22 Aug 2005

Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks

Maggie Xiaoyan Cheng
Missouri University of Science and Technology, chengm@mst.edu

Lu Ruan

Weili Wu

Recommended Citation

M. X. Cheng et al., "Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks," *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 2005*, Institute of Electrical and Electronics Engineers (IEEE), Aug 2005.
The definitive version is available at https://doi.org/10.1109/INFCOM.2005.1498547

Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks

Maggie X. Cheng Lu Ruan Weili Wu

Abstract— **This paper addresses the coverage breach problem in wireless sensor networks with limited bandwidths. In wireless sensor networks, sensor nodes are powered by batteries. To make efficient use of battery energy is critical to sensor network lifetimes. When targets are redundantly covered by multiple sensors, especially in stochastically deployed sensor networks, it is possible to save battery energy by organizing sensors into mutually exclusive subsets and alternatively activating only one subset at any time. Active nodes are responsible for sensing, computing and communicating. While the coverage of each subset is an important metric for sensor organization, the size of each subset also plays an important role in sensor network performance because when active sensors periodically send data to base stations, contention for channel access must be considered. The number of available channels imposes a limit on the cardinality of each subset. Coverage breach happens when a subset of sensors cannot completely cover all the targets. To make efficient use of both energy and bandwidth with a minimum coverage breach is the goal of sensor network design.**

This paper presents the minimum breach problem using a mathematical model, studies the computational complexity of the problem, and provides two approximate heuristics. Effects of increasing the number of channels and increasing the number of sensors on sensor network coverage are studied through numerical simulations. Overall, the simulation results reveal that when the number of sensors increases, network lifetimes can be improved without loss of network coverage if there is no bandwidth constraint; with bandwidth constraints, network lifetimes may be improved further at the cost of coverage breach.

Keywords: Mathematical programming/optimization, Combinatorics, sensor networks, coverage breach, set cover, scheduling, bandwidth, energy conservation.

I. INTRODUCTION

Sensor networks have been used in remote or inhospitable environments for data gathering and will be widely used in diverse environments in the future. A sensor network consists of a large number of battery-powered devices with sensing, computing, and wireless communication capabilities. Sensors in a network can cooperatively gather information from a specified region of observation and transmit this information to the base station. Due to the limited resources in battery energy and radio spectrum, the capacity of wireless sensor networks is often limited by energy and bandwidth constraints.

Department of Computer Science, University of Missouri, Rolla, MO 65409-0350. E-mail: chengm@umr.edu

Department of Computer Science, Iowa State University, Ames, IA 50011. E-mail: ruan@cs.iastate.edu

Dept of Computer Science, University of Texas at Dallas, Richardson, TX 75083. E-mail: weiliwu@utdallas.edu

For a stochastically deployed sensor network, the number of sensors deployed is usually higher than that of its deterministically deployed counterpart. Some targets are redundantly covered as a result. Redundancy can be used to reduce each individual sensor's sensing, computing and communication activities. If it is possible to turn off some of the sensors with the remaining sensors providing satisfying coverage, and to switch the sensors between active mode and inactive mode, the sensor network can last for a longer time. Moreover, to be energy-efficient, the sensor nodes need to stay in a low-power mode for over a certain threshold, the longer the better [1]. On the other hand, the battery lifetime is also extended if it frequently oscillates between active modes and inactive modes. The battery lifetime is twice as much if it is discharged in short bursts with significant off time as in a continuous mode of operation [2].

Network lifetime has been an important factor in sensor network design. To extend sensor network lifetime, one potential approach is to use *disjoint covers*. In this approach, sensors are divided into mutually exclusive subsets without consideration on subset sizes; each subset is switched to active mode and sleep mode alternatively, so that at any time there is only one set of sensors active and the active sensors together can cover all targets. When sensors are divided into mutually exclusive subsets, the number of subsets that can be constructed from the original sensors is critical to network lifetime. By maximizing the number of subsets, the sensor network lifetime can be extended significantly.

However, there is one major potential problem with the *disjoint cover* approach because the size of each subset is not restricted. The ultimate goal of sensor networks is for the observer (usually at the base station) to access the sensory data timely and completely. So eventually, each active sensor will send the sensory data to the base station, which requires that there must be sufficient bandwidths for this activity. Here "bandwidth" could be the total number of time slots if a time division scheme is used on a single shared channel, or the total number of channels if multiple channels are available.

In this paper, we assume a very simple scenario, i.e., every sensor ships its sensory data directly to the base station. So the total number of sensors simultaneously sending to the base station must be restricted by the bandwidth. With bandwidth constraints in sensor networks, complete coverage in each subset is no longer an indicator of timely and complete data access if the subset sizes are not restricted. If there are W channels available, and there are more than W sensors in some subset, while every sensor in the subset is active in sensing and computing, some sensors can not have channel access for data transmission; if a single shared channel is used on the other hand, and there are W time slots available in each cycle, some active sensors that are sensing and computing won't have chance to report their sensory data in every cycle, therefore have to delay the data transmission, which results in latency in observing events at the base station. From the information access point of view, there is no difference between the failure in sensing and the failure in reporting.

