Topology–Aware Placement and Role Assignment for Energy-Efficient Information Gathering in Sensor Networks[1,2]

Koustuv Dasgupta, Meghna Kukreja and Konstantinos Kalpakis

Abstract

Consider a network of energy–constrained wireless nodes that are capable of sensing and communication. These nodes are to be deployed over an area to be monitored. There is a set of points or regions of interest in that area, each of which need to be sensed (covered) by at least one node. The basic operation of this network is to periodically sense, gather, and transmit the sensed information to a concentration point. Nodes are allowed to perform in–network data aggregation, whereby multiple incoming data packets at a node can be aggregated into one single outgoing packet. Since the nodes have limited energy, it is desirable to maximize the system lifetime, i.e. the number of periods (rounds) that information can be collected until the first node is drained of its energy, while ensuring that the coverage requirements are satisfied. Since a node may cover more than one point/region of interest, we alow nodes to assume two roles, namely *sensor* and *relay*. Relay nodes aggregate and transmit data packets, while sensor nodes can also sense their vicinity and generate data packets. Since data transmission is a key energy consumer, the placement and role assignment of these nodes is critical.

We consider the problem of placing nodes in the monitoring area and assigning roles to them such that the system lifetime is maximized, while ensuring that each point/region of interest is covered by at least one sensor node. This is the *maximum lifetime sensor deployment problem with coverage constraints.* We present a novel algorithm to solve this problem. Our algorithm is motivated by force–directed/potential–field based approaches in robotics/graph drawing and is amenable to distributed implementation. We provide detailed experimental results demonstrating that our algorithm can offer significant improvements in system lifetime, when compared with random placement and role assignments of the nodes.

1 Introduction

Rapid advances in micro–sensor technology and novel sensing materials have stimulated great interest in the development of miniscule sensing devices with computing and communicaton capabilities. Such "smart" sensors (physical, chemical and biomedical) can be readily deployed in diverse environments to form distributed wireless networks and collect useful information in a robust and autonomous manner [12, 18, 19, 21]. Potential applications of smart sensors are extensive and range from military surveillance, habitat monitoring and terrain exploration to biomedical applications like glucose monitoring and retina prosthesis. However, there are several obstacles that need to be overcome before this vision becomes a reality. Such obstacles rise from the limited energy, computing capabilities and communication resources available to the sensors.

We consider a system of wireless nodes that are homogeneous and highly energy-constrained. Further, replenishing energy via replacing batteries on hundreds of nodes (in possibly harsh terrains) is infeasible. Each node produces some information as it monitors (senses) its vicinity. The basic operation in such a system is the systematic *gathering* of sensed data from one or more points of interest to be eventually transmitted to a base station for further processing. The key challenge in such information gathering is conserving the sensor energies, thereby maximizing the lifetime and hence the utility of the system.

[1] Supported in part by NASA under Cooperative Agreement NCC5–315 and Contract NAS5-32337.

[2] Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250. E-mail: {dasgupta,meg3,kalpakis}@csee.umbc.edu. Phone: 410-455-3143. Fax: 410-455-3969. **Please send all correspondence to Prof. Kalpakis.**

For increasing the lifetime of sensor networks, it is imperative that the node and the physical link are made as energy–efficient as possible [6]. Second, the collaborative strategy i.e. the strategy that governs how nodes co-operate to perform the sensing and gathering operations, must be energy–efficient as well. To this end, there are several power–aware routing protocols for wireless ad–hoc networks described in the literature [11, 23]. In the context of sensor networks, Tassiulas et al [3] describe near–optimal routing algorithms that maximize the time until the energies of the wireless nodes drain out. In recent developments, data fusion or *aggregation* has emerged as a basic tenet in sensor networks. The key idea is to combine data from different sensors to eliminate redundant transmissions, and provide a rich, multi–dimensional view of the environment being monitored. Krishnamachari et al [14] argue that this paradigm shifts the focus from address–centric approaches of finding routes between pairs of end nodes to a more data–centric approach of finding routes from multiple sources to a destination that allows in–network consolidation of data. Madden et al [16] discuss the implementation of five basic database aggregates, i.e. COUNT, MIN, MAX, SUM, and AVERAGE, based on the TinyOS platform and demonstrate that such a generic approach for aggregation leads to significant power (energy) savings. Directed diffusion [10] is based on a network of nodes that can co–ordinate to perform distributed sensing of an environmental phenomenon. This achieves significant energy savings when intermediate nodes aggregate responses to queries. The SPIN protocol [8] uses meta–data negotiations between sensors to eliminate redundant data transmissions through the network. In PEGASIS [15], sensors form chains so that each node transmits and receives from a nearby neighbor. Gathered data moves from node to node, gets aggregated and is eventually transmitted to the base station. Nodes take turns to transmit so that the average energy spent by each node gets reduced. In related work, Kalpakis et al [13] address the problem of finding an efficient schedule that specifies the manner in which data should be collected and aggregated from all sensors and transmitted to the base station. Given the location of the sensors and the base station, they provide a near–optimal Maximum Lifetime Data Aggregation (MLDA) algorithm to find a data gathering schedule that maximizes the time until the first sensor is drained of its energy. The proposed algorithm significantly outperforms existing data gathering protocols in terms of system lifetime.

