Wireless Networks 11, 333–340, 2005

Improving Wireless Sensor Network Lifetime through Power Aware Organization

MIHAELA CARDEI*
Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA

DING-ZHU DU
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA

Abstract. A critical aspect of applications with wireless sensor networks is network lifetime. Battery-powered sensors are usable as long as they can communicate captured data to a processing node. Sensing and communications consume energy, therefore judicious power management and scheduling can effectively extend operational time. To monitor a set of targets with known locations when ground access in the monitored area is prohibited, one solution is to deploy the sensors remotely, from an aircraft. The loss of precise sensor placement would then be compensated by a large sensor population density in the drop zone, that would improve the probability of target coverage. The data collected from the sensors is sent to a central node for processing. In this paper we propose an efficient method to extend the sensor network operational time by organizing the sensors into a maximal number of disjoint set covers that are activated successively. Only the sensors from the current active set are responsible for monitoring all targets and for transmitting the collected data, while nodes from all other sets are in a low-energy sleep mode. In this paper we address the maximum disjoint set covers problem and we design a heuristic that computes the sets. Theoretical analysis and performance evaluation results are presented to verify our approach.

Keywords: wireless sensor networks, energy efficiency, node organization, disjoint set covers

1. Introduction

Wireless sensor networks provide new applications for environment monitoring, and military surveillance applications. Recent developments in hardware miniaturization combined with low-cost mass production and advances in wireless communications technologies have made possible applications with large numbers of sensors. In some cases ground access to the area of the objectives to be monitored is difficult or dangerous, so one solution to install the sensors is to deploy them from an aircraft. Without precise positioning, the only way to provide adequate target coverage by sensors is to use more sensors than the optimal number. Large sensor density will increase the probability of target coverage, considering that sensors may be randomly dispersed in the targets' proximity.

One of the main issues in sensor networks is network lifetime. With the available technology, the sensors are battery powered. Due to size and cost constraints, the energy available at each sensor for sensing and communications is limited and globally affects the application lifetime. A solution for mitigating the energy problem is to implement mechanisms for efficient energy management. One method is based on scheduling sensor activity so that for each sensor the active state, in which it actually performs its monitoring task alternates with a low-energy idle (sleep) state. As pointed out in [3,11] the ratio of energy consumed between the active and the sleep state is considerable and may be as high as 100.

Another result relevant to this approach is that batteries discharging in short bursts with significant off-time have approximately twice as long a lifetime as compared to a continuous mode of operation (see [2]). Therefore, a mode of operation that alternates active and inactive battery states extends network lifetime.

In this paper we address the problem of energy efficiency in wireless sensor applications for surveillance of a set of targets with known locations. We consider that a large number of sensors are dispersed randomly in close proximity to a set of objectives and send the monitored information to a central processing node. Every target must be monitored at all times by at least one sensor and every sensor is able to monitor all targets within its operational range. One method for extending the sensor network lifetime is to divide the set of sensors into disjoint sets such that every set completely covers all targets. We consider that a target is covered if it is within an active sensor's operational range. These disjoint sets are activated successively, such that at any moment in time only one set is active. The sensors from the active set are into the *active* state and all other sensors are in a low-energy *sleep* state. As all targets are monitored by every sensor set, the goal of this approach is to determine a maximum number of disjoint sets, so that the time interval between two activations for any sensor is longer. By decreasing the fraction of time a sensor is active, the overall time until power runs out for all sensors is increased and the application lifetime is extended proportionally by a factor equal to the number of disjoint sets. As a consequence, the spatial density of active nodes is lowered, thus improving channel access for transmitting sensor data.

*Corresponding author.
E-mail: mihaela@cse.fau.edu

The disjoint sets in our approach are modeled as disjoint set covers, where every cover completely monitors all the target points. We assume that the targets have fixed locations, so the algorithm for computing the covers is executed only once by a central node after the location for all sensors has been determined. After the wireless sensors are deployed, they activate their positioning service and send their location information to the central node. Based on this information, the central node computes the disjoint set covers and sends membership information back to every sensor. Knowing the set it belongs to and the number of covers, every sensor is then able to identify the time periods when it has to be active or in the sleep state. We assume that a time synchronization service is available to sensors, most likely facilitated by periodic beacon messages from the central node or on-board GPS receivers.

In this paper we define the disjoint set covers problem and demonstrate it is NP-complete. Then we determined a lower-bound performance result and propose an efficient heuristic for set covers computation.

There is a significant amount of literature addressing the issue of energy efficiency in wireless networking, at all layers of the protocol stack. In general, proposed techniques for energy saving fall in one of the following categories: (1) schedule operations, to allow nodes to enter low energy states; (2) choose routes that consumes the lowest energy; (3) selectively use wireless nodes based on their energy status; (4) reduce amount of data and avoid useless activity.