One solution to combat the limited bandwidth problem is to make sensors aggregate sensory data before transmitting it to the base station, so only a few designated aggregators will transmit to the base station. The drawback of this approach is that it introduces extra delay for information aggregation from peer sensor nodes, and increases channel contention because part of the radio spectrum is dedicated to peer communication among sensor nodes. Another solution without pre-aggregation is the joint optimization on energy and bandwidth utilization: considering the bandwidth constraints when the sensors are divided into subsets. Specifically, to make efficient use of bandwidth, sensors need to be organized into subsets of bounded size (i.e., $\leq W$), so sensors in each subset can transmit its sensory data to the base station without delay. Subsets are turned on and off alternatively to conserve energy. By this way, events can be detected and reported to the base station timely. To allocate time slots or channels to sensors, a proper scheduling techniques must be used so that the sensors in each subset can satisfy the coverage requirement while being fully restricted by the bandwidth constraints. If a target (or monitor region) is not covered by any active sensor, it is called "breached". The objective of this joint optimization is to minimize the total breach of all targets.

In this paper, a mathematical model of the minimum breach problem is developed, the computational complexity of the problem is analyzed, and two approximate algorithms are presented. Performance of the heuristics are compared via simulations. The effects of increasing the number of sensors on network coverage in bandwidth constrained networks are studied. These simulation results demonstrate that to improve the coverage performance in wireless sensor networks, bandwidths also need to be increased; in bandwidth constrained networks, increasing the number of sensors alone do not always improve coverage results.

Sensor networks are application-specific. It is not likely that the network protocols designed for one application can be applied to all applications without tailoring. The target application of this paper is for a very simple scenario: sensor nodes have the same communication, computing and sensing capability; each active node periodically reports to the base station directly; all sensor nodes together perform a high-level sensing task. A typical application is ambient condition monitoring or target tracking described in [3]. The information that is interested by the base station is: "what is happening in region #2?" or "where is target #2 ?" rather than "what is the data collected by sensor node #4?". Individual sensors can be off duty as long as other sensors provide a satisfying coverage.

The rest of the paper is organized as follows: in section II, we list some of the related works; in section III, we present a formal definition of the minimum breach problem, and prove that to compute a set of disjoint subsets with minimum breach is NP-complete; in section IV, we develop a 0-1 integer programming model, and present two heuristics based on this model; in section V, we provide a performance comparison of the two heuristics; section VI ends this paper with conclusions and extensions for future work.

II. Related Work

Although much work has been done to extend sensor network lifetimes through power aware self-organization, to our knowledge, this is the first effort to consider data transmission bandwidth constraints when dividing tasks among sensors. The most related works are [4], [5] and [6], in which a *Maximum Network Lifetime* problem is addressed. In [4], the coverage problem is modeled as a SET-K cover problem, in which sensors are organized into mutually exclusive sets and each set is meant to cover the monitored area/targets completely. A polynomial time heuristic called *Most Constrained-Minimally Constraining Covering Heuristic* is proposed to solve this NP-complete problem. [5] and [6] also addressed the energy efficient sensor organization problem using the same model. In [5], the disjoint dominating set approach is used to compute the mutually exclusive covers; in [6], a network flow model is used to compute the disjoint covers.

There have been some other research works related to the efficient use of energy through sensor self-organization. For example, [7], [8] and [9]. However, their objectives are focused on either energy efficient operations or sensor coverage connectivity, and none of them deals with bandwidth constraints. [10] derived upper bounds of network lifetimes for non-aggregating sensor networks using the path loss energy model; [11] generalized the bounds to the case of aggregating networks with specified topology and source movement by use of optimal role assignments; [12] proposed another self-organization technique among sensor nodes by use of a distributed randomized algorithm *Span*. *Span* can reduce the per node power consumption by a factor of 2 while maintaining a connected capacity-preserving global topology. In *Span*, a node can make local decisions on whether to sleep, or to join a forwarding backbone as a coordinator based on local information. [13] proposed an adaptive sensing coverage protocol that guarantees the full sensing coverage as well as the degree of coverage. In [13] each node in the sensor network is either in sleeping mode or in working mode. The basic protocol without differentiation is to make as many nodes as possible go to sleep to save energy and extend the lifetime of the sensor network while guaranteeing 100% sensing coverage of the target area. The basic protocol can be exended to provide differentiated surveillance by modifying the working shedule.

Each node can dynamically decide a working schedule for itself to gurantee a certain degree of coverage. [13] efficiently reduces the energy consumption and extended the half-life of the network, where half-life of the network is defined as the time from the begining of the deployment until half of the sensors are dead. [14] studied the activity scheduling problem that deals with rotating the role of each node among a set of operation modes so that the selected dominating nodes are connected and the energy consumption is balanced among wireless nodes. In a dominating set based broadcast routing scheme, only dominating nodes are allowed to retransmit the broadcast packet, therefore dominators usually consume more energy than dominatees. In the process of selecting a dominating node, nodes with a higher energy level are given higher priority. This scheme can significantly prolong the life span of each individual node.

