

# A Decentralized Lifetime Maximization Algorithm for Distributed Applications in Wireless Sensor Networks

Virginia Pilloni, Mauro Franceschelli, Luigi Atzori, Alessandro Giua  
Dept. of Electrical and Electronic Engineering, University of Cagliari  
09123, Cagliari, Italy  
{virginia.pilloni,mauro.franceschelli,l.atzori,giua}@diee.unica.it

**Abstract**—We consider the scenario of a Wireless Sensor Networks (WSN) where the nodes are equipped with a programmable middleware that allows for quickly deploying different applications running on top of it so as to follow the changing ambient needs. We then address the problem of finding the optimal deployment of the target applications in terms of network lifetime. We approach the problem considering every possible decomposition of an application’s sensing and computing operations into tasks to be assigned to each infrastructure component. The contribution of energy consumption due to the energy cost of each task is then considered into local cost functions in each node, allowing us to evaluate the viability of the deployment solution. The proposed algorithm is based on an iterative and asynchronous local optimization of the task allocations between neighboring nodes that increases the network lifetime. Simulation results show that our framework leads to considerable energy saving with respect to both sink-oriented and cluster-oriented deployment approaches, particularly for networks with high node densities and non-uniform energy consumption or initial battery charge.

**Index Terms**—Wireless Sensor Networks, network lifetime.

This work has been partially supported by the European Community’s Seventh Framework Programme under project HYCON2 (Grant Agreement n. FP7-ICT-2009-5/N.257462.)

## I. INTRODUCTION

Thanks to the increasing processing and transmission power of current nodes in Wireless Sensor Networks (WSN), these have become complex systems that are capable of making decisions and acting upon the information gathered about the monitored environment. Indeed, reduction in the cost of the devices has increased nodes capacity, thus they can perform some processing before sending the data to a sink. Still, one of the main open research challenges is the maximization of the network lifetime. Devices in a WSN are typically battery powered, battery that sometimes could be difficult to replace, such as in the case of subterranean or underwater nodes.

These considerations contribute to the vision of an horizontal ambient intelligent infrastructure wherein the sensing, computing and communicating infrastructure is set with a programmable middleware that allows for quickly deploying different applications running on top of it so as to follow the changing ambient needs. Based on this scenario, we focus on the need for a logic that decides on the optimal deployment solution of a target distributed application. Herein, optimality

is expressed in terms of network lifetime and a deploying solution is such that defines which part (single task) of the application has to be executed by each network component. In [?], we have proposed a centralized solution that is based on the assumption that a central node is aware of the status of each node, selects the optimal application deployment and send the corresponding setting to all the involved network components. With the intention to reduce the problem complexity, to reduce the overhead related to the communication of the central node with the rest of the network (for node settings), and to improve the capability of the network to adapt to topology and energy changes, in this paper we propose a distributed solution. The proposed algorithm is based on an iterative and asynchronous local optimization of the task allocations between neighboring nodes. The resulting scheme is based on *gossip*, which consists in a communication paradigm in which at each instant of time each node in the network has some positive probability to interact with one of its neighbors. Simulation results show that our framework results in considerable energy saving with respect to both sink-oriented and cluster-oriented deployment approaches, particularly for networks with high node densities and non-uniform energy consumption or initial battery charge

This paper is organized as follows. The second section provides the preliminaries; the third section introduces the problem and adopted approach; the fourth section describes the decentralized solution aimed at maximizing the network lifetime; the following section presents some simulation results; and conclusions are drawn in the last section.

## II. PRELIMINARIES

### A. Past Studies

Reduction of energy consumption has always been a key challenge for Wireless Sensor Networks. There are a great number of works which have focused on the maximization of the network lifetime, each one taking into account a different approach to achieve it: some are focused on efficient routing techniques [?], [?]; others are aimed at minimizing transmission energy consumption by sending data over may short hops rather than fewer long hops [?]; others propose the use of relay nodes so as to balance network energy consumption among nodes [?], [?].

