

# Energy-Efficient Target Coverage in Heterogeneous Wireless Sensor Networks

Ionut Cardei and Mihaela Cardei  
 Department of Computer Science and Engineering  
 Florida Atlantic University  
 Boca Raton, FL 33431  
 Email: {icardei, mihaela }@cse.fau.edu

**Abstract**— This paper considers a heterogeneous wireless sensor network consisting in several resource-rich supernodes used for data relaying and a large number of energy constrained wireless sensor nodes. Sensor nodes are deployed randomly to monitor a number of targets. Since targets are redundantly covered by more sensors, in order to conserve energy resources, we organize the sensors in set covers that are activate successively. In this paper we introduce the Heterogeneous Connected Set Covers (HCSC) problem that has as objective finding a maximum number of set covers such that each set cover monitors all targets and is connected to at least one supernode. A sensor can participate in multiple set covers, but the total energy spent in all sets is constrained by the battery capacity. In this paper we show that HCSC is NP-complete and we propose two algorithms for solving this problem, an Integer Programming approach and a distributed and localized protocol. Simulation results are presented to evaluate these solutions.

Keywords: heterogeneous wireless sensor networks, energy efficiency, sensor scheduling, target coverage.

## I. INTRODUCTION

Wireless sensor networks (WSNs) constitute the foundation of a broad range of applications related to national security, surveillance, military, health care, and environmental monitoring.

An important issue in sensor networks is power scarcity, driven in part by battery size and weight limitations. Mechanisms that optimize sensor energy utilization have a great impact on prolonging the network lifetime. Power saving techniques can generally be classified in two categories: scheduling the sensor nodes to alternate between active and

sleep mode, and adjusting the transmission and/or sensing range of the wireless nodes. In this paper we deal with the first method. We design a scheduling mechanism in which only some of the sensors are active, while all other sensors are in sleep mode.

In this paper, we study target coverage in heterogeneous WSNs (see Fig. 1) that contains two types of wireless devices: resource-constrained wireless sensor nodes deployed randomly in large number and several resource-rich, predeployed supernodes. The mechanism for data gathering in heterogeneous WSNs works as follows. Sensor nodes transmit and relay measurements. Once data packets encounter a supernode, they are forwarded using fast supernode to supernode communication toward the user application. Additionally, supernodes could process sensor data before forwarding. Hardware components of the heterogeneous WSNs are now available commercially [7].

The main objective in this paper is to address the power scarcity limitation of the wireless sensor nodes. We consider the *target coverage* application where power-constrained sensor nodes are deployed to monitor a set of targets with known locations. The sensor measurements are transmitted to a supernode and from here it uses the fast supernode-to-supernode communication to transmit data to the user application. The method used to extend network lifetime is to organize the sensor nodes into a number of set covers such that all targets are monitored continuously. Additionally, energy constraints for each sensor and connectivity to supernodes must be satisfied. Besides reducing the sensors' energy consume, this method lowers the density of active nodes, thus reducing interference at the MAC layer.

The contributions of this paper are: (1) model the target coverage problem in heterogeneous WSNs by organizing the sensor nodes in set covers; we introduce the Heterogeneous Connected Set Covers (HCSC) problem which is NP-complete, (2) design two algorithms for solving the HCSC problem, one using Integer Programming and the second is a distributed approach, and (3) analyze the performance of our approaches through simulations.

The rest of this paper is organized as follows. In section II we briefly present related works on heterogeneous WSNs and target coverage problem. Section III describes the features of heterogeneous WSNs and introduces the HCSC problem. We continue in sections IV and V with our solutions for solving the HCSC problem. In section VI we present the simulation results, and section VII concludes our paper.

## II. RELATED WORK

The benefits of using heterogeneous WSNs, containing devices with different capabilities, have been presented recently in literature. In [10], it is pointed out that by using a heterogeneous architecture with sensor nodes and gateways, improved network performance are obtained in terms of data gathering delay and network lifetime. In [13], it is reported that properly deployed, heterogeneity can triple the average delivery rate and provide a 5-fold increase in the network lifetime. The work in [11] introduces another type of heterogeneous WSN called actor networks, consisting of sensor nodes and actor nodes which can perform appropriate actions based on sensors' data.

Target coverage is an important application in WSNs. The coverage concept is a measure of the quality of service of the sensing function. The goal is to have each location in the physical space of interest within the sensing range of at least one sensor. The coverage problems can be classified in the following types [5]: (1) area coverage [6], [14] where the objective is to cover an area, (2) point coverage [1], [3], [4], where the objective is to cover a set of targets, and (3) coverage problems that have the objective to determine the maximal support/breach path that traverses a sensor field [?].

An important method for extending network lifetime is to organize the sensor nodes in sets which

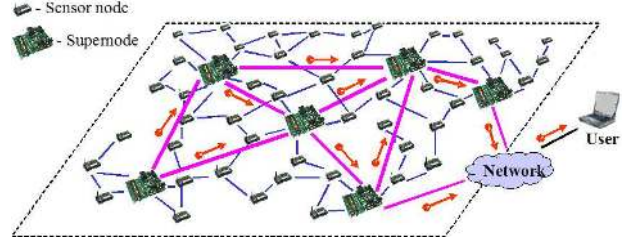


Fig. 1. Heterogeneous Wireless Sensor Networks

are activated successively. Set formation is done based on the problem requirements such as energy-efficiency, area monitoring, connectivity, etc. Different techniques have been proposed in literature [6], [14] for determining which sensors will be active in each round.