A complementary approach to tackling the network longevity problem for data gathering applications is based on strategic sensor deployment with the objective of minimizing the energy consumption of the sensors. In general, a key challenge is to determine a placement of sensors that optimizes cost, provides high coverage of the surveillance zone, and minimizes sensor communications which is a key energy consumer. As an example, consider the millions of acres of land that are lost to forest fires every year. In most of the cases, early warnings are critical in preventing small fires from becoming disastrous infernos. Deploying smart sensors in strategically selected areas can lead to early detection and increased likelihood of success in fire extinguishing efforts. Other critical applications include biomedical sensors for human implants as well as smart sensors for pollution control, climate control in large buildings, traffic monitoring and congestion avoidance, and detection of biological and chemical weapons. In all of the above scenarios, given sufficient knowledge about the regions of interest, e.g. preferential coverage based on security, risk or tactical importance, the location of the base station(s), and the nature of the terrain, e.g. presence of obstacles, one can configure a *topology–aware* placement of the sensors such that the desired coverage is achieved with minimal energy consumption.

A number of recent investigations address the impact of topology–aware design on the performance of sensor networks. Krishnendu et al [5] offer a minimalistic view of distributed sensor networks wherein a minimum number of sensors are deployed to provide sufficient grid coverage of the sensor field. They point out that intelligent sensor placement decreases the need for excessive network communication for surveillance, target location and tracking. Srivastava et al [17] define the coverage problem from different points of view including deterministic, statistical, worst case and best case. In [24], the authors advocate organizing the sensors into mutually exclusive sets, where the members of each of the sets together completely cover the monitored area and only one of the sets is active at any time. Thus, activating only a subset of the sensors at any time can lead to significant energy savings. A critical observation was that the placement of sensors determine the maximum possible utilization of the available sensors. Cheng et al [4] restrict their investigation to an important class of wireless sensor networks such as biomedical sensor networks, in which the locations of the sensors are fixed and the placement can be pre-determined. They introduce the idea of relay sensors to maintain connectivity and study the impact of relays on a minimum power topology. Salhieh at al [20] rightly point out that much of the concentration has been on mobile networks rather than those with fixed node placement. They analyze the performance issues associated with different network topologies used for communication among the sensors.

Most of the previous work described above assume that the nodes are stationary and one can control the placement of sensors. The advent of mobile sensor networks, i.e. sensor nodes with locomotion capabilities, has led to the possibilty of self-organizing or self-deploying sensor networks. To this end, Sibley et al [22] describe the design of Robomote, a robot platform developed to explore problems in large-scale distributed sensor networks. Robomote occupies less than $0.000047m^3$, costs less than \$150 in parts and has functionality capabilities similar to its larger counterparts. Such an infrastructure is suitable for information gathering in harsh (hostile) terrains. Further, from the perspective of sensor placement, this implies that nodes in the network can potentially co-operate with each other and spread out to strategic locations that maximize coverage and/or minimize energy consumption. A recent work on mobile sensor networks [9] presents a distributed and scalable potential-field based approach for deployment of mobile sensors. Fields are constructed such that each node is repelled by obstacles as well as by other nodes, forcing the network to spread itself throughout the environment. Initial results show that the proposed method leads to high quality of coverage in the monitored area. Unfortunately, it is unclear how such a placement affects the lifetime of the sensor network.

In this paper, we study the impact of topology–aware placement on the lifetime of a sensor network. We consider a system of wireless nodes with sensing and communication capabilties that are distributed over a region. We assume that nodes can be of two types, *sensors* and *relays*. Sensors can sense their environment and generate data packets. Relays are nodes that do not sense their environment and thus they do not generate data packets themselves. Both types of nodes can trasmit (receive) data packets to (from) any other node or to the base station, as well as perform in–network data aggregation. The type of a node is referred to as its *role*. Given an area to be monitored, we assume that there is a set of points (regions), which we call *points (regions) of interest*, each of which must be sensed (covered) by at least one sensor. Further, given an initial placement of the nodes along with their roles, our objective is to find a final placement and role assignment for each node in the system such that the system lifetime is maximized, while preserving the coverage constraints of the points (regions) of interest. We refer to this problem as the *maximum lifetime sensor deployment problem with coverage constraints* and propose an iterative algorithm to solve it. To the best of our knowledge, this is the first algorithm that solves the problem of coverage preserving sensor deployment, with the objective of maximizing system lifetime. Our algorithm is motivated by force-directed/potential–field based approaches in robotics/graph drawing [1] and is easily amenable to a distributed implementation. Further, we provide detailed experimental results to demonstrate significant improvements in system lifetime attained by the proposed scheme, when compared to random placement and role assignments of the nodes.

The rest of the paper is organized as follows. In section 2, we describe the system model and formulate the sensor placement and role assignment problem for energy–efficient information gathering. Section 3 provides a detailed description of our algorithm. In section 4, we present experimental results and finally, in section 5, we conclude the paper.

2 Problem Formulation

Consider a network of n nodes $1, 2, \ldots, n$ and a base station B distributed over a region. The location of the base station is fixed and known apriori.

We assume that nodes can be of two types, *sensors* and *relays*. Sensors can sense (measure, observe) their environment and generate data packets. We assume that each sensor has a sensing radius r. Relays are nodes that do not sense their environment and thus they do not generate data packets themselves. Both types of nodes can trasmit (receive) data packets to (from) any other node or to the base station, as well as perform in–network aggregation. The type of a node is referred to as its *role*.

Each sensor produces some information as it monitors its vicinity. We assume that each sensor generates one data packet per time unit to be transmitted to the base station. For simplicity, we refer to each time unit as a *round*. We assume that all data packets have size l bits. The information from all the sensors needs to be gathered at each round and sent to the base station for processing. We assume that each node has the ability to transmit its packet to any other node in the network or directly to the base station. Further, each node i has a battery with finite, non-replenishable energy $\mathcal{E}_i$. Whenever a node transmits or receives a data packet, it consumes some energy from its battery. The base station has an unlimited amount of energy available to it. We define the *lifetime* of a sensor network to be the number of rounds until the first node is drained of its energy.