However, none of the studies mentioned above considers the possibility of processing data in the nodes of the path to the destination. Because most of the energy spent in a Wireless Sensor Network depends on the amount of data that is transmitted over the network, reducing the amount of data may result in a reduction of the transmission energy consumption. This principle has been only partially adopted by LEACH [?], where sensors serve as Cluster Heads aggregating the data and, indeed, decreasing the number of bytes sent over the network. Energy consumption balancing is guaranteed by a random rotation of the role of Cluster Head.

Given the computational capacity of modern sensors, a step forward could be taken not just by aggregating data, but by processing them before they arrive at their destination whenever possible and on the basis of the network topology and power resource detection. In [?] an overlaying framework that determines the distribution of tasks among the nodes in a WSN by means of a centralized optimization algorithm aimed at maximizing the network lifetime was presented. However, one major drawback of this algorithm is its inability to react quickly to network changes.

In this paper we propose a framework of decentralized lifetime optimization for WSNs, adopting a solution in which the communication scheme is based on *gossip* [?], [?], [?]. Gossip algorithms are decentralized and asynchronous, they consist in a communication scheme in which at each instant of time each node in the network has some positive probability to interact with one of its neighbors. By the iterative interaction between nodes, several examples of emerging behaviors have been developed such as load balancing [?], distributed averaging [?], distributed convex optimization over networks [?], failure detection [?] and many more. Thus, communication schemes based on *gossip* that mimic the act of gossiping in a crowd of people, are easy to implement, do not require network routing or multi-hop communications, are inherently asynchronous and decentralized in nature.

### B. Energy Consumption

As mentioned in [?], energy consumption in WSNs is determined by three main components: sensing, processing and transmission.

The sensing energy consumption  $e^{sens}$  is determined by the specific characteristics of the sensor.

The processing energy consumption  $e^{proc}$  is proportional to the complexity of the *task* – that is, the number of instructions needed to complete it – and to the ingress data  $data^{in}$  – the higher the number of samples involved in the processing, the higher the energy consumption. Calling  $M^{task}$  the number of instructions for the *task*,  $smp^{in}$  the number of samples to be processed and  $e^{instr}$  the average energy consumption per executed instruction (determined on the basis of the device datasheet), then

$$e^{proc}(task, data^{in}) = M^{task} \times smp^{in} \times e^{instr} \quad (1)$$

Finally, the energy consumption  $e^{tx}$  required to transmit a packet of  $k$  bits from a node  $A$  to a node  $B$  with a constant

rate  $R$  is:

$$e^{tx}(k, P_{T0A}, \eta_A, P_{R0B}, \varphi_{AB}, \delta_{AB}) = \frac{k}{R} \left( P_{R0B} + P_{T0A} + \frac{\varphi_{AB} \times \delta_{AB}^\alpha}{\eta_A} \right) \quad (2)$$

where:  $P_{T0}$  and  $P_{R0}$  are the components of power consumption of the transmitting and receiving circuitry;  $\eta$  is the drain efficiency of the Power Amplifier;  $\varphi$  is a coefficient proportional to the reception power and the characteristic parameters of the antennas;  $\delta$  is the distance between transmitter and receiver;  $\alpha$  denotes the path loss exponent.

### III. PROBLEM FORMULATION

The model we consider is similar to the one described in major details in [?]. The set of nodes in the WSN is defined in the set  $X = \{1, \dots, i, \dots, N\}$ , where  $i$  can be a sensing node, a router or an actuator (or node with a combination of these roles). The node  $N$  refers to the sink (we assume to have only one sink in the network). In our problem setting, let the network be described by a directed acyclic graph  $\mathcal{G} = \{X, E\}$ , where  $E \subseteq \{X \times X\}$  is the set of edges, each representing a point-to-point communication channel between the nodes; edge  $(i, j)$  has its tail in node  $i$  and its head in node  $j$ , with its orientation representing the direction of the information flow. Let  $\mathcal{N}_i = \{j \in X : (i, j) \text{ or } (j, i) \in E\}$  be the neighborhood of node  $i$ , namely the nodes that share a communication channel with node  $i$ . Let  $\mathcal{N}_{out,i} = \{j \in X : (i, j) \in E\}$  be the set of nodes that receive information from node  $i$  and  $\mathcal{N}_{in,i} = \{j \in X : (j, i) \in E\}$  the nodes that send information to node  $i$ . We complete the network description with