Scheduling nodes to enter low energy states is an efficient way to accomplish energy savings. Next, we review few access protocols which attain energy savings by scheduling the node transmissions, such that every node alternates between active and low energy idle states. IEEE 802.11 MAC [1] proposes a power saving method for use both in ad-hoc environment as well as with PCF mechanism. In ad-hoc environments, the nodes may enter a sleep state, and wake up from time to time to determine if any traffic is pending for them. In PCF control mechanisms, the access point (AP) coordinates the medium access by using a traffic indication map (TIM) which is transmitted periodically and which identifies the stations for which traffic is pending and buffered in the AP. If a station is listed in the TIM, then it stays awake, otherwise will doze until the next TIM is scheduled. The power saving mode in ETSI HIPERLAN [5] is a contract between at least two stations. Each p-saver station is coordinating a dozing cycle with one or more p-supporters, which act as surrogate destinations for the p-saver station's traffic while it is dozing.

An effective way to conserve energy is to schedule apriori the wireless node transmissions, allowing them to enter a low state energy while they are inactive. This idea is explored in [4], where authors study the communication from a base station to a large number of wireless nodes. Three access protocols are designed, considering two important factors: low delay and low energy requirements. These protocols propose a transmission scheduling strategy at the base station as well as a wake-up schedule at each node. In the grouped-tag TDMA protocols, the nodes are divided in groups and each group is assigned a TDMA slot for communication with the base station. In directory protocols, the base station broadcasts a directory which lists the destinations, permitting nodes to schedule their wake-up slots to coincide with the broadcast of their packets. In the pseudorandom protocols, the base station knows when every node is awake, based on sharing the seed for a random number generator, and thus knows when to send packets to specific destinations. In [11], the authors perform a comprehensive study of the problem of scheduling the communication between the central controller and other wireless nodes, with focus on energy conservation. The paper contributes three directory protocols that may be used by the central node to coordinate data transmissions considering multiple factors such as traffic-type (e.g. downlink, uplink, peer-to-peer) and the effects of packets errors.

In [10], the authors propose an energy conservation technique for wireless sensor networks that works by selecting and successively activating mutually exclusive sets of sensor nodes, where every set completely covers the entire monitored area. Their method achieves energy savings by increasing the number of disjoint covers. The authors propose a heuristic solution to this problem. In Section 4, we compared the performance of this heuristic versus the performance of our heuristic. In [3], we proposed an efficient node organization scheme, by grouping the sensors in disjoint dominating sets, with every set successively responsible for area monitoring.

In [9], the authors proposed a new multiaccess protocol, PAMAS, based on MACA [8], with the addition of a separate signaling channel. PAMAS achieves energy savings by powering off the nodes which are not actively transmitting or receiving packets.

The rest of the paper is structured as follows. In Section 2 we present the disjoint set covers problem, show its NP-completeness and a lower bound result. Section 3 continues with a heuristic for computing the maximum number of disjoint set covers. Section 4 presents performance evaluation results and Section 5 concludes the paper.

2. Disjoint set covers (DSC) problem

In this section we define the disjoint set covers (DSC) problem and prove its NP-completeness. We also prove that any polynomial-time approximation algorithm for DSC problem has a lower bound of 2.

Let us assume that n sensors $S_1, S_2, \ldots, S_n$ are deployed in territory to monitor m targets $T_1, T_2, \ldots, T_m$. In order to increase the energy savings, the goal is to divide the sensors into a maximum number of disjoint sets, such that every set completely covers all the target points. We consider that a target, identified by its position, is covered by a sensor when it lies within the sensing range of that sensor.

Our problem is modeled as a collection of sensors $C = \{S_1, S_2, \ldots, S_n\}$, where each sensor covers a subset of the targets in $T = \{T_1, T_2, \ldots, T_m\}$, e.g. $S_i = \{T_{i_1}, T_{i_2}, \ldots, T_{i_n}\}$, $1 \leq i \leq n$. We want to determine a maximum number of disjoint covers, where every cover is a set of sensors which together monitor all the target points.

Next, we define the Disjoint Set Covers problem (see [10]), which can be seen as a generalization of the minimum cover problem [6], and show its NP-completeness.

Definition 1. DSC: given a collection C of subsets of a finite set T, find the maximum number of disjoint covers for T. Every cover C_i is a subset of C, $C_i \subseteq C$, such that every element of T belongs to at least one member of C_i, and for any two covers C_i and C_j, $C_i \cap C_j = \phi$.

Next, we present the Double-Set-Covering problem, which will be used to show that 2-DSC is NP-complete.

Definition 2. *Double-Set-Covering*: given two disjoint sets A and B and a collection C of subsets of $A \cup B$, determine whether C can be partitioned into two disjoint subcollections C_A and C_B covering A and B respectively.

Theorem 1. Double-Set-Covering is NP-complete.

Proof. It is easy to show that *Double-Set-Covering* $\in$ NP, since a nondeterministic algorithm needs only to partition C into two disjoint subcollections and then verify in polynomial time if one subcollection covers A and the other covers B.