The works most relevant to our approach are [4] and [2]. Paper [3] introduces the target coverage problem, where disjoint sensor sets are modeled as disjoint set covers, such that every cover completely monitors all the target points. A mixed integer programming solution is proposed. This problem is further extended in [1], [4], where sensors are not restricted to participation in only disjoint sets, that is, a sensor can be active in more than one set. Paper [1] is the first work that proposed an approximation algorithm for the point coverage problem. Still these works deal only with the coverage requirement, and do not address connectivity.

Recently, the work [2] is concerned with ensuring the connectivity within each set cover. This applies to the case when not all sensors are within communication range of the base station (BS). Ensuring BS-connectivity within each set cover is needed to allow data collecting within each round.

This paper is an extension of the connected set covers problem addressed in [2] to heterogeneous WSNs. Our objective is to efficiently use the benefits of the heterogeneous architecture in order to prolong network lifetime.

## III. PROBLEM DEFINITION

### A. Heterogeneous Network Architecture

We consider a heterogeneous sensor network consisting of two-types of wireless devices: resource-constrained wireless sensor nodes and resource-rich "supernodes", as illustrated in Fig. 1.

Sensor nodes have low cost, limited battery power, short transmission range, low data rate (up to several hundred Kbps) and a low duty cycle. The main tasks performed by a sensor node are sensing, data processing, and data transmission/relaying. Supernodes have two radio transceivers, one for communicating with sensor nodes and the other for communicating with other supernodes. Supernodes are more expensive, have more power reserves, higher data rate, and better processing and storage capabilities than sensor nodes. The main task performed by a supernode is to relay data from sensor nodes to the user application.

### B. HCSC Problem Definition

Let us consider a heterogeneous WSN consisting of  $N$  sensors  $s_1, s_2, \dots, s_N$  and  $M$  supernodes  $g_1, g_2, \dots, g_M$ , with  $M \ll N$ . The supernodes are pre-deployed in the sensing area, they are connected, and their main task is to relay data from sensor nodes to the user application. On the other hand, sensor nodes are deployed randomly in the area of interest to continuously monitor  $T$  targets  $t_1, t_2, \dots, t_T$ . We assume there exists a path from any sensor node to a super-node.

Each sensor has an initial energy  $E$ , communication range  $R_c$  and sensing range  $R_s$  (usually  $R_c \geq R_s$ ). A sensor covers a target if the Euclidean distance between the sensor and the target is less than or equal to  $R_s$ . Additionally, a sensor can communicate with another sensor or with a supernode if the Euclidean distance between them is less than or equal to  $R_c$ .

In order to conserve sensor energy resources and thus to prolong the network lifetime, we schedule the sensor nodes activity to alternate between sleep and active mode. The set of active sensors must satisfy two application requirements: *coverage* and *connectivity*.

We model the data gathering requirement as the requirement to send the sensed data to at least one supernode. We consider that, once the sensed data reach a supernode, the supernode would relays the data to the user application using supernode to supernode communication.

First, all targets must be continuously *covered* by the set of active sensors. Secondly, the monitoring sensors must be *connected* to supernodes. More

specifically, there must be a path of active sensors between each monitoring sensor and at least one supernode. The formal definition is given below:

#### Definition 1: Target Coverage Problem in Heterogeneous WSNs

Given  $T$  targets with known location and an heterogeneous WSN with  $M$  supernodes and  $N$  energy-constrained sensors that are randomly deployed in the targets' vicinity, schedule the sensor nodes' activity such that (1) all targets are continuously monitored, (2) each active sensor is connected to at least one supernode, and (3) network lifetime is maximized.

We measure the network lifetime as the time interval that all  $T$  targets are monitored by a subset of sensor nodes that are connected to supernodes through active sensors, while satisfying the sensor energy constraint. The approach that we used in this paper for maximizing network lifetime is to organize sensors in set covers. The network activity is organized in rounds, such that each set cover is active in one round. Each round takes  $\delta$  time units, and only the sensors in the active set cover are responsible for targets monitoring and data relaying, while all other sensors are in sleep mode.

A heterogeneous topology with 20 targets, a 2 x 2 supernode grid and 50 motes organized in a cover is shown in Figure 2.

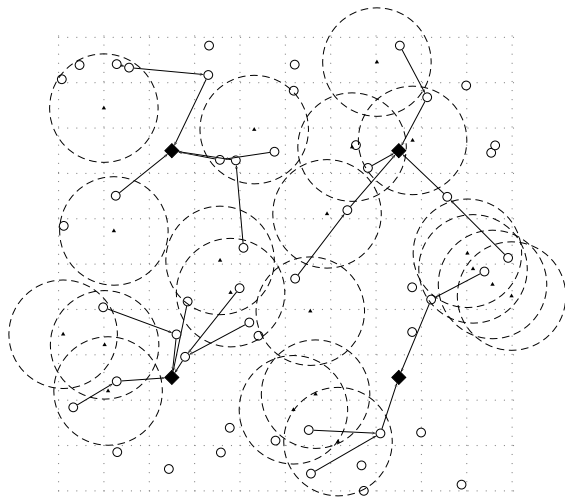


Fig. 2. Heterogeneous WSN topology with a cover for 20 targets (triangles=targets, circles=motes, diamonds=supernodes).