- the distance matrix  $\mathbf{\Delta} = (\delta_{ij})$ , which contains the pairwise distances (in meters) between adjacent nodes. If nodes  $i$  and  $j$  are not adjacent, then  $\delta_{ij} = \infty$ ;
- the matrix  $\mathbf{\Phi} = (\varphi_{ij})$ , with the parameters  $\varphi_{ij}$  introduced in Section II-B, calculated for each couple of adjacent nodes  $i$  and  $j$ . If nodes  $i$  and  $j$  are not adjacent, then  $\varphi_{ij} = \infty$ ;
- the set of characteristic parameters  $V_i = \{P_{R0i}, P_{T0i}, \eta_i\}$ , which are useful to compute the transmission energy consumption as defined in (2) in Section II-B.

We assume that a given operation  $O$ , which can be decomposed into a sequence of tasks, has to be deployed in the network. This could represent diverse operations, including: computing the average temperature in certain geographical areas, measuring the light intensity in a room, videosurveillance of a specific geographical area, or a combination of these. We then define the sequence  $C^s = \{c_1^s, \dots, c_W^s\}$  of the sensing tasks which must be executed by the network to perform the operation  $O$ , where  $W$  is the total number of sensing tasks required. We also define the sequence  $C^p = \{c_1^p, \dots, c_L^p\}$  of the processing tasks which must be executed by the network to perform the operation  $O$ , where  $L$  is the total number of processing tasks required. If, for instance,  $O$  is a measurement of temperature from a geographical areas that has to be spatially averaged, the set  $C^s$  would be made of a number of elements corresponding to the number of geographical areas

where the temperature has to be measured.  $C^p$  would certainly be “measurement of temperature” and “spatial averaging”. Tasks are listed in  $C^p$  in priority order: if a node has to execute both  $c_1^p$  and  $c_2^p$ , the former must be executed before the latter.

As to the sensing we assign a binary state  $M_i \in \{0, 1\}^W$  to each node, which codes the sensing operations executed by node  $i$ . The status  $m_{iw}$  of the node is equal to 1 if the node performs the relating sensing task.

As to the processing tasks, we assign a binary state vector  $s_i \in \{0, 1\}^L$  to each node that represents the processing tasks currently assigned to the node. To each configuration of the node setting corresponds different energy consumption, as it will better explained in the following sections. We assume that different nodes consume different amounts of energy for the same processing to include heterogeneity of devices in the modeling. To each node  $i$  is associated a binary vector  $d_i \in \{0, 1\}^L$  which represents the kinds of processing that node  $i$  is allowed to execute. In particular, the following holds  $\forall i \in X$ , and  $\forall l \in \{1, \dots, L\}$ ,  $d_{il} \geq s_{il}$ . To simplify the notation we denote the matrix that collects all the statuses of each node as  $S = [s_1, s_2, \dots, s_N]^T$  and the matrix that represents all the constraints on the execution of processing as  $D = [d_1, d_2, \dots, d_N]^T$ .

We consider a scenario where the sensing operations are already assigned to the network members (i.e., binary vectors  $M_i$  are given, for  $i = 1, \dots, N$ ). Differently, the processing tasks in  $O$  can be executed according to different solutions: gathered data can be immediately sent to a sink, or it can be processed before being transmitted. In the case of the latter, the number of bits to be sent would be smaller, reducing the transmission energy consumption; however, processing energy consumption could be higher in this second case. Quantifying the energy consumption in both cases, it could be possible to establish which one determines a reduction of battery consumption in the sensors, incrementing the network lifetime. The addressed problem is then defined as the processing status matrix  $S$  that minimizes the impact of the operation  $O$  on the network, maximizing the network lifetime. In the following, we elaborate the considered scenario by defining further constraints and providing a distributed solution.

#### IV. DEPLOYMENT OF DISTRIBUTED APPLICATIONS

In the following, we present the proposed solution towards a distributed application deployment in WSN. The following Subsections present: the constraints on the traffic generated by the distributed applications; the cost functions built on the basis of the energy consumption formulas; the network lifetime maximization algorithm.