To show that *Double-Set-Covering* is NP-hard, we reduce the *3SAT* problem [6] to it. A boolean formula is in conjunctive normal form (*CNF*) if it is expressed as an AND of clauses, each of which is the OR of one or more literals. A boolean formula is in *3-CNF* if each clause has exactly three distinct literals. The *3SAT* problem is defined as follows: given a *3-CNF* formula F, determine whether F has a satisfiable assignment. Let F be a *3-CNF* formula with m clauses $c_1, c_2, \ldots, c_m$, over n variables $x_1, x_2, \ldots, x_n$. Let us define $A = \{x_1, \ldots, x_n, c_1, \ldots, c_m\}$ and $B = \{\bar{x}_1, \ldots, \bar{x}_n\}$. Let C be the collection of following $2n$ subsets of $A \cup B$: $S_i = \{x_i, \bar{x}_i\} \cup \{c_j \mid c_j \textit{ contains the literal } x_i\}$ and $T_i = \{x_i, \bar{x}_i\} \cup \{c_j \mid c_j \textit{ contains the literal } \bar{x}_i\}$, where $i = 1, \ldots, n$.

Next, we show that F is satisfiable if and only if C can be partitioned into two subcollections covering A and B respectively.

First, suppose F is satisfiable. Define $C_A = \{S_i \mid x_i = 1\} \cup \{T_i \mid \bar{x}_i = 1\}$ and $C_B = C - C_A$. Clearly, C_A and C_B cover A and B respectively.

Now, assume that C can be partitioned into two subcollections C_A and C_B covering A and B respectively. Define $x_i = 1$ if $S_i \in C_A$ and $x_i = 0$ otherwise. Then, every clause is satisfied since C_A covers A.

Finally, we note that this reduction is polynomial-time computable. □

The decision version of the *DSC* problem is stated as follows:

k-DSC (disjoint set covers): Given a set T and a collection C of subsets of T, determine whether C can be partitioned into k disjoint set covers or not.

Theorem 2. 2-DSC is NP-complete.

Proof. It is easy to show that *2-DSC* $\in$ NP, since a nondeterministic algorithm needs only to partition C into two disjoint subcollections and then verify in polynomial time if every subcollection covers T .

To show that *2-DSC* is NP-hard, we reduce the *Double-Set-Covering* problem to it in polynomial-time. Consider an instance of the *Double-Set-Covering* problem, which consists of two disjoint sets A and B and a collection C of subsets of $A \cup B$. Choose an element u not in $A \cup B$ and define $U = \{u\} \cup A$ and $V = \{u\} \cup B$. Now, we show that C can be partitioned into two disjoint subcollections covering A and B respectively if and only if $C \cup \{U, V\}$ contains two disjoint set covers for $\{u\} \cup A \cup B$.

First, suppose C can be partitioned into two disjoint subcollections C_A and C_B covering A and B respectively. Then $C_A \cup \{V\}$ and $C_B \cup \{U\}$ form two disjoint set covers for $\{u\} \cup A \cup B$.

Next, let us assume that $C \cup \{U, V\}$ contains two disjoint set covers C_1 and C_2 for $\{u\} \cup A \cup B$. Since there are only two sets U and V containing u, implies that U and V must belong to different set covers. Without loss of generality, assume C_1 contains U and C_2 contains V. Then $C_1 - \{U\}$ must cover B and $C_2 - \{V\}$ must cover A. □

Corollary 1. For any $k \geq 2$, k-DSC is NP-complete.

Proof. We can construct a polynomial-time reduction from *2-DSC* to *k-DSC* ($k > 2$) by adding $k - 2$ set covers into input collection of subsets in a proper way. □

Corollary 2. If $NP \neq P$, then DSC has no polynomial-time approximation algorithm with performance p for any $p < 2$.

Proof. Suppose such an approximation algorithm *APPROX* exists. Then for a collection C having at least two disjoint set covers, *APPROX* can tell that it contains at least $2/p > 1$ disjoint set covers. For a collection C containing at most one set cover, *APPROX* tells that C contains ≤ 1 set cover. Therefore, *APPROX* can solve *2-DSC* in polynomial-time, contradicting $NP \neq P$. □

3. An heuristic to compute maximum disjoint set cover

In this section we present a heuristic for the *DSC* problem. Given a collection C of subsets of a finite set T we want to determine the maximum number of disjoint subcollections, each covering the set T. Let us consider $C = \{S_1, S_2, \ldots, S_n\}$ and $T = \{T_1, T_2, \ldots, T_m\}$, where every S_i, $1 \leq i \leq n$ is a set of elements in T.

In order to compute the maximum number of covers, we first transform DSC into a maximum-flow problem (MFP), which is then formulated as a mixed integer programming (MIP). Based on the solution of the MIP, we design a heuristic to compute the number of covers. Next, we present every step in detail.

Let us first transform DSC problem into a MFP as follows:

Step 1. Consider a bipartite directed graph $G = (V, E)$ where the vertex set $V = C \cup T$ and $S_i T_j \in E$ if and only if T_j is in S_i, where $1 \leq i \leq n$ and $1 \leq j \leq m$. Then assign to every edge $S_i T_j$ a capacity $c_{S_i T_j} = 1$. Create a vertex X and connect every vertex T_j in T to X with an edge of capacity 1.