In many applications, sensors cannot reach the base station within one-hop transmission. In this case, the construction of the aggregating tree is also critical to the lifetime of the sensor network. The Maximum Lifetime Data Aggregation Problem is defined as: given a set of sensor locations and energy levels associated with each sensor, as well as the location of the base station, find an efficient manner in which data should be collected and aggregated from all sensors and transmitted to the base station so that the system lifetime is maximized. [15] addressed the Maximum Lifetime Data Aggregation Problem using a scheme based on the intelligent selection of aggregation trees.

While all the above works model sensor coverage as a discrete 0-1 coverage problem, [16] addressed the continuous-domain coverage problem. [16] defined exposure as a function of intensities of multiple sensors, presented the concept of exposure-based coverage, and developed an efficient algorithm for exposure calculation in sensor networks, which can be used to find the worst case exposure-based coverage in sensor networks. Other works that deal with the coverage problem in continuous domain include [17], [18] and [19]. [18] proposed a polynomial time algorithm for finding the maximal breach path and the maximal support path based on the coverage calculation; [19] proposed an efficient distributed algorithm to find a path with maximum observability using a different sensing model. [17] formulated both the 0-1 minimum cover problem and the sensor field intensity based Minimal Cover problem, which is to find the minimum set of sensors that cover the same regions as the complete set of sensors; [17] also addressed the balanced operation scheduling problem in sensor networks, which is to compute a scheduling matrix such that the total time slices where each sensor is active is minimized, or the number of active sensors in each time slice is minimized.

III. Minimum Breach Problem in Sensor Networks

A. Problem Definition

To study the coverage breach problem, we use a discrete target model, in which the source of observation is given as a set of fixed points. Each point source has a range of detection. If sensors have equal probability of detecting objects from different directions, and objects have equal chance of being detected from all directions, the range of detection can be represented by a circular area. Different source activity can have different detection area, as long as some sensor lies within the area boundary, the point source is considered *covered*. In a more general case, the source to be monitored could be a specified region or an event that could happen at any point in the region. Since no pre-specified fixed point source is given, a straightforward way to solve this problem is to transform the area coverage problem into a fixed point coverage problem by dividing the monitored area A into a set of fields $\{a_1 ... a_M\}$, and then treat the fields as discrete point sources.

Using the discrete target model, we can formally define the Minimum Breach Problems as follows:

Definition 1: Given a set A of fixed points, and a set S of sensor nodes, organize sensor nodes into disjoint subsets $C_i = \{s_{i_1}, s_{i_2} ...\}$, $i = 1, ..., K$, where each subset $|C_i| \leq W$ and the overall breach is minimized.

For example, the monitored area is divided into five fields: $A = \{a_1, a_2, a_3, a_4, a_5\}$, and there are six sensors deployed in these fields $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$. Assume s_1 covers fields $\{a_1, a_2, a_3, a_4\}$, denoted as $s_1 = \{a_1, a_2, a_3, a_4\}$, $s_2 = \{a_1, a_2, a_5\}$, $s_3 = \{a_2, a_3, a_4, a_5\}$, $s_4 = \{a_2, a_3, a_5\}$, $s_5 = \{a_1, a_3, a_5\}$ and $s_6 = \{a_3, a_4, a_5\}$. For $W = 2$, the optimal solution is: $C_1 = \{s_1, s_4\}$, $C_2 = \{s_3, s_5\}$, and $C_3 = \{s_2, s_6\}$. Each of the disjoint subsets can completely cover all the fields in A.

The decision version minimum breach problem is formulated as follows:

PROBLEM: MINIMUM BREACH

INSTANCE: A collection S of sensors, a collection A of targets, and the sensor-target coverage map.

QUESTION: Can we divide S into disjoint subsets such that the overall breach is at most B and each subset has at most W sensors in it?

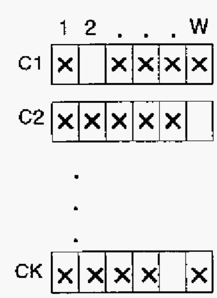


Fig. 1. Sensor organization to satisfy the bandwidth constraints

We show in section III-B that this problem is NP-complete.

B. Complexity Classification of the Minimum Breach Problem

To prove that *MINIMUM BREACH* problem is NP-complete, we first define a new problem:

Given a set of sensors, a set of targets and the sensor-target coverage map, divide the sensors into two disjoint subsets to minimize the overall breach. We name it *MINIMUM 2SET BREACH* problem. *MINIMUM 2SET BREACH* does not have constraints on the cardinality of each subset. The decision version can be described as:

PROBLEM: MINIMUM 2SET BREACH

INSTANCE: A collection S of sensors, a collection A of targets and the sensor-target coverage map.

QUESTION: Is there a partition of S into two subsets (without constraints on the cardinality of each subset) such that the overall breach is at most B?

Lemma 1: MINIMUM 2SET BREACH is NP-complete.

Proof. It is easy to see that *MINIMUM 2SET BREACH* $\in$ NP because a non-deterministic algorithm can guess a solution and check in polynomial time if the resulting overall breach is within the given bound B. The NP-completeness of the *MINIMUM 2SET BREACH* problem can be proved by a polynomial time transformation from *MAXIMUM SET SPLITTING* problem.