Our energy model for the nodes is based on the *first order radio model* described in [7]. A node consumes $\epsilon_{elec} = 50nJ/bit$ to run the transmitter or receiver circuitry and $\epsilon_{amp} = 100pJ/bit/m^2$ for the transmitter amplifier. Thus, the energy consumed by a node i in receiving a l-bit data packet is given by,

$$R\text{x}_i = \epsilon_{elec} \cdot l \tag{1}$$

while the energy consumed in transmitting a data packet to node j is given by,

$$T\text{x}_{i,j} = \epsilon_{elec} \cdot l + \epsilon_{amp} \cdot d^2_{i,j} \cdot l \tag{2}$$

where $d_{i,j}$ is the distance between nodes i and j.

Given an area to be monitored, we assume that there is a set of points (regions), which we call *points (regions) of interest*, that must be sensed (covered) by at least one node. A sensor p is said to cover a point of interest q, if and only if, q lies within the sensing radius of p. Similarly, a sensor p is said to cover a region of interest $\mathcal{Q}$, is and only if, the entire region $\mathcal{Q}$ lies within the sensing radius of p. For the sake of simplicity, we discuss our algorithm for points of interest. We assume that each point of interest has at least one node within distance r from it. Initially, roles are assigned to nodes at random, ensuring that each point of interest has at least one sensor node within distance r from it. The role of a node may change, provided that coverage of all points of interest is maintaned.

Given the location of the base station and an initial placement of the nodes along with an initial role assignment, our objective is to find a placement and role assignment of the nodes such that the system lifetime is maximized, while ensuring that all points of interest are covered by at least one sensor node. This is the *maximum lifetime sensor deployment problem with coverage constraints.*

3 SPRING: Sensor Placement and Role assignment for energy–efficient INformation Gathering

In this section, we describe the SPRING algorithm (Sensor Placement and Role assignment for energy–efficient INformation Gathering) to solve the maximum lifetime sensor deployment problem with coverage constraints.

Before, we present our algorithm we need a few additional definitions. Let k be a paramater to be determined later. Given a placement of the nodes, we associate with each node p

- the set $D(p)$ of its downstream neighbors defined by $D(p) = \{\ u \mid d(u, B) \leq (p, B)\ \}$, where $d(u, v)$ is the (Euclidean) distance between u and v.
- the set $DNN_k(p)$ consisting of its *k–nearest donwstream neighbors.* If the a node has fewer than k downstream neighbors, then $DNN_k(p) = D(p)$.

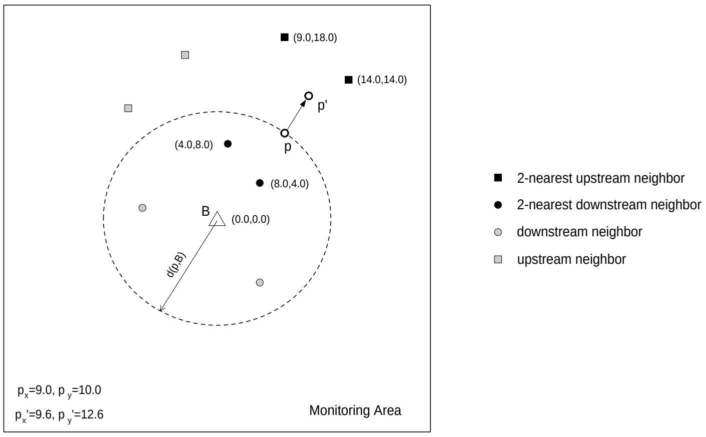


Figure 1: Node p with its upstream, downstream, 2–nearest upstream and 2–nearest downstream neighbors. The base station B is located at $(0.0, 0.0)$. $d(p, B)$ denotes the distance of node p from the base station B. The initial location of p is at $(9.0, 12.0)$. The target location $(9.6, 12.6)$ of p is computed using equations (4) and (5), with respect to the locations of its 2–nearest downstream and 2–nearest upstream neighbors.

- the set $U(p)$ of its *upstream neighbors* defined by $U(p) = \{\ u \mid d(u, B) > d(p, B)\ \}$.
- the set $UNN_k(p)$ of its *k–nearest upstream neighbors*, which consists of at most k nearest upstream neighbors of p, that have p among their k–nearest downstream neigbhors.

The *candidate nodeset* $\Lambda(q)$ of a point of interest q is defined by the set of nodes that are within distance r from q, i.e. $\Lambda(q) = \{\ u \mid d(u, q) \leq r\}$. To each point of interest q we assign a sensor node $\lambda(q) \in \Lambda(q)$ that covers q.

Our objective is to find a placement of the nodes and role assignment so that the system lifetime is maximized. To this end, we would like to minimize the average energy consumed by a node over all the rounds.