Step 2. Find a critical element in T which is contained by a minimum number of subsets in the collection C and note this number with k. Draw k copies of G, namely $G_1, G_2, \ldots, G_k$. In these k copies (components), let the first index in a vertex notation reflect the component it belongs to, e.g. a vertex S_i in G, is named $S_{1i}, S_{2i}, \ldots, S_{ki}$ in $G_1, G_2, \ldots, G_k$.

Step 3. Create a source node S and for each S_i in C, create a vertex S_{0i}. Then connect the source S with S_{0i} with an edge with capacity equal with the degree of S_i in G. Also, connect S_{0i} with S_{ji} for any $1 \leq j \leq k$ and assign a capacity equal with the degree of S_i in G.

Step 4. Create two sinks Y_1 and Y_2. Connect each vertex X_j with $1 \leq j \leq k$ to Y_2 and assign a capacity m. Then connect every vertex T_{ij} with $1 \leq i \leq k$ and $1 \leq j \leq m$ to Y_1 and assign the capacity n.

We define the flow f as an integer-valued function, that satisfies the following properties:

P1. Flow constraint: for all $uv \in E$, $0 \leq f_{uv} \leq c_{uv}$. An additional condition to the classic flow network is that for any $v \neq Y_1$, $f_{uv} \in \{0, c_{uv}\}$.

P2. Flow conservation: for all $u \in V - \{S, Y_1, Y_2\}$, $\sum_{v \in V, uv \in E \text{ or } vu \in E} f_{uv} = 0$.

The goal of this maximum-flow problem is to maximize the flow received in Y_2.

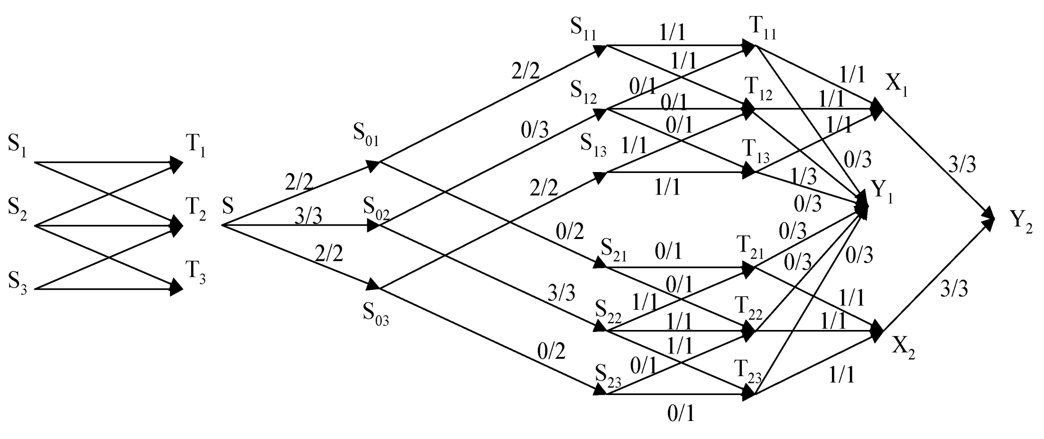
In the example in figure 1, we present the flow network construction when $C = \{S_1, S_2, S_3\}$, $T = \{T_1, T_2, T_3\}$ and $S_1 = \{T_1, T_2\}$, $S_2 = \{T_1, T_2, T_3\}$, $S_3 = \{T_2, T_3\}$. Figure 1(a) shows the bipartite graph G and in figure 1(b) the whole graph is presented, with flow/capacity values assigned for each edge.

Theorem 3. Given a collection $C = \{S_1, S_2, \ldots, S_n\}$ of subsets of a finite set $T = \{T_1, T_2, \ldots, T_m\}$, the DSC problem returns c^* covers if and only if the maximum-flow problem obtains the flow c^*m in Y_2.

Proof. Let us note with f_{Y_2} the flow received in Y_2, $f_{Y_2} = \sum_{i=1\ldots k} f_{X_i Y_2}$. As every flow $f_{X_i Y_2}$ can be only 0 or m, the flow f_{Y_2} is a multiple of m.

Let us first consider that the maximum number of covers is c^*. We show that the maximum flow that can be obtained in Y_2 is c^*m. Suppose by contradiction that a larger flow cm, $c > c^*$ is obtained in Y_2. Then there exists c vertices $X_{i_1}, X_{i_2}, \ldots, X_{i_c}$ such that $f_{X_{i_j} Y_2} = m$, $j = 1 \ldots c$. We construct c covers as follows: $C_j = \{S_u | f_{S_{0u} S_{ju}} = |S_u|\}$ for $j = i_1 \ldots i_c$. These covers are disjoint, because for a sensor S_u, there is at most one component p such that $f_{S_{0u} S_{pu}} = |S_u|$. Also, every element of T belongs to at least one member of C_j. Every vertex T_{ja}, $1 \leq a \leq m$ receives a flow greater or equal than 1, therefore there is a vertex $S_{jb} \in C_j$, $1 \leq b \leq n$ such that T_a belongs to S_b. Therefore we have constructed more than c^* covers, contradicting our assumption.