##### A. Constraints on the Traffic Flows

In our scenario we assume that the sources of traffic in the network (the sensors) generate samples of  $k$  bits at a certain frequency  $f$ . The processing in the network is performed on this type of traffic flow coming from different nodes. The generic node  $i$  receives the traffic  $T_i^{in}$  over which it performs the task corresponding to its assigned status  $s_i$ . The effect of

this task is the generation of the output traffic  $T_i^{out}$ , which is computed by function  $p$  as follows

$$T_i^{out} = p(T_i^{in}, s_i) \quad (3)$$

The output traffic is then sent to the next node. The data generated by  $p$  in node  $i$  is modeled by the  $H$ -dimensional vector  $T_i^{out} = (t_{i1}^{out}, \dots, t_{ih}^{out}, \dots, t_{iH}^{out})$ , where each element  $t_{ih}^{out} = \{k_{ih}^{out}, f_{ih}^{out}\}$  corresponds to a traffic flow where each sample of  $k_{ih}^{out}$  bits is transmitted at the frequency  $f_{ih}^{out}$ . The number of bits  $k_{ih}^{out}$  for each output flow  $t_{ih}^{out}$  resulting from  $p(x, y)$  is either constant or linear with the number of input flows, but it cannot be non linear. Further details are available in [?]. Data  $T_i^{out}$  are sent to the following node  $j$  according to the directed acyclic graph  $\mathcal{G}$ .

Node  $j$  receives data from all adjacent nodes that reach the sink through  $j$

$$T_j^{in} = \bigcup_{i \in N_{in,i}} T_i^{out} \times z_i, \quad \text{with } z_i = \begin{cases} 0 & s_i \equiv \text{“no actions”} \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

As defined by (3), data  $T_j^{in}$  received by node  $j$  are processed, according to the status of  $j$ . There are many processing tasks that can be performed in a WSN. For each one of these, an operator  $p(x, y)$  is defined. Note that for our objective, this operator is needed to figure out the traffic flows that will be traversing the network for each deployment scenario. We identified three common kinds of processing, which are spatial, temporal and single sample processing. We refer to [?] for further details about this distinction.

##### B. Cost Functions

The objective of the proposed algorithm is to evaluate the viability of each deployment solution on the basis of cost functions that are connected to energy consumption. We consider three cost functions: one for the sensing, one for the processing and one for the transmission. The sensing cost function for the node  $i$  is expressed as

$$E_i^{sens} = \sum_{w=1}^W f_i^{out} \times e_{iw}^{sens} \times m_{iw} \quad (5)$$

with  $e_{iw}^{sens}$  representing the sensing energy consumption for node  $i$  performing sensing task  $w$  if its status  $m_{iw}$  is equal to 1, as defined in Section II-B. Recall that  $f_i^{out}$  is the node output traffic frequency, which also represents the sensing frequency. We define processing cost function as follows

$$E_i^{proc} = \sum_{l=1}^L \sum_{h=1}^H f_{ih}^{out} \times e_{ih}^{proc}(c_{sil}, T_i^{in}) \times s_{il} \quad (6)$$

where  $e_{ih}^{proc}$  is the processing energy consumption defined in (1) which depends on the task  $c_{sil}$  that has to be executed, which in turn depends on the status  $s_{il}$  of the node, and the received data  $T_i^{in}$  described in (4). Because the processing cost depends on the number of processing per second performed by the same node  $i$ , it is proportional to the frequency  $f_{ih}^{out}$  of each of the  $H$  egress traffic flows, where  $H$  is the size of