Consider a node p whose location is (p_x, p_y). Suppose that p transmits data packets only to its k–nearest donwstream neighbors, in a round robin manner. Then, node p receives packets only from its upstream neighbors that have p in their k–nearest downstream neigbhors. In to decrease the the enrgey consumption, one may want to move p closer to its k–downstream neigbhors. Doing so however, increases the energy consumed by its upstream neigbhors that transmit to it. Therefore, we advocate that node p should be placed at a location that minimizes

$$F(p) = \frac{\sum_{u \epsilon DNN_k(p)} d(p, u)^2}{k} + \sum_{v \epsilon UNN_k(p)} d(v, p)^2. \tag{3}$$

Note that, doing so may change the upstream and downstream neigbhors of p as well as of other nodes. Let us put this aside for a moment. The function $F(p)$ is minimized when p is placed at a location for which $\partial F(p)/\partial p_x = 0$ and $\partial F(p)/\partial p_y = 0$, which leads to the target location

$$p'_x = \frac{\frac{\sum_{u \epsilon DNN_k}(u_x - p_x)}{|DNN_k|} + \sum_{u \epsilon UNN_k} u_x}{1 + |UNN_k|} \tag{4}$$

$$p'_y = \frac{\frac{\sum_{u \epsilon DNN_k}(u_y - p_y)}{|DNN_k|} + \sum_{u \epsilon UNN_k} u_y}{1 + |UNN_k|} \tag{5}$$

```
Algorithm SPRING
for $i = 1$ to MaxIterations do
    Sort the nodes in decreasing order of the number of points of interest assigned to them
    for each node $p$ do
        call procedure Move($p$)
    if no node has moved by more than $\epsilon$ from its previous location then
        return
return

Procedure Move(node $p$)
Compute the $k$–nearest upstream and downstream neighbors of $p$
Compute the target location $(p'_x, p'_y)$ of $p$ using Eqs. 4 and 5.
Compute the direction vector $\vec{v}$
for each point of interest $q$ covered by $p$ do
    // $p$ is a sensor node
    if there exists a node $u \neq p$ in the candidate set $\Lambda(q)$ then
        assign $u$ to $q$ and set $u$'s role to sensor
if $p$ is not assigned to any point of interest then
    move $p$ by $\delta$ along the direction vector $\vec{v}$
    set $p$'s role to relay
else
    move $p$ along $\vec{v}$ by at most $\delta$ as long as
        $d(p, q) \leq r$ for all $q$ still assigned to $p$
    set $p$'s role to sensor
return
```

Figure 2: Pseudocode for placement and role assignment of nodes using the SPRING algorithm.

Figure 1 provides an illustration of our approach for finding the placement of a node.

Let $\vec{v}$ be the vector that starts at (p_x, p_y) and ends at (p'_x, p'_y). Once, the target location of p is calculated, we could move p to that location. However, this may lead into undesirable effects (e.g. oscillation, etc). Thus, we move the node p along the direction of the vector $\vec{v}$ defined by its current and target location by a small amount δ. At the same time, recall that we also want to ensure that all points of interest are still covered. Moving node p may lead into a violation of that constraint. In some cases, we may not be able to move node p at all, e.g. if p is the only sensor node within distance r of some point of interest. Thus, in addition to the target location of p, we need to consider p's role with respect to its current and target locations.

If node p is a relay node, then p can freely move to its new location and retain its relay role at that new location. Otherwise, p is a sensor node. In this case, for each point of interest q assigned to p, we attempt to assign another node u to q, among those in the candidate set $\sigma(q)$ of q, and set u's role to be a sensor. Eventually, if p is not assigned to any points of interest, we set its role to relay and move it along the direction $\vec{v}$ by δ. Otherwise, p remains a sensor, and we move p along the direction $\vec{v}$ at most δ, such that p is within distance r from all points of interest still assigned to p.

We repeat the above procedure for each node of the system, until either the locations of the nodes do not change significantly or a predefined number of iterations have been performed. Fig 2 give the pseudocode for the SPRING algorithm. In the next section, we use the MLDA data gathering and aggregation protocol [13] to measure the system lifetime achieved by the placement and role assignment computed by SPRING.

4 Performance Evaluation

In this section, we evaluate the performance of the SPRING algorithm. Our objective is to analyze the impact of intelligent placement and role assignment on the lifetime of a sensor network, where the basic operation of the network is the systematic gathering (and aggregation) of information from an area of surveillance and transmitting it to a base station. To completely specify our experimental procedure and understand the comparative gains delivered by SPRING, we next discuss (a) the data gathering protocol used in our experiments, and (b) an alternative strategy for placement and role assignment of the nodes.

Recall that the lifetime of the sensor network is defined as the number of rounds until the first sensor is drained of its energy. For a particular placement and role assignment of the nodes, we measure the lifetime of the network under that particular placement and role assignment, obtained using a suitable data gathering protocol. We use the MLDA algorithm described in the literature [13] as our candidate protocol. [3] For the sake of completeness, we next include a brief description of the data gathering protocol.

4.1 MLDA : Maximum Lifetime Data Gathering with Aggregation

Given a collection of nodes $1, 2, \ldots, n$ and a base station labelled $n + 1$, together with their locations and the energy of each node, the MLDA algorithm finds a *maximum lifetime data gathering schedule.* Such a schedule specifies how data should be collected, aggregated and transmitted in each round of data gathering so that the time (number of rounds) until the first node is drained of its energy is maximized. Observe that each node is assigned to be a sensor or a relay. Further, each node is allowed to aggregate (fuse) multiple incoming packets into one single outgoing packet.

Consider a schedule $\mathcal{S}$ with lifetime T rounds. Let $f_{i,j}$ be the total number of packets that node i (sensor or relay) transmits to another node j (or base station) in S. Since any valid schedule must respect the energy constraints at each node, it follows that for each node $i = 1, 2, \ldots, n$,

$$\sum_{j=1}^{n+1} f_{i,j} \cdot T\textsc{x}_{i,j} + \sum_{j=1}^{n} f_{j,i} \cdot R\textsc{x}_i \leq \mathcal{E}_i. \tag{6}$$

Recall that each sensor, for each one of the T rounds, generates one data packet that needs to be collected, possibly aggregated, and eventually transmitted to the base station. Relays do not sense their environment and hence do not generate data packets.