Let us consider now that the maximum flow obtained in Y_2 is c^*m. We show that DSC problem has maximum c^* covers. Suppose by contradiction that DSC problem could return c covers, $c > c^*$, namely $C_1, \ldots, C_c$. We assign the flow in the network as follows. For $j = 1 \ldots c$, if $S_a \in C_j$, then assign $f_{S_{0a} S_{ja}} = |S_a|$, otherwise $f_{S_{0a} S_{ja}} = 0$. For $j = c+1, \ldots, k$ assign $f_{S_{0a} S_{ja}} = 0$ for $a = 1 \ldots n$. The flow on the remaining edges can easily be computed, resulting in $f_{Y_2} = cm$, therefore contradicting our assumption. □



(a) Bipartite graph G

(b) The flow network with a flow/capacity assignment

Figure 1. Construction of the flow network for $C = \{S_1, S_2, S_3\}$, $T = \{T_1, T_2, T_3\}$, $S_1 = \{T_1, T_2\}$, $S_2 = \{T_1, T_2, T_3\}$, $S_3 = \{T_2, T_3\}$ and a flow/capacity assignment.

Next, we formulate this maximum flow problem as a mixed integer programming (MIP):

maximize f_{Y_2}
subject to

$$
\begin{array}{lll}
(1) & f_{uv} \le c_{uv} & uv \in E \\
(2) & \displaystyle\sum_{u:uv\in E} f_{uv} - \sum_{u:vu\in E} f_{vu} = 0 & v \in V; v \neq \{S, Y_1, Y_2\} \\
(3) & f_{S_{pi}T_{pi_1}} = f_{S_{pi}T_{pi_2}} & i = 1 \dots n; p = 1 \dots k; \\
 & = \cdots = f_{S_{pi}T_{pi_j}} & S_i = \{T_{i_1}, T_{i_2}, \dots, T_{i_j}\}; \\
 & & i_j = |S_i| \\
(4) & f_{T_{p1}X_p} = f_{T_{p2}X_p} & p = 1 \dots k \\
 & = \cdots = f_{T_{pm}X_p} & \\
(5) & f_{uv} \ge 0 & uv \in E
\end{array}
$$

such that:

- $f_{S_{pi}T_{pr}} \in \mathbf{N}$, for any $i = 1 \dots n$, $p = 1 \dots k$ and r such that $T_r \in S_i$
- $f_{T_{pj}X_p} \in \mathbf{N}$, for any $p = 1 \dots k$ and $j = 1 \dots m$
- all other flow variables $\in \mathbf{R}$.

Relations (3) and (4) assure that for any $v \neq Y_1$ the flow $f_{uv} \in \{0, c_{uv}\}$. Therefore the flow of each edge in the network is calculated such that to satisfy the flow constraint and flow conservation properties. Note that in every component G_p, all $f_{T_{pj}X_p}$ needs to have the same value 1 or 0, therefore $f_{X_pY_2}$ has the value of m or 0, and f_{Y_2} is a multiple of m.

Next we present the Maximum Covers using Mixed Integer Programming (*MC-MIP*) heuristic, which computes the covers based on the solution f_{Y_2} returned by the MIP:

MC-MIP Heuristic:

1. compute f_{Y_2} using MIP

2. $\alpha = f_{Y_2}/m$; $h = 0$

3. for each $p = 1 \dots k$

4. if $(f_{X_pY_2} \neq 0)$

5. $h++$; $C_h = \phi$

6. for each $i = 1 \dots n$

7. if $(f_{S_{0i}S_{pi}} \neq 0)$ then $C_h = C_h \cup S_i$

8. endfor

9. endfor

10. return the disjoint covers $C_1, C_2, \dots, C_\alpha$

Our heuristic, *MC-MIP*, uses the output of the *MIP* to compute the disjoint set covers. Recall that k is the number of components. Lines 1 ... 9 set α, the number of disjoint covers and construct the covers $C_1, C_2, \dots, C_\alpha$. The complexity of our heuristic is dominated by the complexity of the *MIP*.

In the example in figure 1(b), we present the flow assignment for each edge in the flow network. In this case $\alpha = 2$ and there are two covers $C_1 = \{S_1, S_3\}$ and $C_2 = \{S_2\}$.

4. Performance evaluation

In this section we evaluate the performance of the MC-MIP heuristic, designed to compute the disjoint set covers. We simulate a stationary network with sensor nodes and target points randomly located in a 500 m × 500 m area. We assume the transmission range is equal for all the sensors in the network. To solve the mixed integer programming *MIP*, we used the Optimization Solutions Library (OSL) [7] software developed by IBM. The method we used in our code is *branch and bound*.

We compare the number of covers produced by our heuristic with the number of covers produced by the most constrained-minimally constraining heuristic proposed by Slijepcevic and Potkonjak in [10], which was developed for area monitoring. The area to be monitored is divided into a number of *fields*, such that all points from a field are covered by the same set of sensors. By viewing every field as a target, we directly applied most constrained-minimally constraining heuristic to our problem and compared its performances versus *MC-MIP*. The approach in [10] builds each cover by successively adding the sensors that cover the sparsely covered parts of the area. Priority is giving to the sensors that (1) cover a high number of uncovered areas (2) cover more sparsely covered areas (3) do not cover areas redundantly and (4) redundantly cover the areas which are not sparsely covered. This heuristic has complexity $O(n^2)$ when the number of fields is not considered in the computation and n is the number of sensors in the network.