$\mathbf{T}_i^{out}$  as described in Section IV-A. The number of samples to calculate  $e_i^{proc}$  is defined differently for each kind of processing  $p(x, y)$  detected in Section IV-A. Both sensing and processing are followed by a transmission. The related cost function is

$$E_i^{tx} = f_i \times e^{tx}(\mathbf{T}_i^{out}, V_i, \bigcup_{j \in \mathcal{N}_{in,i}} V_j, \bigcup_{j \in \mathcal{N}_{in,i}} \varphi_{ij}, \bigcup_{j \in \mathcal{N}_{in,i}} \delta_{ij}) \quad (7)$$

with  $f_i$  transmission frequency and  $e^{tx}$  transmission energy consumption defined by (??) depending on: the data to be transmitted  $\mathbf{T}_i^{out}$ ; the characteristic parameters  $V_i$  of the node  $i$ ; the characteristic parameters  $V_j$  of all the  $j$  nodes that will receive the data from  $i$  which, for a connected graph, is just one; the parameter  $\varphi_{ij}$  concerning nodes  $i$  and  $j$ ; the distance  $\delta_{ij}$  between  $i$  and  $j$ .

Given (5), (6) and (??), the overall cost function for any node  $i$  is

$$E_i = (E_i^{sens} + E_i^{proc} + E_i^{tx}) \quad (8)$$

### C. Algorithm description

We consider a network described by a directed acyclic graph  $\mathcal{G} = \{X, E\}$  (see III) in which the information flow has reached a stationary state. The processing state of the network is described by the processing matrix  $\mathbf{S}$ , where  $s_{il} = 1$  if node  $i$  processes task  $l$ , else  $s_{il} = 0$ . Our objective is that of modifying the processing state so as to maximize the *lifetime of the network*  $\tau(\mathbf{S})$ , intended as the time in which at least one node has exhausted its energy reserve from the battery: in fact when this condition is reached the network topology is disrupted.

If we denote the energy reserve of node  $i$  at time  $t$  as  $\gamma_i(t)$  then we can define

$$\tau(\mathbf{S}) = \inf\{t \mid (\exists i \in X) \gamma_i(t) = 0\}.$$

Assuming a stationary state, we observe that the optimal processing state is:

$$\mathbf{S}_{opt} = \arg \max_{\mathbf{S}} \tau(\mathbf{S}) = \arg \max_{\mathbf{S}} \min_{i \in X} \frac{\gamma_i}{E_i(\mathbf{S})}$$

where the last equation follows from the fact that in a stationary state the time required for a node to drain its battery is  $\gamma_i/E_i$  where  $E_i$  is the energy consumed (per unit of time).

Since the tasks a node can process are limited, we get a constrained optimization problem of the form:

$$\begin{aligned} & \max_{\mathbf{S}} \min_{i \in X} \frac{\gamma_i}{E_i(\mathbf{S})}, \\ & s.t. \\ & \forall i \in X \quad \mathbf{d}_i \geq \mathbf{s}_i \\ & \mathbf{S} \in \{0, 1\}^{N \times L} \end{aligned} \quad (9)$$

where  $\mathbf{d}_i$ , as previously defined, is the characteristics vector of the tasks that can be processed by node  $i$  and  $\mathbf{s}_i$  is the processing state of node  $i$ .

We wish to solve problem (??) in a decentralized way by iteratively and asynchronously solving an equivalent local

optimization problem that involves at each iteration only one node  $i$  and its in-neighbors  $\mathcal{N}_{in,i}$ .

We now propose the main result of this paper, namely the Decentralized Life Maximization for WSNs Algorithm (DLMA).

**Algorithm 1** (Decentralized Life Maximization for WSNs):

- 1) Each node  $i \in X$  is initialized with the residual energy of the battery  $\gamma_i$  and state  $\mathbf{s}_i = \{0\}^L$  (no processing assigned).
- 2) Let  $k = 0$  and  $t_0 = 0$ .
- 3) At time  $t_{k+1} > t_k$ , a node  $i$  not involved in a gossip, and thus, as defined in II-A, not communicating with any of its neighbours, is selected at random to interact with nodes in its in-neighborhood  $\mathcal{N}_{in,i}$ .
- 4) If any node in  $\mathcal{N}_{in,i}$  is already involved in a gossip, then  $k = k + 1$  and go to 3.
- 5) Node  $i$ , with a current processing state  $\bar{\mathbf{s}}_i$  (which is the state of node  $i$  at current time  $t_{k+1}$ ), obtains the state  $\bar{\mathbf{s}}_j$  for all  $j \in \mathcal{N}_{in,i}$ .
- 6) Solve the following local mixed integer linear programming problem in the unknown variables  $\alpha$  and  $\mathbf{s}_j$  for  $j \in \mathcal{N}_{in,i} \cup \{i\}$ :<sup>12</sup>