Next, we formally define the Heterogeneous Connected Set Covers (HCSC) problem that we

used to solve the target coverage problem in heterogeneous WSNs.

**Definition 2: HCSC Problem**

Given a set of targets  $t_1, t_2, \dots, t_T$ , a set of supernodes  $g_1, g_2, \dots, g_M$ , and a set of randomly deployed sensors  $s_1, s_2, \dots, s_N$ , find a family of sensor set covers  $c_1, c_2, \dots, c_P$ , such that (1)  $P$  is maximized, (2) sensors in each set cover  $c_p$  ( $p = 1, \dots, P$ ) are connected to supernodes, (3) each sensor set monitors all targets, and (4) each sensor appearing in the sets  $c_1, c_2, \dots, c_P$  consumes at most  $E$  energy.

In HCSC definition, the requirement to maximize  $P$  is equivalent with maximizing the network lifetime. Other requirements include targets coverage by the active sensor set, active sensor sets connectivity to supernodes, and satisfying the sensor energy constraints. Paper [2] introduces the Connected Set Covers (CSC) problem that considers homogeneous sensor networks with only one supernode (Base Station) for data collecting and shows that CSC is NP-complete.

HCSC problem is NP-complete by restriction method [8], since CSC is a particular case of HCSC problem for  $M = 1$ , that is the case when we have only one supernode deployed for data gathering.

Let us next formulate the framework of our solutions. We consider the network activity is organized in rounds, and each round has two phases: *initialization* and *data collection*. During the initialization phase, a set of active sensors (let us say the set cover  $c_i$ ) is established such that conditions 2, 3, and 4 in the HCSC problem are satisfied. During the data collection phase, sensors in the set cover  $c_i$  are active while all other sensors are in the sleep mode for the rest of the round and they will wake-up for the next initialization phase.

Sensor nodes active in a set cover  $c_i$  are classified as *sensing nodes* and *relay nodes*. Sensing nodes are sensors that monitor one or more targets. They consume energy both for sensing and for data transmission. Relay nodes are sensors that are active to relay data from sensing nodes to supernodes.

In this paper, we consider the following data gathering model. We assume that data gathering is performed periodically, with a sensing cycle of  $\delta'$  time. That means that every  $\delta'$  time every sensing node will send a packet with sensed data. We also

consider that each round (each set cover) is active  $\delta = p \times \delta'$  time. Then  $p$  packets will be generated by each sensing node during each round.

A sensing node consumes  $E_s$  energy per round for sensing. A relaying node consumes  $E_r$  energy for receiving a packet and  $E_t$  energy for transmitting a packet. We do not account for the energy spent on supernodes, since we consider that supernodes have enough energy resources.

The energy model considered in this paper is similar to the model used for LEACH [9]. The energy spent for transmissions is  $E_t(l, R_c) = E_{elec}l + \epsilon_{amp}lR_c^2$ , where  $l$  is the packet length (bits),  $\epsilon_{amp}$  is a transmit amplifier parameter and  $R_c$  is the transmission range. The receiver energy for a packet is  $E_r = E_{elec}l$ .  $E_{elec} = 50\text{nJ/bit}$  is the energy dissipated per bit by the radio electronics for Rx/Tx.

Next, we propose two solutions for determining the active sensors (sensing nodes and relay nodes) in each round: in section IV we present a centralized IP-based algorithm and in section V we propose a distributed approach.

#### IV. INTEGER PROGRAMMING SOLUTION FOR THE HCSC PROBLEM

We first formulate the *HCSC* problem using Integer Programming (IP) in section IV-A and then propose an centralized, IP-based algorithm in section IV-B.

##### A. Integer Programming Formulation of the HCSC Problem

In order to model the connectivity requirement for each set cover  $c_k$ , for  $k = 1, \dots, K$ , we model the heterogeneous WSN as a flow network  $G = (V, E)$ . The vertex set  $V = \{n_1, n_2, \dots, n_M, n_{M+1}, \dots, n_{M+N}\}$  contains the nodes in the heterogeneous WSN. The first  $M$  nodes in  $V$  are the supernodes and the last  $N$  nodes are the sensor nodes. Please note that when we refer in general to a *node*  $n_i$ , this means that  $n_i$  can be either a supernode or a sensor node. If we specify the index  $i$  as being between  $1 \dots M$  then, we refer to a supernode. If  $i > M$  then  $n_i$  refers to a sensor node.

We assume all sensor nodes have the same communication range  $R_c$ . We define the neighborhood of a node  $n_u$  as  $\Gamma(n_u) = \{n_v \in V | \text{dist}(n_u, n_v) \leq$

$R_c$ }. Then we define the directed edge set  $E = \{(n_u, n_v) | n_u, n_v \in V \text{ and } n_v \in \Gamma(n_u)\}$ . Since we assumed a connected heterogeneous WSN, the graph  $G$  is strongly connected as well.

In order to model the connectivity of each sensing node to a supernode, we use the flow concept, where flow enters the network through the sensing nodes and is collected at supernodes. Each sensing node is a source that inserts one unit of flow in the network. Each unit of flow represents a packet that has to be forwarded by the sensor nodes visited by that flow. We note with  $f_{vuk}$  the flow sent from node  $v$  to node  $u$  in the set cover  $c_k$ , where  $f : V \times V \times \{0, \dots, K\} \rightarrow Z$  and  $f_{vuk} \geq 0$ .  $K$  is an upperbound for the number of set covers. The flow values and the source vertices vary with each set cover  $c_k$ .