In the first set of experiments, we consider 10 target points randomly distributed, and we vary the number of sensors between 50–90 with an increment of 5 and the sensing range between 100–300 m with an increment of 20. For every value of the number of sensors and the sensing range, we repeated the experiment 5 times, for different sensor node random positioning.

In figure 2, we present the average number of covers computed by the *MC-MIP* heuristic, depending on the number of sensors and the sensing range. As the number of sensors or the sensing range increases, the number of disjoint covers increases too, since every target would be covered by more sensors.

In Table 1 we consider the measurements for 90 sensor nodes and 10 targets and compare the results produced by *MC-MIP* and the heuristic proposed by Slijepcevic and Potkonjak in [10]. Our heuristic produces consistently more covers, therefore achieving better energy savings. Table 1 shows the running time, in seconds for the *MC-MIP* heuristic. The heuristic in [10] is faster. However this algorithm is executed by the central node only once. Therefore trading off the running time in favor of more disjoint sets may be justified.

Figure 3 compares the average number of covers computed by *MC-MIP* and [10] for networks with 90 sensors and 10 targets. As the transmission range increases, redundancy also

Table 1
Measurements for 90 sensors and 10 targets randomly distributed.

	MIP				Slijepcevic		
Sensor range	Avg. runtime (s)	Min. covers	Avg. covers	Max. covers	Min. covers	Avg. covers	Max. covers
100	0	0	2.4	4	0	2.4	4
120	0.2	3	5.4	7	3	5	7
140	0.2	4	6.6	8	4	6	8
160	1	8	8.6	11	6	7.6	9
180	2.2	6	11.6	15	6	10.2	13
200	4.8	13	15	17	11	12.6	15
220	12.2	16	18.4	21	14	16.8	18
240	17.6	13	19.6	23	13	18.2	21
260	28.6	15	22.2	26	15	20.4	23
280	56.8	21	27	30	21	24.4	27
300	97.2	27	31.4	33	27	29.2	31

grows, reflected in more components in our network flow and therefore more disjoint covers.

In the second set of experiments, we consider between 10–50 target points and between 50–90 sensor nodes randomly distributed with a sensing range of 250 m. For every such set of values, we repeated the experiment 5 times, for different sensor nodes random placements.

Table 2
Measurements for 90 sensors with sensing range of 250 m.

	MC-MIP				Slijepcevic		
Number targets	Avg. runtime (s)	Min. covers	Avg. covers	Max. covers	Min. covers	Avg. covers	Max. covers
10	31.2	17	22.8	27	16	20.8	23
15	50	17	19.4	22	16	18.2	21
20	92	18	20.4	23	16	18.2	19
25	151.2	18	21.2	24	16	18	19
30	179.6	11	19	23	11	16.6	19
35	244.4	16	19.4	22	16	16.8	18
40	278	17	18.4	20	15	16	17
45	504.8	18	20.6	23	15	17.2	20
50	404.8	14	17	21	14	16	18

Figure 4 illustrates the average number of disjoint covers computed by the *MC-MIP* heuristic. As the number of sensors increases, the average number of covers increases, too.

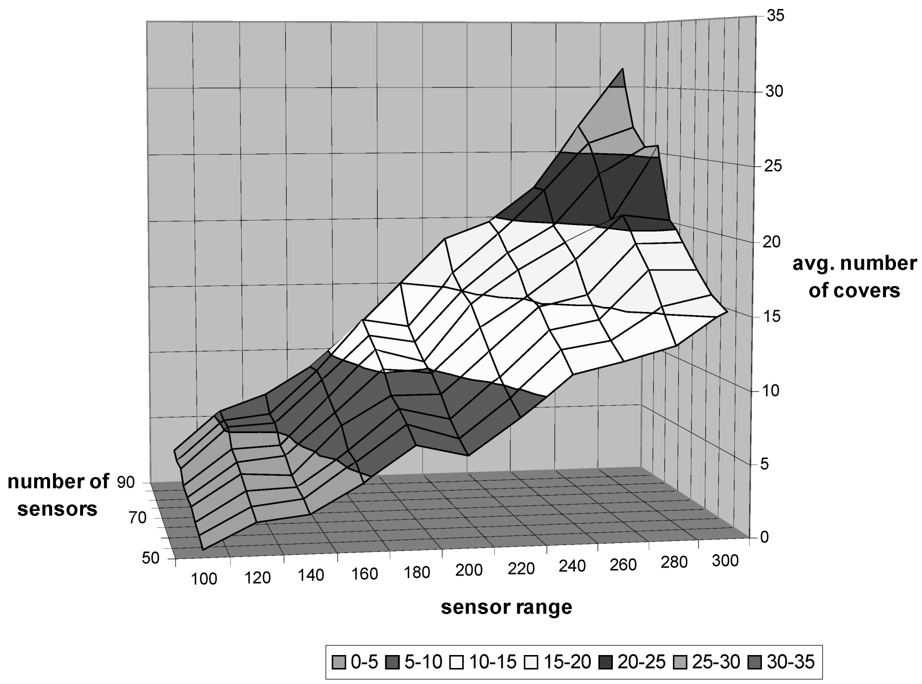


Figure 2. Average number of covers computed by MC-MIP, depending on the number of sensors and range.