MAXIMUM SET SPLITTING problem is formally defined as:

INSTANCE: Given a collection C of subsets of a finite set S.

QUESTION: Is there a partition of S into two subsets S_1 and S_2 such that the cardinality of the subsets in C that are not entirely contained in either S_1 or S_2 (splitted) is at least $|C| - B$?

For each instance of *MAXIMUM SET SPLITTING* problem (I), we can construct an instance of *MINIMUM 2SET BREACH* problem (II) as follows:

Construct a set of sensors $S_{II} = S_I$, and a set of target $A_{II} = C_I$, make each element $a \in A_{II}$ correspond to an element $c \in C_I$. Each $a = \{s\}$ is a collection of sensors that cover the target a. If an element c is completely contained in subset $S1_I$ or $S2_I$, then the corresponding target a is breached in subset $S2_{II}$ or $S1_{II}$ respectively. If the solution $\{S1_{II}\} \cup \{S2_{II}\}$ satisfies that the total breach is at most B, then the corresponding solution $\{S1_I\} \cup \{S2_I\}$ also guarantees that the cardinality of the subsets in C that are splitted is at least $|C| - B$, and vice versa. This proves that the *MINIMUM 2SET BREACH* problem is NP-complete. ■

Next we can show with the size constraint W on each subset $S1$ and $S2$, *MINIMUM 2SET BREACH* problem remains NP-complete. Let's call the new problem *MINIMUM 2-W BREACH*.

PROBLEM: MINIMUM 2-W BREACH

INSTANCE: A collection S of sensors, a collection A of targets and the sensor-target coverage map.

QUESTION: Is there a partition of S into two subsets such that the overall breach is at most B and $|S1| \le W, |S2| \le W$?

Lemma 2: MINIMUM 2-W BREACH is NP-complete.

Proof. An instance of *MINIMUM 2SET BREACH* can be transformed into an instance of *MINIMUM 2-W BREACH* by adding additional sensors S' into S and one additional target a' into A. Make each new sensor $s' \in S'$ cover only a', and the new target a' is covered by all new sensors in S'. Make $W = |S|+1$, and $|S'| = 2W - |S|$. *MINIMUM 2SET BREACH* can be satisfied if and only if *MINIMUM 2-W BREACH* can be satisfied. ■

Next we can show that the *MINIMUM BREACH* is NP-complete. We can transform *MINIMUM 2-W BREACH* directly to *MINIMUM BREACH*: each instance of *MINIMUM 2-W BREACH* is an instance of *MINIMUM BREACH*. In fact, *MINIMUM 2-W BREACH* is a subclass of *MINIMUM BREACH* where the number of subsets is restricted to 2. Therefore *MINIMUM BREACH* is NP-complete.

Theorem 1: MINIMUM BREACH Problem is NP-Complete.

IV. Approximation Algorithms

To solve the *MINIMUM BREACH* Problem, we first formulate it as a 0-1 integer programming problem, then provide two heuristics based on this formulation.

A. Integer Programming Formulation of the Minimum Breach Problem

We use the following notations in the integer programming formulation:

- i the i^{th} sensor, when used as a subscript;
- j the j^{th} target, when used as a subscript;
- k the k^{th} subset, when used as a subscript;
- $x_{k,i}$ variable, $x_{k,i} = 1$ if the k^{th} subset includes sensor i, otherwise $x_{k,i} = 0$;
- $y_{k,j}$ variable, $y_{k,j} = 1$ if the k^{th} subset covers target j, otherwise $y_{k,j} = 0$;
- K the upper bound for the total number of subsets;
- W bandwidth, used as the upper bound for subset sizes;
- N the number of sensors;
- M the number of targets;
- $a_{i,j}$ $a_{i,j} = 1$ if sensor i covers target j, otherwise $a_{i,j} = 0$.

The reason that W is used as the upper bound for subset sizes is that we assumed each active sensor ships its data directly to the base station periodically, so the base station can only receive from at most W sensors in each cycle. The problem is illustrated in Fig. 1. The given N sensors are organized into K subsets, and in each subset C_k, $k = 1..K$, at most W sensors can be arranged. The minimum breach problem can be formulated as a zero-one Integer Programming problems as follows.

IP

$$\min\{\sum_{k=1}^{K}\sum_{j=1}^{M}(1 - y_{k,j})\} \tag{1}$$

We assume the total number of sensors N is a multiple of W, so $K = \frac{N}{W}$.