Given:

- $M$  supernodes  $n_1, n_2, \dots, n_M$  and  $N$  sensor nodes  $n_{M+1}, \dots, n_{M+N}$ .
- the neighborhood of each node  $n_u$  (sensor node or supernode) is computed as  $\Gamma(n_u) = \{n_v \in V | \text{dist}(n_u, n_v) \leq R_c\}$ .
- $T$  targets  $t_1, t_2, \dots, t_T$ .
- $E$ , the initial energy of each sensor and  $p$  the number of data packets sent by a sensing node per round. Each sensing node consumes  $E_s$  energy per round for sensing. Each sensor node consumes  $E_r$  energy per packet received and  $E_t$  energy to transmit a packet.
- the coefficients showing the coverage relationship between sensors and targets:  $a_{uj} = 1$  if sensor  $s_u$  covers the target  $t_j$ , that means  $\text{dist}(s_u, t_j) \leq R_s$ , where  $R_s$  is the sensing range of each sensor node. Otherwise,  $a_{uj} = 0$ .

Variables:

- $c_k$ , boolean variable, for  $k = 1, \dots, K$ ;  $c_k = 1$  if this subset is a valid connected set-cover, otherwise  $c_k = 0$ .
- $x_{uk}$ , boolean variable, for  $u = M+1, \dots, M+N$ , and  $k = 1, \dots, K$ ;  $x_{uk} = 1$  if sensor  $u$  is a sensing node in the set cover  $c_k$ , otherwise  $x_{uk} = 0$ .
- $f_{vuk} \in [0, T]$ , integer variable, for any  $s_v \in \{n_{M+1}, \dots, n_{M+N}\}$ ,  $s_u \in \Gamma(n_v)$ , and  $k = 1, \dots, K$ .

Integer Programming:

Maximize  $c_1 + c_2 + \dots + c_K$

Subject to:

- $\sum_{k=1}^K (x_{uk} \cdot E_s + p(\sum_{n_v \in \Gamma(n_u)} f_{vu})E_r + p(\sum_{n_w \in \Gamma(n_u)} f_{uw})E_t) \leq E$  for all  $u = M+1, \dots, M+N$  (1)
- $\sum_{u=M+1}^{M+N} x_{uk} \cdot a_{uj} \geq c_k$  for all  $j = 1, \dots, T$ ;  $k = 1, \dots, K$  (2)
- $c_k \geq x_{uk}$  for all  $u = M+1, \dots, M+N$ ;  $k = 1, \dots, K$  (3)
- $\sum_{u=1}^{M+N} f_{vuk} \leq \sum_{w=M+1}^{M+N} x_{wk}$  for all  $v = M+1, \dots, M+N$ ;  $k = 1, \dots, K$  (4)
- $\sum_{s_w \in \Gamma(s_u)} f_{uwk} - \sum_{s_v \in \Gamma(s_u)} f_{vuk} = x_{uk}$  for all  $u = M+1, \dots, M+N$ ;  $k = 1, \dots, K$  (5)
- $\sum_{u=1}^M \sum_{v=M+1}^{M+N} f_{vuk} = \sum_{w=M+1}^{M+N} x_{wk}$  for all  $k = 1, \dots, K$  (6)
- $\sum_{u=1}^M \sum_{v=1}^{M+N} f_{vuk} = 0$  for all  $k = 1, \dots, K$  (7)
- $\sum_{s_v \notin \Gamma(s_u)} f_{vuk} = 0$  for all  $u = M+1, \dots, M+N$ ;  $k = 1, \dots, K$  (8)

where:

- $c_k, x_{uk} \in \{0, 1\}$  for  $u = M+1, \dots, M+N$ ;  $k = 1, \dots, K$
- $f_{vuk} \in \{0, 1, 2, \dots, T\}$  for  $u, v = 1, \dots, M+N$ ;  $k = 1, \dots, K$

Remarks:

- 1) Constraint (1) guarantees that the energy consumed by each sensor node  $s_u$ ,  $u = M+1, \dots, M+N$  is less than or equal to  $E$ , which is the starting energy of each sensor.
- 2) The second constraint (2) guarantees that for each valid set cover  $c_k$  (that is  $c_k = 1$ ), each target is covered by at least one sensing node.
- 3) Constraint (3) assures that if a set-cover is invalid ( $c_k = 0$ ) then no sensor participates in sensing.
- 4) Constraint (4) upperbounds the out-flow of a sensor node to the maximum number of flow units inserted in the network.
- 5) Constraint (5) specifies that one unit of flow is inserted in each sensing node. Also, the sensor nodes verifies the flow conservation property.
- 6) Constraint (6) specifies that supernodes are the sinks, where all the flow is collected.
- 7) Constraint (7) assures that there is no out-flow for any supernode.

- 8) Constraint (8) assures that for any sensor node, the flow outside its neighborhood is 0. This is because a sensor node cannot communicate with other nodes outside its communication range.
- 9)  $K$  represents an upper bound for the number of covers.