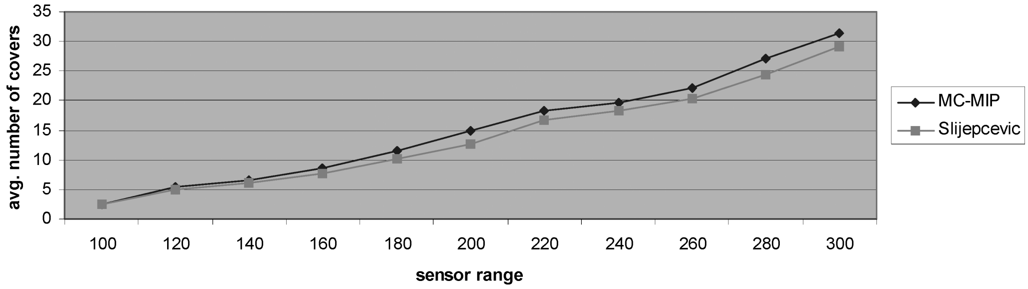


Figure 3. Average number of covers with 90 sensors and 10 targets.

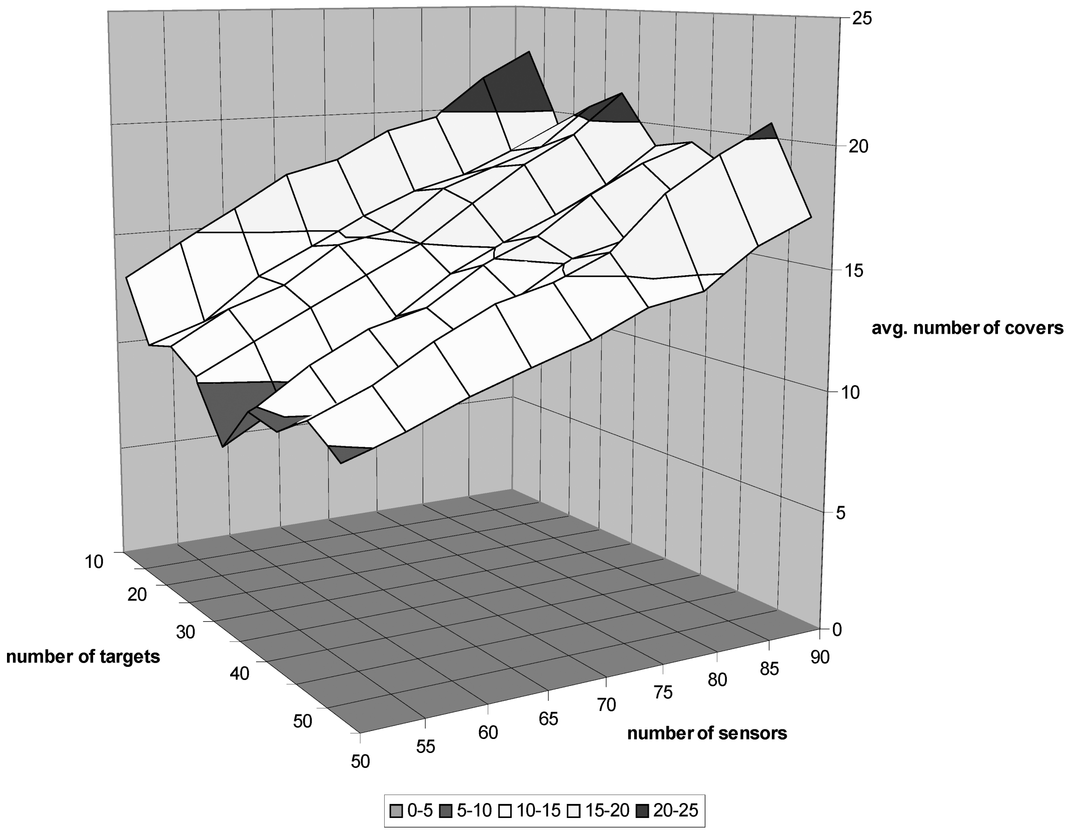


Figure 4. Average number of covers computed by MC-MIP, depending on the number of sensors and number of targets.