The schedule $\mathcal{S}$ induces a flow network $G = (V, E)$. The flow network G is a directed graph having as vertices all the nodes and the base station, and having edges (i, j) with capacity $f_{i,j}$ whenever $f_{i,j} > 0$.

Theorem 1 *Let $\mathcal{S}$ be a schedule, and let G be the flow network induced by $\mathcal{S}$. Then, $\mathcal{S}$ has lifetime T if and only if, for each sensor s, the maximum flow from s to the base station t in G is $\geq T$.*

Proof. Refer to [13] for details. ■

Thus, a necessary and sufficient condition for a schedule to have lifetime T is that each sensor in the induced flow network can push flow T to the base station t. Stated otherwise, each sensor s must have a minimum $s - t$ cut of capacity (size) $\geq T$ to the base station. Next, we consider the problem of finding a flow network G with maximum T, that allows each sensor to push flow T to the base station, while respecting the energy constraints in (8) at all the nodes. We call such a flow network G an *admissible* flow network with lifetime T. An admissible flow network with maximum lifetime is called an *optimal admissible* flow network. Clearly, we need to find the capacities of the edges in G.

An optimal admissible flow network can be found using an integer program with linear constraints. The integer program, in addition to the variables for the lifetime T and the edge capacities $f_{i,j}$, uses the following variables: for each node $k = 1, 2, \ldots, n$, let $\pi_{i,j}^{(k)}$ be a flow variable indicating the flow that k sends to the base station t over the edge (i, j). The integer program computes the maximum system lifetime T subject to the energy constraint (8) and the additional linear constraints (9)–(13) for each sensor, as shown in Table 1. Constraint (9) ensures that the capacity constraints on the edges of the flow network are respected. For each relay $k = 1, 2, \ldots, n$, constraint (10) enforces the flow conservation principle at the relay. For each sensor

[3] Note that the SPRING algorithm is independent of the data gathering protocol used. We choose MLDA since it significantly outperforms other data gathering protocols in terms of system lifetime [13].

Objective :

$$\text{maximize } T \tag{7}$$

Energy constraint for a node:

$$\sum_{j=1}^{n+1} f_{i,j} \cdot T\mathrm{X}_{i,j} + \sum_{j=1}^{n} f_{j,i} \cdot R\mathrm{X}_i \leq \mathcal{E}_i. \tag{8}$$

Capacity constraint for each edge:

$$0 \leq \pi_{i,j}^{(k)} \leq f_{i,j}, \quad \forall i = 1, 2, \ldots, n \text{ and } \forall j = 1, 2, \ldots, n+1 \tag{9}$$

Flow conservation constraint for a relay k:

$$\sum_{j=1}^{n} \pi_{j,i}^{(k)} = \sum_{j=1}^{n+1} \pi_{i,j}^{(k)}, \quad \forall i = 1, 2, \ldots, n \tag{10}$$

Flow conservation constraints for a sensor k :

$$\sum_{j=1}^{n} \pi_{j,i}^{(k)} = \sum_{j=1}^{n+1} \pi_{i,j}^{(k)}, \quad \forall i = 1, 2, \ldots, n \text{ and } i \neq k \tag{11}$$

$$T + \sum_{j=1}^{n} \pi_{j,k}^{(k)} = \sum_{j=1}^{n+1} \pi_{k,j}^{(k)} \tag{12}$$

$$\sum_{i=1}^{n} \pi_{i,n+1}^{(k)} = T \tag{13}$$

where $k = 1, 2, \ldots, n$ and all variables T, $f_{i,j}$, and $\pi_{i,j}^{(k)}$ are required to be non-negative integers.

Table 1: Integer program for finding an optimal admissible flow network.

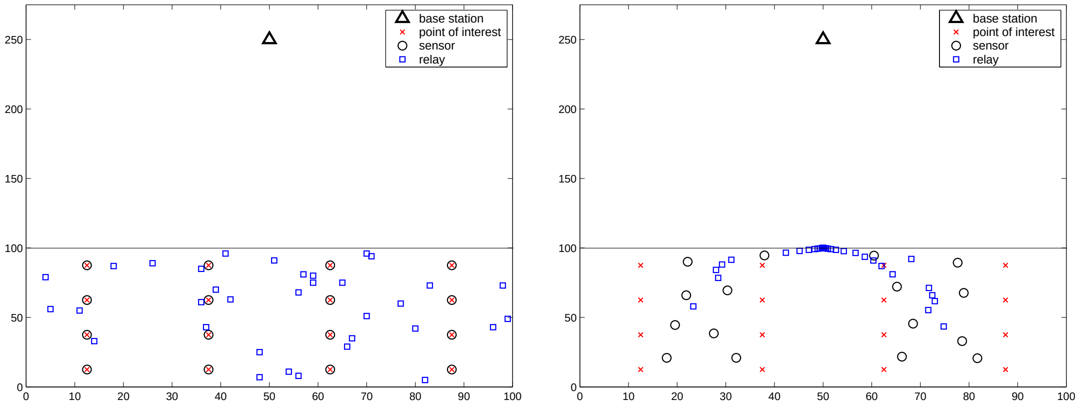
$k = 1, 2, \ldots, n$, constraints (11) and (12) enforce the flow conservation principle at the sensor. In addition, constraint (13) ensures that T flow from sensor k reaches the base station. Moreover, constraint (8) is used to guarantee that the edge capacities of the flow network respect the node's available energy. Finally, for the integer program, all variables are required to take non–negative integer values. The linear relaxation of the above integer program, i.e. when all the variables are allowed to take fractional values, can be computed in polynomial–time. Then, we can obtain a good approximation for the optimal admissible flow network by first fixing the edge capacities to the floor of their values obtained from the linear relaxation so that the energy constrains are all satisfied; and then solving the linear program (7) subject to constraints (9)–(13) without requiring anymore that the flows are integers (since a solution with integer flows can be found). [4]