### B. Integer Programming-based Algorithm

In this subsection, we propose a solution for the HCSC problem, using the Integer Programming (IP) formulation in subsection IV-A. Based on the solution returned by the IP solver ( $\bar{x}_{uk}$ ,  $\bar{f}_{vuk}$ , and  $\bar{c}_k$ ), our algorithm constructs the connected set-covers. For each set cover  $\bar{c}_k$ , we check if this is a valid set cover, that is if  $\bar{c}_k = 1$ . If the equality is true, then a new set cover  $(S_h, R_h)$  is formed. The set  $S_h$  (set  $R_h$ ) contains the sensing sensors (relay sensors respectively) in the set cover  $h$ . All other sensors are in the sleep mode.

For each sensing or relay node  $s_u$  in the set cover  $h$ , we record the in-neighborhood  $\Gamma_{in}^h(s_u)$ , given by the in-flows, which contains pairs  $(s_v, p_v)$  meaning that  $s_u$  receives  $p_v$  packets from  $s_v$  per sensing cycle  $\delta'$ . Similarly, the out-neighborhood  $\Gamma_{out}^h(s_u)$  is given by the out-flows and contains pairs  $(s_w, p_w)$  meaning that  $s_u$  sends  $p_w$  packets to  $s_w$  per sensing cycle  $\delta'$ .

The complexity of our algorithm is dominated by the complexity of the Integer Programming solver. The network lifetime is computed as  $\delta * h$ , with  $h$ -number of set covers and  $\delta$ -time each set cover is active. Simulations results are presented in section VI.

### IP-HCSC

- 1: solve the IP from subsection IV-A and get the solution  $\bar{x}_{uk}$ ,  $\bar{f}_{vuk}$ , and  $\bar{c}_k$
- 2:  $h \leftarrow 0$
- 3: **for all**  $k \leftarrow 1$  to  $K$  **do**
- 4:   **if**  $\bar{c}_k = 1$  **then**
- 5:      $h \leftarrow h + 1$ ;  $S_h \leftarrow \phi$ ;  $R_h \leftarrow \phi$ ;
- 6:     **for all**  $i \leftarrow M + 1$  to  $M + N$  **do**
- 7:       **if**  $\bar{x}_{ik} = 1$  **then**
- 8:          $S_h \leftarrow S_h \cup \{s_i\}$
- 9:       **end if**
- 10:      $in\_packets \leftarrow 0$ ;  $\Gamma_{in}^h(s_i) \leftarrow \phi$ ;
- 11:      $\Gamma_{out}^h(s_i) \leftarrow \phi$ ;
- 11:     **for all**  $s_v \in \Gamma(s_i)$  **do**

- 12:       **if**  $\bar{f}_{vik} > 0$  **then**
- 13:          $in\_packets \leftarrow in\_packets + 1$
- 14:          $\Gamma_{in}^h(s_i) \leftarrow \Gamma_{in}^h(s_i) \cup (s_v, \bar{f}_{vik})$
- 15:       **end if**
- 16:     **if**  $\bar{f}_{ivk} > 0$  **then**
- 17:          $\Gamma_{out}^h(s_i) \leftarrow \Gamma_{out}^h(s_i) \cup (s_v, \bar{f}_{ivk})$
- 18:     **end if**
- 19:   **end for**
- 20:   **if**  $in\_packets > 0$  **then**
- 21:      $R_h \leftarrow R_h \cup \{s_i\}$
- 22:   **end if**
- 23:   **end for**
- 24: **end if**
- 25: **end for**
- 26: return the connected set covers  $(S_1, R_1), (S_2, R_2), \dots, (S_h, R_h)$  as well as the in- and out-neighborhoods for each set cover.

## V. A DISTRIBUTED ALGORITHM FOR THE HCSC PROBLEM

Our distributed algorithm designed to select the set-cover  $c_i$  for the round  $i$  has two steps: (1) sensing nodes selection and (2) relay nodes selection.

The algorithm for selecting the sensing nodes is presented in section V-A, followed by the algorithm for selecting the relay nodes in section V-B.

### A. Algorithm for Selecting Sensing Nodes

The algorithm for selecting the sensing nodes is *distributed and localized*, that means the decision process at each node makes use of only information for a neighborhood within a constant number of hops. A distributed and localized approach is desirable in sensor networks since it is scalable and adapts better to dynamic and large topologies.

In this section we describe how a sensor  $s_u$  decides whether or not it will be a *sensing node* during the current round. Let us consider the following notations:

- $E'_u$  is the residual energy of  $s_u$  and  $E$  is the initial energy
- the set  $M_u$  contains all the targets located within the sensing range of  $s_u$
- the set  $TARGETS_u$  is maintained by  $s_u$  and contains all the targets in  $s_u$ 's sensing range that are not covered by any node that has

declared and advertised itself as a *sensing node* until now.

- $T$  is the total number of targets

Let us consider that sensing node selection takes  $W$  time. Sensor node  $s_u$  computes a back-off time  $T_u \leq W$ . If  $s_u$  has the residual energy  $E'_u < E_s + pE_t$ , then it does not have sufficient energy to become a sensing node and  $T_u = W$ .

Otherwise,  $T_u$  is computed as  $T_u = (1 - \alpha \frac{E'_u}{E} - \beta \frac{|TARGETS_u|}{T})W$ , where  $\alpha$  and  $\beta$  are parameters used to decide the weight of residual energy and the weight of the number of uncovered targets in computing the back-off time,  $\alpha + \beta < 1$ . Parameters  $\alpha$  and  $\beta$  are initialized at the beginning of the application and do not change during the application lifetime.