$$\begin{aligned} & \min \quad \alpha \\ & s.t. \\ & \frac{E_j(\mathbf{s}_k : k \in \mathcal{N}_{in,i} \cup \{i\})}{\gamma_j} < \alpha \quad \forall j \in \mathcal{N}_{in,i} \cup \{i\} \\ & \mathbf{d}_j \geq \mathbf{s}_j \quad \forall j \in \mathcal{N}_{in,i} \cup \{i\} \\ & \mathbf{T}_i^{out}(\mathbf{s}_{\mathcal{N}_{in,i} \cup \{i\}}) \leq \mathbf{T}_i^{out}(\bar{\mathbf{s}}_{\mathcal{N}_{in,i} \cup \{i\}}) \\ & \alpha \in \mathbb{R}^+ \\ & \mathbf{s}_j \in \{0, 1\}^L \quad \forall j \in \mathcal{N}_{in,i} \cup \{i\} \end{aligned} \quad (10)$$

and let its optimal solution be  $(\alpha_{opt}, \mathbf{s}_{\mathcal{N}_{in,i} \cup \{i\}, opt})$ .

- 7) If  $\mathbf{s}_{\mathcal{N}_{in,i} \cup \{i\}, opt}$  exists then set the new processing status to  $\mathbf{s}_j = \mathbf{s}_{j, opt}$ , for all  $j \in \mathcal{N}_{in,i} \cup \{i\}$ .
- 8) Let  $k = k + 1$  and go to 3. ■

We will formally prove in the following (see Theorem ??) that at each iteration  $k$  of the previous algorithm, the objective function of (??) either improves or does not change, although we cannot guarantee that the optimal solution of (??) is eventually found as  $k$  increases. However, Algorithm ?? offers several advantages with respect to centralized algorithms to compute a solution for problem (??). Both problems (??) and (??) are hard to solve. However, the computational complexity of the local optimization is function only of the number of

<sup>1</sup>With a slight abuse of notation we denote by  $\mathbf{s}_{\mathcal{N}_{in,i} \cup \{i\}}$  the set of vectors  $\mathbf{s}_i$  that represent the available processing tasks in the nodes in the set  $\mathcal{N}_{in,i} \cup \{i\}$ .

<sup>2</sup>With a slight abuse of notation we denote by  $E_j(\mathbf{s}_{\mathcal{N}_{in,i} \cup \{i\}})$  the energy consumed by node  $j$  as function of the local processing assignment, it is intended that the contribution of other nodes to this term can be computed simply by considering the term  $\mathbf{T}_j^{in}$  set of vectors  $\mathbf{s}_i$  that represent the available processing tasks in the nodes in the set  $\mathcal{N}_{in,i} \cup \{i\}$ .

nodes involved in the optimization, thus despite being a mixed integer linear programming problem, its complexity does not grow by increasing the number of nodes in the network and is small in absolute terms if the number of processing that may be allocated locally between the nodes is small. Since the allocation of processing is dynamic, the algorithm reacts to unexpected drops in battery charge by changing the status of the nodes involved.

We now characterize the behavior of the network while algorithm DLMA is being executed.

**Theorem 4.1:** Consider a WSN that executes Algorithm ???. Let the initial processing state  $\mathbf{S}$  be feasible for problem (??) in each neighborhood  $\mathcal{N}_i$  of  $\mathcal{G}$ . Let  $V(t) = \max_{i \in X} \frac{E_i(\mathbf{S})}{\gamma_i}$  be the inverse of minimum lifetime in the network set by the nodes with the smallest ratio between energy reserve and power consumption. Then, if the network executes Algorithm ???

$$\forall t \in \mathbb{R}^+ : V(t^+) \leq V(t).$$

*Proof:* During the algorithm execution, at each iteration  $k$  two situations may occur:

**Case 1:** The processing state matrix  $\mathbf{S}$  of the network does not change, then  $V(t^+) = V(t)$  by definition.