Subject to

$$\sum_{i=1}^{N} a_{i,j} x_{k,i} \geq y_{k,j}, \quad \forall j = 1..M,\ k = 1..K; \quad (2)$$

$$\sum_{k=1}^{K} x_{k,i} = 1, \quad \forall i = 1..N; \quad (3)$$

$$\sum_{i=1}^{N} x_{k,i} = W, \quad \forall k = 1..K; \quad (4)$$

$$y_{k,j} \in \{0,1\}, \quad \forall k = 1..K,\ j = 1..M; \quad (5)$$

$$x_{k,i} \in \{0,1\}, \quad \forall k = 1..K,\ i = 1..N. \quad (6)$$

Remarks:

If N is not a multiple of W, the above equations (3) and (4) can be adjusted as follows:

make $K = \lceil \frac{N}{W} \rceil$, then

$$\sum_{k=1}^{K} x_{k,i} = 1, \quad \forall i = 1..N; \quad (3)$$

$$\sum_{i=1}^{N} x_{k,i} \leq W, \quad \forall k = 1..K; \quad (4)$$

or make $K = \lfloor \frac{N}{W} \rfloor$, then

$$\sum_{k=1}^{K} x_{k,i} \leq 1, \quad \forall i = 1..N; \quad (3)$$

$$\sum_{i=1}^{N} x_{k,i} = W, \quad \forall k = 1..K; . \quad (4)$$

B. Heuristic I: RELAXATION

We propose a polynomial time algorithm `RELAXATION` for the above Integer Programming problem. `RELAXATION` is a three-step algorithm. At the first step, the Integer Programming problem (IP) is relaxed to a Linear Programming problem (LP), and an optimal solution for (LP) is computed. The optimal solution to (LP) may be fractional, so it may not satisfy the integer constraints (5) and (6). At the second step, a greedy algorithm is employed to find an integer solution based on the optimal solution obtained at the first step. At the third step, the solution from (IP) problem is used to construct the subsets.

At the first step, we remove the integer constraints on variables $x_{k,i}$ and $y_{k,j}$, and then solve the (LP) problem. Integer constraints (5) and (6) now become:

$$0 \leq y_{k,j} \leq 1, \quad \forall k = 1..K, \quad j = 1..M; \quad (5')$$

$$0 \leq x_{k,i} \leq 1, \quad \forall k = 1..K, \quad i = 1..N. \quad (6')$$

At the second step, after we get the optimal solution $\{x^*_{k,i}\}$ and $\{y^*_{k,j}\}$ to the (LP), we sort the components of $\{y^*_{k,j}\}$, for $k = 1..K$, $j = 1..M$ in non-increasing order; and then for each k, we sort $\{x^*_{k,i}\}$, for $i = 1..N$ in non-increasing order separately. Next we round those fractional components in $\{y^*_{k,j}\}$ and $\{x^*_{k,i}\}$ and obtain an integer solution to the (IP). Here we use a greedy strategy that tries to set variables with larger values to 1. Let $\{y^A_{k,j}\} \cup \{x^A_{k,i}\}$ be an approximate solution to the (IP). The heuristic is formally presented as follows:

`Algorithm RELAXATION`

`/** STEP One: Solve LP **/`

Solve the LP problem, get optimal solution $\{x^*_{k,i}\}$ and $\{y^*_{k,j}\}$

`/** STEP Two: Rounding **/`

initialize $y^A_{k,j} = 0$ and $x^A_{k,i} = 0$, $\forall k = 1..K,\ i = 1..N,\ j = 1..M$.

Sort the obtained optimal solution $\{y^*_{k,j}\}$ in non-increasing order and put them in a list Y.

while Y is not empty **do**
  remove an element $y^*_{k,j}$ from the head of Y
  sort the obtained optimal solution $\{x^*_{k,i} | a_{i,j} = 1\}$, $i = 1..N$ in non-increasing order and put them in a list X_k
  while X_k is not empty **do**
    remove an element $x^*_{k,i}$ from the head of X_k
    if $a_{i,j} = 1$, and making $x^A_{k,i} = 1$ satisfies
      $\sum_{k=1}^{K} x^A_{k,i} \leq 1, \quad \forall i = 1..N$ and
      $\sum_{i=1}^{N} x^A_{k,i} \leq W, \quad \forall k = 1..K$
    then
      set $x^A_{k,i} = 1$ and $y^A_{k,j} = 1$, break
    end if
  end while
end while
get solution $\{y^A_{k,j}\}$ and $\{x^A_{k,i}\}$.

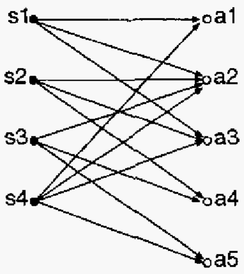
`/** STEP Three: Construct Subsets **/`

for $k = 1..K$ **do**
  $C_k = \phi$
  for i=1..N **do**
    if $x^A_{k,i} = 1$ **then**
      set $C_k = C_k \bigcup$ *sensor* s_i
    end if
  end for
end for
return the final solution $\{C_k\}$

`END of RELAXATION`

The runtime of `RELAXATION` is dominated by the (LP) solver, which is $O(n^{3.5})$ if Karmarkar's Interior Point method is used, or $O(n^3)$ if Ye's algorithm is used [20]. For a sensor network that contains N sensors and M targets with a constant bandwidth W, the number of variables is $n = N(N+M)/W$.