The rationale of this formula is to give higher priority (smaller  $T_u$ ) to sensors that have higher residual energy and cover a larger number of uncovered targets.

When  $T_u$  expires, if  $TARGETS_u \neq \emptyset$  and  $E'_u \geq E_s + pE_t$ , then  $s_u$  declares itself as a *sensing node* during the current round. Additionally,  $s_u$  broadcasts this decision together with the set  $M_u$  to its 2-hop neighbors. When a node  $s_v$  receives such an advertisement message, it updates its  $TARGETS_v$  set and  $T_v$  timer accordingly. On the other hand, if  $TARGETS_u$  becomes empty, then  $s_u$  will not be a sensing node in this round.

Sensors' broadcasts in their local neighborhood are serialized by different waiting times, which also gives priority to the sensors with higher residual energy that cover more uncovered targets. Since the 2-hops advertisement messages are very small, we neglect the energy consumed in forwarding them.

If, when  $T_u$  expires,  $TARGETS_u \neq \emptyset$  and  $E'_u < E_s + pE_t$ , then there are targets that cannot be covered in the current round, and  $s_u$  sends this failure notification to one or more supernodes.

Next, we present the **Decide Sensing Status** procedure that is run by each sensor  $s_u$ ,  $u = 1, \dots, N$ :

#### Decide Sensing Status( $s_u, \alpha, \beta$ )

- 1: initialize the set  $M_u$  and set  $TARGETS_u \leftarrow M_u$
- 2: **if**  $E'_u \geq E_s + pE_t$  **then**
- 3: compute waiting time  $T_u \leftarrow (1 - \alpha \frac{E'_u}{E} - \beta \frac{|TARGETS_u|}{T})W$ , and start timer  $t$

- 4: **else**
- 5:  $T_u \leftarrow W$ , and start timer  $t$
- 6: **end if**
- 7: **while**  $t \leq T_u$  and  $TARGETS_u \neq \emptyset$  **do**
- 8: **if** message from neighbor sensor is received **then**
- 9: update  $TARGETS_u$ , by removing the targets now covered by the advertising sensing node; update the back-off timer  $T_u$
- 10: **if**  $TARGETS_u = 0$  **then**
- 11: return;
- 12: **end if**
- 13: update the waiting time  $T_u$
- 14: **end if**
- 15: **end while**
- 16: **if**  $E'_u < E_s + pE_t$  **then**
- 17:  $s_u$  reports failure to one or more supernodes, indicating the targets it cannot cover due to energy constraints
- 18: **else**
- 19:  $s_u$  will be a *sensing node* in this round;  $s_u$  broadcast to its 2-hop neighbors its status and the set  $M_u$
- 20: **end if**
- 21: return

#### B. Algorithm for Selecting Relay Nodes

The goal of this algorithm is to activate a minimum number of relay nodes in order to satisfy the supernode connectivity requirement for each sensing node.

The relay nodes selection mechanism is initiated by the sensing nodes, which were established as explained in the section V-A. Each sensing node  $s_u$  is responsible for activating relay nodes needed to achieve a communication path to one of the supernodes. This mechanism is a three step protocol *DISC\_REQ* / *DISC\_REPLY* / *RELAY\_SETUP*, as follows.

First, a sensing node  $s_u$  needs to discover a valid path to a closest supernode. For this,  $s_u$  locally broadcasts a message *DISC\_REQ*( $s_u, max\_hops, hops=0$ ). Each sensor with residual energy at least  $p(E_r + E_t)$  receiving a *DISC\_REQ* message, increases the value of the *hops* field ( $hops = hops + 1$ ) and forwards a copy of the message if  $hops \leq max\_hops$ . Please note that a relay node will consume  $p(E_r + E_t)$  energy per round for forwarding

data generated by a sensing node.

Any supernode  $g_i$  receiving a  $DISC\_REQ(s_u, max\_hops, hops)$  message replies back with a  $DISC\_REPLY(g_i, s_u, \#hops \text{ between } s_u \text{ and } g_i)$  message. This reply is sent along the temporary reverse links set-up during the request.

The  $max\_hops$  value can be computed as follows. If sensors know the supernodes location, then  $s_u$  can estimate the  $max\_hops$  value. If  $s_u$  does not receive on time any  $DISC\_REPLY$  message from at least one supernode, then  $max\_hops$  value is increased and the process is repeated. If  $s_u$  does not know the location of its closest supernode, then we can use the *expanding ring search* mechanism [12]. In this mechanism, smaller  $max\_hops$  values are tried first, and if no  $DISC\_REPLY$  message is received on time, then the  $max\_hops$  value is increased and the neighbor discovery process is repeated.

If the sensor  $s_u$  receives more  $DISC\_REPLY$  messages, then will select a message with the minimum number of hops to a supernode  $g_i$ . Then  $s_u$  will send a message  $RELAY\_SETUP(s_u, g_i, hops = \#hops \text{ between } s_u \text{ and } g_i)$  towards  $g_i$ , using the temporary reverse links set-up during  $DISC\_REPLY$  transmission. All forwarding nodes decrement the  $hops$  field and change their status to *relay nodes* during the current round.

When a sensor becomes relay node, it reserves  $p(E_r + E_t)$  energy from its residual energy. Please note that a sensor might serve as relay node for multiple sensing nodes, but then it must reserve  $p(E_r + E_t)$  energy for each sensing node data transmission.

