM*, 2003.