Observe that a schedule is a collection of directed trees rooted at the base station that span all the sensors, with one such tree for each round. Each such tree specifies how data packets are gathered and transmitted to the base station. We call these trees *aggregation trees*. An aggregation tree may be used for one or more rounds; we indicate the number of rounds f, that an aggregation tree is used, by associating the value f with each one of its edges; we call f to be the lifetime of the aggregation tree. An aggregation tree A with lifetime $f \leq T$ can be obtained from an admissible flow network G with lifetime T, using the polynomial–time GetTree algorithm described in [13]. Further, we can compute a collection of aggregation trees from an admissible flow network G with lifetime T, by repeatedly using the GetTree algorithm until T data packets from each of the sensors are aggregated and transmitted to the base station t.

4.2 Experimental Results

Using the data gathering algorithm described above, we compare the system lifetime achieved by spring–based placement and role assignment with a random approach, whereby the nodes are placed randomly in

[4] Experimental results in [13] demonstrate that the lifetime obtained by this approximation is near–optimal.



(a) RANDOM

(b) SPRING

Figure 3: Sensor placement and role assignment using the RANDOM and SPRING approach for a regular distribution of points of interest.

the area to be monitored and are assigned random roles, while ensuring that each point (region) of interest has at least one sensor node covering it. We report our findings on two different sets of experiments.

For the initial set of experimental results, we consider a network of nodes randomly distributed in a 100m × 100m monitoring field. The number of nodes in the network, i.e. the network size n, is varied to be 10, 20, 30, 40, 50, 60, 80 and 100 respectively. Each sensor has an initial energy of 1J and the base station is located at (50m, 250m). Each node has a sensing radius of 10m and generates packets of size 1000 bits . The energy model for the nodes is based on the first order radio model described in section 2.

Given the field to be monitored, we model preferential coverage (e.g. based on security or tactical importance) by assigning some points of interest within the field. We consider two different cases – (i) when the points of interest are randomly distributed, (ii) when the points of interest are distributed at regular intervals. In the latter case, we can imagine the sensor field to be represented as a (two–dimensional) grid of points. We denote $\mathcal{N}$ to be the number of points of interest. For each experiment, we start with an initial configuration where nodes are placed at random locations in the monitoring field and are assigned roles randomly, while ensuring that each point of interest is covered (sensed) by at least one sensor. We refer to this approach as the RANDOM approach. Subsequently, we compute a placement and role assignment of the nodes using the SPRING algorithm. [5] Figure 3 provides an illustration of SPRING– based placement and role assignment for a network of 50 nodes and 16 points of interest in a 100m × 100m field. It is interesting to observe how the sensors and relays adust their locations (using SPRING) in a topology–aware manner.

For each of the configurations given by RANDOM and SPRING, we measure the number of sensors S that cover one or more points of interest, and the system lifetime T obtained using the MLDA data gathering protocol. We define the *percentage gain* G to be the (percentage) improvement in the system lifetime achieved by SPRING–based placement and role assignment, when compared to the RANDOM approach. Tables 2 and 3 show our main results averaged across 10 experiments for each network size. We make the following observations.

- For a random distribution of points of interest, the lifetime of a data gathering schedule obtained using the SPRING algorithm is 27.65% to 49.48% better than the lifetime attained using a RANDOM approach. In fact, SPRING does a superior job of intelligent role assignment by using a fewer number of sensors to cover the points of interest. Once coverage is guaranteed, it allows more nodes (relays) to freely position themeselves at strategic locations that are beneficial for aggregating the sensed information and transmitting this information to the base station.

[5] The value of k was set to 3 for all our experiments.

INPUT		RANDOM		SPRING		G
n	$\mathcal{N}$	S	T	S	T	
10	4	4	1726	4	2289	32.62
20	9	9	1971	9	2516	27.65
30	9	9	2012	8	2875	42.89
40	16	16	2156	12	3134	45.36
50	16	16	2056	14	2911	41.59
60	16	16	2499	12	3635	45.46
80	25	25	2384	24	3472	45.64
100	25	25	2684	21	4012	49.48

Table 2: Performance measures for a random distribution of points of interest

INPUT		RANDOM		SPRING		G
n	$\mathcal{N}$	S	T	S	T	
10	4	4	1842	4	2240	21.61
20	9	9	1808	9	2189	21.07
30	9	9	2145	8	2672	24.57
40	16	16	2317	13	2985	28.83
50	16	16	2099	12	2866	36.54
60	16	16	2417	14	3201	32.44
80	25	25	2620	22	3526	34.58
100	25	25	2802	22	3897	39.08