An alternative protocol for selecting relay nodes could use a different metric instead of hop count. Residual energy may be a good choice. Intermediary motes on the route back from the supernode report in  $DISC\_REPLY$  messages their current energy level. Upstream motes, including the sensing mote, set up forward pointers to motes with the highest energy levels.

## VI. SIMULATION RESULTS

In this section, we evaluate the performance of the *IP-HCSC* algorithm from section IV and the distributed algorithm *Distr-HCSC* from section V.

We simulate a stationary network with sensor nodes and target points located randomly in a  $500m \times 500m$  area.

Additionally, we consider the following parameters:

- initial battery energy of each sensor is 2700J. The power used for sensing is 1.73mW (acoustic sensor) and the power dissipated by the CPU is 24mW.
- the sensing range is 50m or 60m and the radio transmission range is 100 m
- the data rate is 38400 bps and the measurement packet size is 36 bytes (20 bytes preamble + 16 bytes payload)
- the sensing round duration is 10 or 20 days
- the time interval between two consecutive measurement reports from active sensing motes (the sensing cycle) is 10 seconds.
- the motes and targets have uniformly distributed random positions, while the supernodes are positioned in rectangular grids (1, 2 x 2, 3 x 3 etc.)
- for sensing node selection (V-A), parameters  $\alpha = 0.5, \beta = 0.2$ .

The mote energy model was described in section III-B. For each experiment we keep the sensing range and the communication range constant. All measurements are averaged over 10 runs with different random node/target placement. The performance metric we focus on is the number of covers computed by the IP solution and the distributed protocol. This is equal to the number of successive rounds full target coverage is guaranteed. This is an indicator for network lifetime.

For simulations we consider the following variables:

- $M$ , the number of supernodes. We vary the size of the supernode grid between  $1 \times 1$  and  $4 \times 4$  to study the impact of the additional supernode connectivity on network lifetime. When  $M = 1$  this corresponds to a traditional WSN, with one sink.
- $N$ , the number of sensor nodes. We vary the number of randomly deployed sensor nodes between 50 and 100 to study the effect of node density on performance.
- $T$ , the number of targets to be covered. We vary the number of targets between 1 and 50.



We used the *CPLEX* optimization library to implement the IP solution. The localized and distributed protocol *Distr-HCSC* was implemented with a custom C++ packet-level heterogeneous WSN simulator. To compare the two algorithms we assumed the energy spent for control messages between two sensing rounds to be negligible compared with the energy spent during a sensing round  $\delta$  (10 or 20 days). We also assumed a reliable communication channel between neighbor nodes. As part of our future work we will improve the protocol to cope with a non-ideal communication channel.

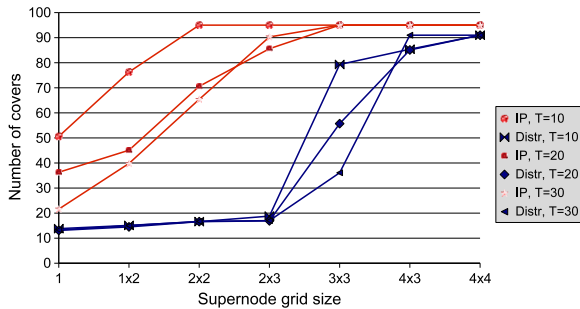


Fig. 3. Number of covers depending on supernode population.

In the first experiment, illustrated in Figure 3, we vary the size of the supernode grid, from  $1 \times 1$  to  $4 \times 4$ . 50 sensor motes cover 10, 20 and 30 targets, in three scenarios. The graph shows the number of covers produced by the two solutions with a varying supernode grid size.

The first observation is that using supernode topologies can greatly increase the network lifetime. We notice the common upper bound of 95 covers reached by the IP approach for all three target populations. The distributed protocol exhibits a similar common upper bound at 91 covers regardless of supernode count. The number of covers becomes limited by the energy used for sensing, and additional connectivity from additional supernodes does not help.

We note the slow improvement in network lifetime for the distributed protocol up to a  $2 \times 3$  supernode grid, followed by a sharp improvement for a  $4 \times 3$  grid (12). This is due to a reduced load (in-flow) for bottleneck motes which are close to a supernode. Beyond the  $2 \times 3$  threshold supernode connectivity relieves the communication energy

load on nearby sensors, providing additional energy reserves more covers.

The graph also shows that for a high supernode density, the distributed protocol can achieve performance close to the optimum.

Figure 4 shows how the number of covers varies for the two algorithms when the sensor node population grows from 40 to 120. The network has a  $2 \times 2$  supernode grid and the number of targets is 10. We note a gradual increase in cover number for both algorithms. The number of covers for the distributed protocol is consistently above 70% of the value computed with the IP algorithm.

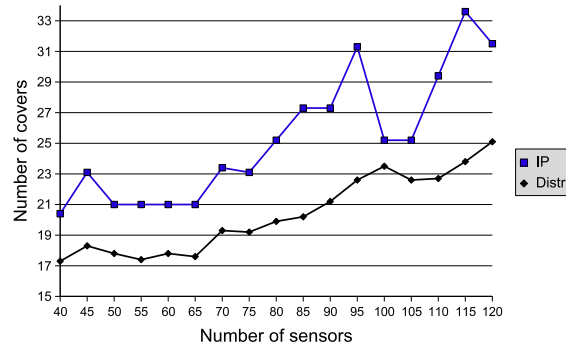


Fig. 4. Number of covers depending on mote population size.