We implemented the `RELAXATION` heuristic through the



(a) bipartite graph

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

(b) coverage matrix $(a_{i,j})$

Fig. 2. An example: 4 sensors, 5 targets, with bandwidth=2

example in Fig. 2. The final solution from RELAXATION is:

$$y_{1,j} = 1, j = 1..5, \quad \{x_{1,1}\ x_{1,2}\ x_{1,3}\ x_{1,4}\} = \{0\ \ 1\ \ 0\ \ 1\}$$

$$y_{2,j} = 1, j = 1..5, \quad \{x_{2,1}\ x_{2,2}\ x_{2,3}\ x_{2,4}\} = \{1\ \ 0\ \ 1\ \ 0\}$$

So we get optimal solution $C1 = \{s_2, s_4\}$ and $C2 = \{s_1, s_3\}$.

C. Heuristic II: MINBREACH

The linear programming based RELAXATION has a scalability problem since to obtain the optimal solution of the (LP) requires at least $O(n^3)$ running time. Using the above formulation, solving (LP) significantly slows down the solution process. To avoid solving the linear programming problem, we introduce a fast heuristic MINBREACH. Using A to denote the coefficient matrix, and x to denote all variables, the integer programming problem can be presented as:

$$\max \quad c^T x, \text{ where } c_j \geq 0$$

$$\textit{Subject to} \quad Ax \leq b$$
$$x \geq 0$$

Where the coefficient matrix A has entries $\{0, -1, 1\}$, and it can be partitioned into a lower part and an upper part: the lower part is related to constraints (3) and (4), and the upper part is related to constraint (2). As shown in Fig. 3, we use I_1 to denote the rows in upper part, which contain entries $\{0, 1, -1\}$, and use I_2 to denote the rows in the lower part, which only contain entries $\{0, 1\}$. We also use J_X to represent the columns that correspond to the $\{x_{k,i}\}$ in the original (IP), and use J_Y to represent the columns that correspond to the $\{y_{k,j}\}$ in the original (IP). The objective function is to maximize $\sum_j c_j x_j$, so we initialize $x_j = 1$. If some relations are violated, we find the variable x_r that would most likely reduce the total number of violations, then reduce x_r to 0. The heuristics *MINBREACH* is presented as follows:

Algorithm MINBREACH
  /** Phase I **/
  set $x_j = 1$, for $j = 1, 2, ...n$.
  set $J = J_X$
  while $X = \{x_j\}$ is not a feasible solution to the (IP) and $J \neq \phi$ **do**
    set $diff_i = |\sum_{j=1}^{n} A_{ij} x_j - b_i|, \quad \forall i = 1, 2, ...m$
    set $I_U = \{i | A_{i1} x_1 + ... A_{in} x_n > b_i, \ and \ i \in I_1\}$
    set $I_L = \{i | A_{i1} x_1 + ... A_{in} x_n > b_i, \ and \ i \in I_2\}$
    For each $j \in J$, set

$$l_j = \sum_{i \in I_L, \ diff_i = 1} A_{ij} X_j$$

$$u_j = \sum_{i \in I_1 - I_U, \ diff_i = 0} A_{ij} X_j$$

    set $J^+ = \{j | j \in J, \ and \ l_j - |u_j| > 0\}$
    if $J^+ \neq \phi$ **then**
      Find $r \in J^+$, such that

$$l_r - |u_r| = \max_{j \in J^+} \{l_j - |u_j|\}$$

    else
      For each $j \in J$, set

$$p_j = \sum_{i \in I_L} A_{ij} X_j \cdot diff_i$$

      Find $r \in J$, such that

$$p_r = \max_{j \in J} p_j$$

    end if
    set $x_r = 0$.
    set $J = J - \{r\}$.
    if relation (4) has "=" **then**
      set $J_k = \{j |\ x_j$ and x_r belong to the same subset$\}$.
      for each $j \in J_k$ **do**
        if $x_j = 1$ and $\sum_{j \in J_k} x_j = W$ **then**
          set $J = J - \{j\}$.
        end if
      end for
    end if
  end while
  /** Phase II**/
  if $X = \{x_j\}$ is not a feasible solution to the (IP) **then**
    set $I_{U'} = \{i | A_{i1} x_1 + ... A_{in} x_n \geq b_i + 1, \ and \ i \in I_1\}$.
    for each $i \in I_{U'}$ **do**
      set $x_i = 0$
    end for
  end if
  return solution
END of MINBREACH()

In the above algorithm, I_1, I_2, J_X and J_Y are shown in Fig. 3. $l_j - |u_j|$ is the contribution index of variable x_j, which is an indicator of how many violations can be removed by setting x_j to 0; p_j is the potential index of variable x_j, which indicates if x_j is reduced to 0, how much it will contribute to remove violations of the lower part in the future. The selection of r guarantees that the number of violated rows in the lower part is non-increasing in every round of Phase I.

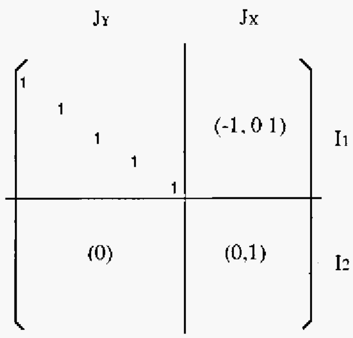


Fig. 3. Coefficient matrix A in $Ax \leq b$

At the end of phase I, there are no violations in the lower part; if x is not a feasible solution to (IP), then there must be some violations in the upper part. Phase II set more x_j's to 0 to make x a feasible solution.