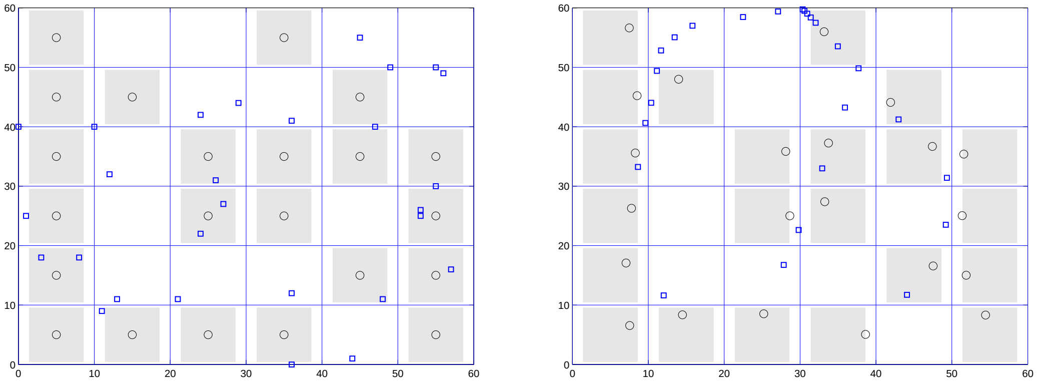
Table 3: Performance measures for a regular distribution of points of interest

INPUT		RANDOM		SPRING		G
n	$\mathcal{N}$	S	T	S	T	
40	27	27	5986	27	7944	32.71
50	27	27	6223	25	8365	34.42
60	27	27	6489	24	8909	37.29
80	27	27	6651	24	9386	41.12
100	27	27	6423	25	9728	41.46

Table 4: Performance measures for a random distribution of regions of interest

- For a regular distribution of points of interest, the SPRING algorithm improves the lifetime of a data gathering schedule by as much as a factor of 39.08% when compared to the RANDOM approach. Once again, SPRING does a superior job of intelligent role assignment by using a fewer number of sensors to cover the points of interest, thereby allowing more nodes (relays) to position themeselves at strategic locations that are beneficial for data gathering and aggregation.

For the second set of experiments, we consider a 60m × 60m field with the base station located at (25m,150m). The number of nodes in the network, i.e. the network size n, is varied to be 40, 50, 60, 80 and 100 respectively. The initial energies, sensing radii, and energy model of the nodes remain unchanged. For this set of experiments, we divide the field into 10m × 10m cells (regions). Intuitively, the choice of the cell resolution is dependent on the quality of coverage desired. Next, we pick random cells and assign them to be the regions of interest. Recall that a region is covered by a sensor, if and only if, it lies entirely within the sensing radius of the sensor. We denote $\mathcal{N}$ to be the number of regions of interest. We define the *percentage coverage* to be the number of regions of interest as a percentage of the total number of regions in the field.



(a) RANDOM

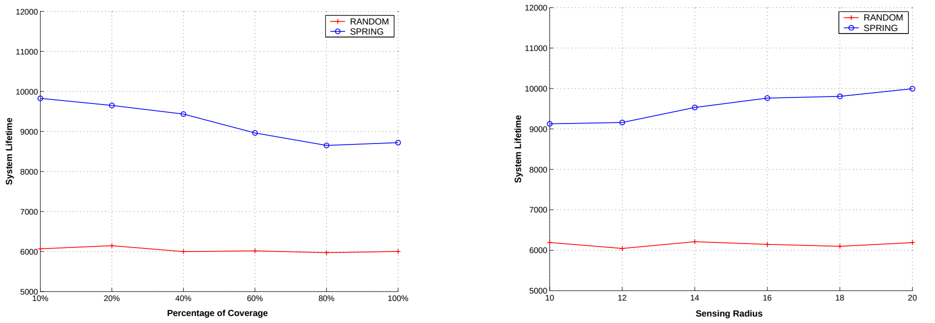
(b) SPRING

Figure 4: Sensor placement and role assignment using the RANDOM and SPRING approach for a random distribution of regions of interest. The shaded regions denote the regions of interest. The base station is located at (25m,150m) and is not shown in the plot. Circles denote sensors and squares denote relays.

As in the previous case, we start with an initial configuration where nodes are placed at random locations in the monitoring field and assigned roles randomly, while ensuring that each region of interest is covered (sensed) by at least one sensor. We refer to this approach as the RANDOM approach. Subsequently, we compute a placement and role assignment of the nodes using the SPRING algorithm. Figure 4 provides an illustration of SPRING– based placement and role assignment for a network of 50 nodes and 60% coverage in a 60m × 60m field. Observe how the sensors and relays adust their locations (using SPRING) in a topology–aware manner.

Table 4 shows our main results for 75% coverage (i.e. randomly assigned 27 regions of interest) averaged across 10 experiments for each network size. For each of the configurations given by RANDOM and SPRING, we measure the number of sensors S that cover one or more regions of interest, and the system lifetime T obtained using the MLDA data gathering protocol. The *percentage gain* G refers to the (percentage) improvement in the system lifetime achieved by SPRING–based placement and role assignment, when compared to the RANDOM approach. We observe that the SPRING algorithm improves the lifetime of a data gathering schedule by as much as a factor of 41.46% when compared to the RANDOM approach. This improvement can be attributed to the intelligent role assignment and strategic position of the sensors and relays, such that the energy consumption of the nodes (during the process of data gathering) is minimized.

Figures 5(a) and 5(b) explore the impact of the percentage of desired coverage and the sensing radius respectively, on the performance of the RANDOM approach and the SPRING algorithm. We observe that SPRING–based placement and role assignment always outperforms the RANDOM approach. Moreover, from figure 5(a) we witness that the lifetime obtained by SPRING decreases with increasing percentage of coverage. The result sheds light on the tradeoff associated with coverage and lifetime i.e. increased coverage comes at the cost of reduced system lifetime. Finally, figure 5(b) shows that the SPRING–based role assignment adjusts efficiently with increased sensing radius i.e. it uses fewer sensors to cover the regions of interest and allows more relays to be freely positioned in the field, thereby leading to an increase in the system lifetime attained by the data gathering protocol. As expected, the performance of the RANDOM approach is independent of the sensing radius (or coverage desired) and is significantly worse in terms of system lifetime.