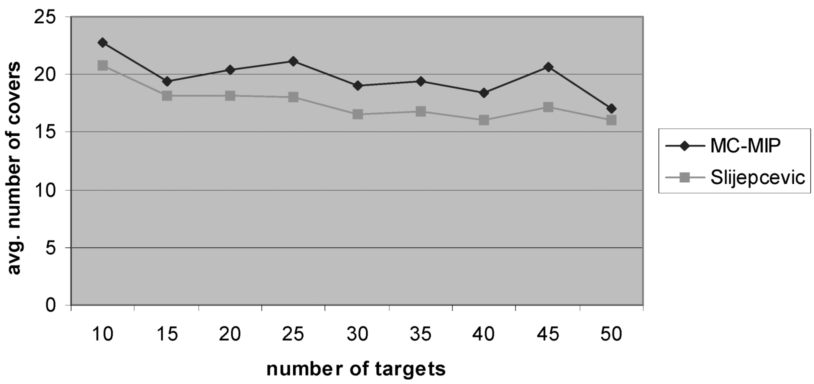


Figure 5. Average number of covers with 90 sensors with sensing range of 250 m.

In Table 2, we present the maximum, average and minimum number of covers computed by *MC-MIP* and the heuristic in [10] for 90 sensors randomly distributed, with a sensing range of 250 m when number of targets vary between 10 . . . 50. The general remark is that the number of covers obtained by *MC-MIP* is larger, but the heuristic in [10] has lower execution time.

Figure 5 compares the number of covers output by *MC-MIP* and the heuristic in [10]. The oscillations in cover numbers occur depending on the sensors and targets random distribution in the 500 m × 500 m given area. As the number of targets grows, the average number of sensors that cover every target decreases, resulting in fewer covers.

5. Conclusion

Wireless sensor networks are battery powered, therefore prolonging the network lifetime through a power aware node organization is highly desirable. An efficient method for energy saving is to schedule the sensor node activity such that every sensor alternates between sleep and active state. One solution is to organize the sensor nodes in disjoint covers, such that every cover completely monitors all the targets. These covers are activated in turn, in a round-robin fashion, such that at a specific time only one sensor set is responsible for sensing the targets, while all other sensors are in a low-energy, sleep state. This problem is modeled as maximum disjoint set covers problem. We presented a theoretical analysis for this problem and proposed an efficient heuristic *MC-MIP* with a mixed integer programming formulation. We evaluated its performance by simulation, against the most constrained—minimally constraining heuristic proposed in [10].

References

[1] ANSI/IEEE, Standard 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY) specifications (1999).

[2] L. Benini, G. Castelli, A. Macii, E. Macii, et al., A discrete-time battery model for high-level power estimation, in: *Design, Automation and Test in Europe Conference* (2000) pp. 35–39.

[3] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li and D.-Z. Du, Wireless sensor networks with energy efficient organization, Journal of Interconnection Networks 3(3/4) (2002) 213–229.

[4] I. Chlamtac, C. Petrioli, and J. Redi, Energy-conserving access protocols for identification networks, IEEE/ACM Transactions on Networking 7(1) (1999) 51–59.

[5] ETSI EN 300 652, V 1.2.1, Broadband Radio Access Networks (BRAN); High Performance Radio Local Area Network (HIPERLAN) Type 1; Functional Specification (July 1998).

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).

[7] http://www-3.ibm.com/software/data/bi/osl

[8] P. Karn, MACA—A New Channel Access Method for Packet Radio, in: *ARRL/CRRL Amateur Radio 9th Computer Networking Conference* (1990) pp. 134–140.

[9] C.S. Raghavendra and S. Singh, PAMAS-Power Aware Multi-Access protocol with Signaling for Ad Hoc Networks, Computer Communications Review (July 1998).

[10] S. Slijepcevic and M. Potkonjak, Power efficient organization of wireless sensor networks, in: *IEEE International Conference on Communications*, Helsinki, Finland (June 2001).

[11] J. Stine and G. de Veciana, Improving energy efficiency of centrally controlled wireless data networks, Wireless Networks 8(6) (2002) 681–700.

Mihaela Cardei is an Assistant Professor in the Department of Computer Science and Engineering at the Florida Atlantic University. Her research interests are in the areas of wireless networking, wireless sensor networks, algorithm and protocol design in communication networks and resource management. Mihaela Cardei received her M.S. and Ph.D. in Computer Science from the University of Minnesota in 1999 and 2003 respectively. Her Ph.D. adviser was Dr. Ding-Zhu Du. During her graduate studies, she worked with Honeywell Laboratories on the Real Time Adaptive Resource Management DARPA project. She is also a recipient of the University of Minnesota Graduate School Doctoral Dissertation Fellowship for 2002–2003.
E-mail: mihaela@cse.fau.edu

Ding-Zhu Du received his M.S. degree in 1982 from Institute of Applied Mathematics, Chinese Academy of Sciences under the supervision of Minyi Yue, and his Ph.D. degree in 1985 from the University of California at Santa Barbara under the supervision of Ronald V. Book. He worked at Mathematical Sciences Research Institute, Berkeley in 1985–86, at MIT in 1986–87, and at Princeton University in 1990–91. Currently, he is a professor at Department of Computer Science, University of Minnesota and also a research professor at Institute of Applied Mathematics, Chinese Academy of Sciences. His research interests include combinatorial optimization, communication networks, and theory of computation. He has published more than 140 journal papers and 30 books. Currently, he is the editor-in-chief of Journal of Combinatorial Optimization and book series on Network Theory and Applications. He is also in editorial boards of eight journals.
E-mail: dzd@cs.umn.edu