For a sensor network of N sensors and M targets, the time complexity of MINBREACH is $O(N^2M(N+M))$, while the time complexity of RELAXATION is $O(n^3)$ using Ye's algorithm [20] and $O(n^{3.5})$ using Karmarkar's algorithm, where $n = O(N(N+M))$. Next, we compare the performance of the above algorithms by simulation.

V. Simulation Study

The objectives of this simulation are to provide a performance comparison of the two heuristics, and meanwhile, using the overall breach rate as a performance metric, to study the effects of different network design parameters on the network performance. Network design parameters include the bandwidth W, the number of sensors N, and a breach factor f, which is related to the density of the coverage matrix. The breach rate is defined as:

$$breach\ rate = \frac{\sum_{k=1}^{K}\sum_{j=1}^{M} y_{k,j}}{\sum_{k=1}^{K}\sum_{j=1}^{M} 1} = \frac{\sum_{k=1}^{K}\sum_{j=1}^{M} y_{k,j}}{K \cdot M}$$

We start from a bipartite graph of sensor nodes and targets where the link between a sensor node and a target node exists if the sensor covers the target. The link probability is controlled by a breach factor f. High values of f indicate low link probabilities. For a constant breach factor, when we increase the total number of sensors, the average #sensors covering each target is also increased. For example, a breach factor 8 results in 12.72 sensors/target in a 100-sensor 50-target network, but in a 20-sensor 50-target network, the average #sensors/target is 2.86. Higher values of f also result in higher breach rates, as we can see from the following experiments.

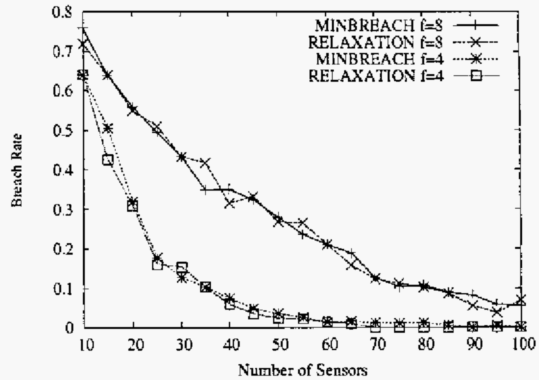


Fig. 4. Comparison of RELAXATION and MINBREACH

Fig. 4 shows the performance comparison of the two heuristics. The networks are setup as follows: as the number of targets increases from 10 to 100, the number of sensors also increases from 10 to 100, and bandwidth increases from 2 to 20. Breach factor $f = 8$ and $f = 4$ are used. For both $f = 8$ and $f = 4$, the two heuristics generated very similar results. The curves with $f = 8$ are always above the curves with $f = 4$, which verifies that higher f leads to higher breach rate.

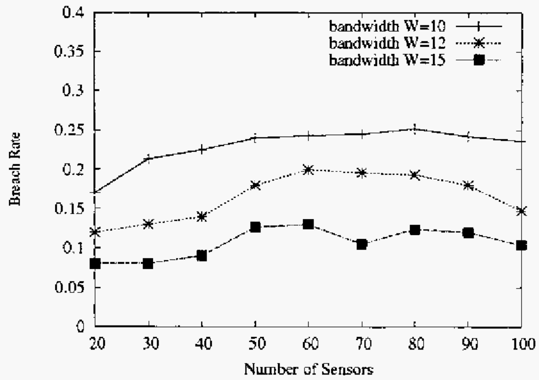
Fig. 5(a) shows the effect of increasing sensors on improving network coverage. It shows that with a constant bandwidth, increasing sensors alone may not result in improved coverage, since none of the three curves shows an obvious trend of decrease in the breach rate. In contrast, the three curves of different bandwidths show that there is a clear trend that the breach rate is decreased when the bandwidth increases, which is also consistent with the result in Fig. 5(b). The network instances are generated with a constant breach factor $f = 8$ and target number $M = 50$.

Fig. 5(b) shows the effect of increasing bandwidth on improving network coverage. For a collection of 40-sensor 50-target networks, as bandwidth increases, the breach rate monotonically decreases. Bandwidth constraint is more of a limiting factor for networks with a higher breach factor. This is because in networks with a higher breach factor, each target is covered by fewer sensors; Therefore to cover all targets requires more sensors in each subset. However, the bandwidth constraint forbids to add more sensors in each subset.

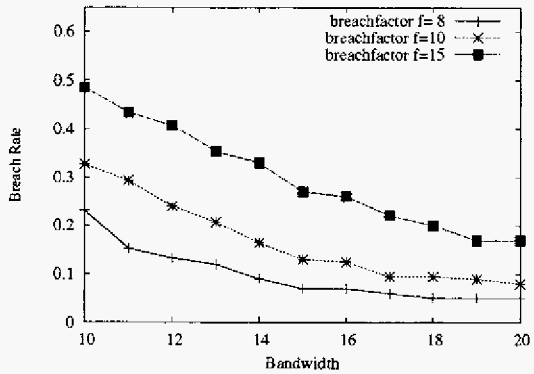
In conclusion, this simulation study verified the prediction that bandwidth constraints forbid to improve network coverage by adding more sensors. Network performance can be improved only if bandwidth increases as well when more sensors are deployed.