(a) Lifetime with varying percentage of coverage (b) Lifetime with varying sensing radius

Figure 5: Impact of percentage of coverage and sensing radius on system lifetime

5 Conclusions and Future Directions

In this paper, we proposed a novel algorithm for topology–aware placement and role assignment of nodes in a sensor network, where the basic operation is the systematic gathering of information from an area of interest and transmitting this information to a base station. We assume that nodes are allowed to perform in–network data aggregation. The proposed algorithm is shown to deliver significant improvements in the lifetime of a sensor network, when compared to random placement and role assignments of the nodes. This is the first algorithm that solves the problem of coverage preserving sensor deployment, with the objective of maximizing system lifetime.

In the case of stationary sensor networks, where one can control the placement of the nodes, we can use SPRING to compute the positions and the role of each node apriori at a centralized point of control (base station). With the advances in mobile sensor networks [22] and recent work on localization in sensor networks [2], it is possible to implement our algorithm in a distributed manner. Each node simply needs to know its location with respect to the locations of a few neighboring nodes (periodically) and can compute its new position (and role) based on this information. There are quite a few directions in which we want to extend our research. In this paper, we have assumed that the coverage depends only on the Euclidean distances from the sensors. In future, we want to consider other factors like obstacles, environmental conditions, and noise (imprecise detection) in the placement and role assignment algorithm. Further, we wish to explore the tradeoffs associated with coverage and lifetime in greater detail, in order to come up with more practical solutions for sensor networks.

References

[1] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, 1999.

[2] N. Bulusu, J. Heidemann, and D. Estrin. Adaptive Beacon Placementies Over Ad-Hoc Wireless Sensor Networks. In *Proceedings of 21st IEEE International Conference on Distributed Computing Systems*, 2001.

[3] J.H. Chang and L. Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In *Proceedings of IEEE INFOCOM*, 2000.

[4] X. Cheng, D. Z. Du, L. Wang, and B. Xu. Relay Sensor Placement in Wireless Sensor Networks. In *IEEE Transactions on Computers*, 2001.

[5] S. S. Dhillon, K. Chakrabarty, and S.S. Iyengar. Sensor Placement for Grid Coverage under Imprecise Detections. In *Proceedings of International Conference on Information Fusion*, 2002.

[6] A. Chandrakasan et al. Design Considerations for Distributed Microsensor Systems. In *Proceedings of IEEE Custom Integrated Circuits Conference*, 1999.

[7] W. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocols for Wireless Microsensor Networks. In *Proceedings of Hawaiian International Conference on Systems Science*, 2000.

[8] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proceedings of 5th ACM/IEEE Mobicom Conference*, 1999.

[9] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile Sensor Network Deployment usinf Potential Fields: A Distributed,Scalable Solution to the Area Coverage Problem. In *Proceedings of 6th International Symposium on Distributed Autonomous Robotics Systems*, 2002.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of 6th ACM/IEEE Mobicom Conference*, 2000.

[11] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, 1996.

[12] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Mobile Networking for Smart Dust. In *Proceedings of 5th ACM/IEEE Mobicom Conference*, 1999.

[13] K. Kalpakis, K. Dasgupta, and P. Namjoshi. Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks. In *Proceedings of IEEE International Conference on Networking*, 2002.

[14] B. Krishnamachari, D. Estrin, and S. Wicker. Modelling Data-Centric Routing in Wireless Sensor Networks. In *Proceedings of IEEE Infocom*, 2002.

[15] S. Lindsey and C. S. Raghavendra. PEGASIS: Power Efficient GAthering in Sensor Information Systems. In *Proceedings of IEEE Aerospace Conference*, 2002.

[16] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler. Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks. In *Proceedings of 4th IEEE Workshop on Mobile Computing and Systems Applications*, 2002.

[17] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage Problems in Wireless Ad-hoc Sensor Networks. In *Proceedings of IEEE INFOCOM*, 2001.

[18] R. Min, M. Bhardwaj, S.H. Cho, A. Sinha, E. Shih, A. Wang, and A.P. Chandrakasan. Low-Power Wireless Sensor Networks. In *VLSI Design*, 2001.

[19] J. Rabaey, J. Ammer, J.L. da Silva Jr, and D. Patel. PicoRadio: Ad-hoc Wireless Networking of Ubiquitous Low-Energy Sensor/Monitor Nodes. In *Proceedings of the IEEE Computer Society Annual Workshop on VLSI*, 2000.

[20] A. Salhieh, J. Weinmann, M. Kochhal, and L. Schwiebert. Power Efficient Topologies for Wireless Sensor Networks. In *Proceedings of International Conference on Parallel Processing (ICPP)*, 2001.

[21] L. Schwiebert, S. Gupta, and J. Weinmann. Research Challenges in Wireless Networks of Biomedical Sensors. In *Proceedings of ACM SIGMOBILE*, 2001.

[22] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme. Robomote: A Tiny Mobile Robot Platform for Large-Scale Ad-hoc Sensor Networks. In *Proceedings of International Conference on Robotics and Automation*, 2002.

[23] S. Singh and M. Woo. Power–aware Routing in Mobile Ad hoc Networks. In *Proceedings of 4th ACM/IEEE Mobicom Conference*, 1998.

[24] S. Slijepcevic and M. Potkonjak. Power Efficient Organization of Wireless Sensor Networks. In *Proceedings of IEEE International Conference on Communications*, 2001.