Figure 5 illustrates the number of covers in an experiment where the target count varies from 1 to 50 and the number of sensor nodes is 50. The graph shows the performance for  $1 \times 1$  and  $2 \times 2$  supernode topologies for the two solutions. The initial sharp drop in network lifetime when the target count is in the  $[1, 10]$  range is caused by the additional sensing and communications load on a growing imposed on constant number of motes. The number of covers levels off beyond 15 or 20 targets because of per target coverage saturation. A higher number of targets will not cause additional overhead since: 1. each active sensing node sends only one packet per measurement for all targets within its sensing range, and 2. a mote is activated for sensing when it has at least one uncovered target in its sensing range. The  $2 \times 2$  supernode topology has better performance since the supernodes will reduce the input flow on bottleneck relay motes, typically those close to supernode. We notice that, on average, the distributed protocol yields a cover

count higher than 70% of the optimal value computed with the IP algorithm. We also notice that employing a  $2 \times 2$  supernode grid increases the network lifetime with 20-30% for both algorithms when compared to a one sink topology.

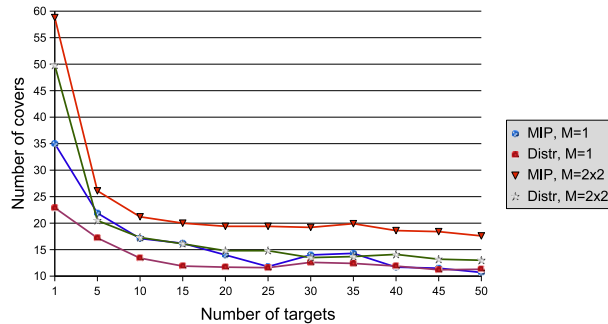


Fig. 5. Number of covers depending on target count.

The simulation results can be summarized as follows:

- using a heterogeneous architecture with a supernode grid greatly improves the network lifetime.
- the supernode grid must exceed a size threshold ( $3 \times 3$  in our simulations) to get the best improvement in cover number
- for dense supernode grids, the performance of the distributed protocol solution comes within 4% to the optimal value
- doubling the mote population increases the number of covers with 15% for the distributed protocol and with 24% for the IP algorithm. Better lifetime improvement can be achieved by deploying a supernode grid.
- the impact of the number of targets on cover count (network lifetime) is much higher in the lower range (1 -15). With a higher target count, sensors will cover more than one target.

## VII. CONCLUSIONS

In this paper we describe approaches for improving sensor network lifetime in heterogeneous wireless sensor networks. We formulate the heterogeneous connected set covers (HCSC) problem, addressing point coverage in heterogeneous WSNs. The HCSC problem has the objective to determine a HWSN setup that maximizes network lifetime provided: a) all targets are covered by at least a

sensor, b) sensor energy resources are constrained, and, c) active sensors are connected to at least one supernode through a multihop path. We proposed two approaches for solving this problem, using Integer Programming, and a distributed and localized protocol that can be practically implemented in a real network. Simulation results have shown that the distributed protocol exceeds 70% of the optimal number of covers, as computed with the IP solution, in scenarios where the supernode grid exceeds a size threshold.

The distributed protocol can be further improved to deal with a non-ideal communication channel. It is also of interest to study how the data gathering communication pattern (periodic or event-based) affects the network lifetime.

## REFERENCES

- [1] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, Power efficient monitoring management in sensor networks, in *IEEE WCNC*, 2004.
- [2] M. Cardei et. al., Energy-Efficient Connected-Coverage in Wireless Sensor Networks, under review.
- [3] M. Cardei and D.-Z. Du, Improving Wireless Sensor Network Lifetime through Power Aware Organization, *ACM Wireless Networks*, Vol 11, No 3, May 2005.
- [4] M. Cardei, M. Thai, Y. Li, and W. Wu, Energy-Efficient Target Coverage in Wireless Sensor Networks, *IEEE INFOCOM*, Mar. 2005.
- [5] M. Cardei and J. Wu, Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks, *Computer Communications*, special issue on Sensor Networks, in press.
- [6] J. Carle and D. Simplot, Energy Efficient Area Monitoring by Sensor Networks, *IEEE Computer*, Vol 37, No 2, 2004.
- [7] Crossbow, <http://www.xbow.com>.
- [8] M. R. Garey, D. S. Johnson, *Computers and Intractability. A guide to the Theory of NP-Completeness*, Freeman, 1979.
- [9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, *Hawaii Intl. Conf. on System Sciences*, Jan. 2000.
- [10] Heterogeneous Networks with Intel XScale, <http://www.intel.com/research/exploratory/heterogeneous.htm>
- [11] T. Melodia, D. Pompili, V. C. Gungor, and I. Akyildiz, A Distributed Coordination Framework for Wireless Sensor and Actor Networks, *MobiHoc'05*, 2005.
- [12] RFC 3561, Ad hoc On-Demand Distance Vector (AODV) Routing, <http://www.faqs.org/rfcs/rfc3561.html>
- [13] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, Exploiting Heterogeneity in Sensor Networks, *IEEE INFOCOM*, 2005.
- [14] H. Zhang and J. C. Hou, Maintaining Sensing Coverage and Connectivity in Large Sensor Networks, *NSF Intl. Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Feb. 2004.