VI. Conclusion and Extensions

This paper presents the breach problem in wireless sensor networks due to the communication bandwidth limitation. *MINIMUM BREACH* Problem is defined and proved to be NP-complete. A 0-1 integer programming model is developed,



(a) Bandwidth constraint is the limiting factor when more sensors join the network; increasing bandwidth can significantly decrease the breach rate



(b) Effect of increasing bandwidth

Fig. 5. Interactions between network design parameters W and N and their effects on breach rate

and two polynomial time approximation algorithms based on this model are proposed. Extensive simulation is conducted to compare the performance of the two algorithms. Conclusions derived from the simulation are consistent with the prediction: bandwidth constraint is indeed a limiting factor on sensor network coverage; to improve coverage, deploying more sensors must be accompanied by increasing bandwidth, otherwise, the coverage may be decreased as a result.

While total breach is used as a metric for sensor coverage in this paper, the maximal breach in any subset of sensors might be more of interest to some applications. For example, at any time, the required coverage rate of a monitored area must be at least 95%. To minimize the maximal breach is also an NP-complete problem that requires efficient approximation algorithms, which will be addressed in our future work. Another complementary problem is, for any specific target, what is the worst case estimation of the longest breach time? To bound the longest breach time of any target is also of importance for a lot of applications. For example, in a factory warehouse, the items are tracked by the tags attached to them. If items move from one place to another, the sensors embedded on the wall should be able to detect it. This requires that items must be kept tracking continuously or with bounded intervals of breach. The methodology developed in this paper may be generalized to address these problems.

References

[1] I. A. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey of sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug 2002.
[2] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "A discrete-time battery model for high-level power estimation," in *Proceedings of DATE*, 2000, pp. 35–39.
[3] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Mobile Computing and Networking*, 1999, pp. 263–270.
[4] Sasa Slijepcevic and Miodrag Potkonjak, "Power efficient organization of wireless sensor networks," in *IEEE International Conference on Communications, Helsinki, Finland*, June 2001, vol. 2, pp. 472–476.
[5] M. Cardei, D. MacCallum, M. X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du, "Wireless sensor networks with energy efficient organization," *Journal of Interconnection Networks*, vol. 3, no. 3&4, pp. 213–229, 2002.
[6] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," 2002, manuscript.
[7] Himanshu Gupta, Samir R. Das, and Quinyi Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," in *MobiHoc'03, June 1-3, Annapolis, Maryland, USA*, 2003.
[8] A. Srivastava, J. Sobaje, M. Potkonjak, and M. Sarrafzadeh, "Optimal node scheduling for effective energy usage in sensor networks," in *IEEE Workshop on Integrated Management of Power Aware Communications Computing and Networking*, 2002.
[9] D. Tian and N. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proc. ACM Workshop on Wireless Sensor Networks and Applications, Atlanta, Oct.2002.*, 2002.
[10] M. Bhardwaj, A. Chandrakasan, and T. Garnett, "Upper bounds on the lifetime of sensor networks," in *IEEE International Conference on Communications, Pages 785 - 790, 2001*, 2001.
[11] M. Bhardwaj and A. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *IEEE InfoCom, 2002*, 2002.
[12] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *International Conference on Mobile Computing and Networking*, 2001, pp. 85–96.
[13] Ting Yan, Tian He, and John A. Stankovic, "Differentiated surveillance for sensor networks," in *roceedings of the first international conference on Embedded networked sensor systems*, November 2003, pp. 51–62.
[14] I. Stojmenovic and J. Wu, "Broadcasting and activity-scheduling in ad hoc networks," in *Ad Hoc Networking, IEEE Press, S. Basagni, M. Conti, S. Giordano and I. Stojmenovic (eds.)*, June 2004.
[15] Koustuv Dasgupta, Konstantinos Kalpakis, and Parag Namjoshi, "Improving the lifetime of sensor networks via intelligent selection of data aggregation trees," in *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference, Orlando, Florida*, January 19-23 2003.
[16] Seapahn Megerian, Farinaz Koushanfar, Gang Qu, Giacomino Veltri, and Miodrag Potkonjak, "Exposure in wireless sensor networks: Theory and practical solutions," *Journal of Wireless Networks*, vol. 8, no. 5, pp. 443–454, september 2002.
[17] Seapahn Meguerdichian Miodrag, "Low power 0/1 coverage and scheduling techniques in sensor networks," UCLA Technical Reports 030001, January 2003.
[18] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *INFOCOM*, April 2001, vol. 3, pp. 1380–1387.
[19] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in wireless ad-hoc sensor networks," in *IEEE Int'l Conf. on Communications (ICC 2002), New York, New York*, April 2002.
[20] Y. Ye, "An $o(n^3 l)$ potential reduction algorithm for linear programming," *Mathematical Programming*, vol. 50, pp. 239–258, 1991.