**Case 2:** The processing state matrix  $\mathbf{S}$  changes according to the solution of the local optimization problem (??).

The solution of problem (??) minimizes locally the energy consumption for the node with the shortest lifetime between nodes  $i$  and  $j \in \mathcal{N}_i$ . Thus, if a feasible solution is found, and we update the processing state of the nodes involved in the optimization from  $\bar{\mathbf{s}}_{\mathcal{N}_{in,i} \cup \{i\}, opt}$  to  $\mathbf{s}_{\mathcal{N}_{in,i} \cup \{i\}, opt}$  then

$$\begin{aligned} \min_{j \in \mathcal{N}_{in,i} \cup \{i\}} \frac{\gamma_i}{E_i(\mathbf{s}_{\mathcal{N}_{in,i} \cup \{i\}, opt})} &\geq \\ \min_{j \in \mathcal{N}_{in,i} \cup \{i\}} \frac{\gamma_i}{E_i(\bar{\mathbf{s}}_{\mathcal{N}_{in,i} \cup \{i\}})} &. \end{aligned} \quad (11)$$

Now we need to show that the nodes not involved in the local optimization do not decrease their life-time as a result. In the proposed local optimization a processing state update is performed only if each single information flow passing from node  $i$  to the nodes in  $\mathcal{N}_{out,i}$  is not increased due to the constraint  $T_i^{out}(\mathbf{s}_i) \leq T_i^{out}(\bar{\mathbf{s}}_i)$ . Now the transmission cost for any node in  $\mathcal{N}_{out,i}$  can only decrease as  $T_i^{out}(\mathbf{s}_i)$  is decreased, as shown in eq. (??), and so does its processing cost, as shown in eq. (6). Furthermore also the nodes in the downstream path toward the sink receive a smaller information flow thus consuming less power. Finally, the nodes in the upstream path have their information flow left unchanged and such is their power consumption. Thus it follows that

$$\min_{j \in X} \frac{\gamma_i}{E_i(\mathbf{S}_{opt})} \geq \min_{j \in X} \frac{\gamma_i}{E_i(\mathbf{S})},$$

and  $V(t^+) \leq V(t)$  thus proving the statement.  $\square$

## V. PERFORMANCE ANALYSIS

### A. Test Cases and Simulations Setup

To evaluate the effectiveness of the algorithm on a realistic WSN, two test cases have been taken into account, according

to some of the most significant realistic scenarios considered in past works, such as in [?]: uniform energy consumption and uniform initial energy at each node (UC-UE); non uniform energy consumption and non uniform initial energy (NUC-NUE) at each node (the energy consumption of the nodes has been assigned randomly from 60% to 140% of the energy consumption in case UC-UE; the initial energy has been assigned randomly from 20% to 100% of the total battery charge).

The analysis has been conducted in a MatLab environment, considering an outdoor agricultural scenario. It has been supposed to monitor a rectangular-shaped environment, where the nodes have been deployed with densities of 0.2, 0.3 and 0.4 nodes/m<sup>2</sup>.

It has been assumed that the nodes deployment follows a uniform distribution. Each node is equipped with sensors gathering information of temperature, humidity, PH and light exposure. The data are then sent to the Coordinator.

We have focused our analysis on one operation: calculation of the mean values of gathered information over an hour, starting from the values gathered every 10 minutes.

We have assumed that each sensed value is represented as a double numerical value, which is 64 bits long. The nodes communicate using IEEE 802.15.4 radio interfaces on the 2.4 GHz frequency band. The packets maximum size is 137 bytes, with a payload of 0 to 125 bytes. To keep things simple, any possible overhead has not been taken into account.

The local optimization problem has been solved using the integer linear programming solver GLPK (GNU Linear Programming Kit).

### B. Analysis of Case Studies

The optimization algorithm has been applied to each of the cases mentioned in ???. The results have been compared with three other mechanisms:

- i) data processed only by the Coordinator (mechanism C);
- ii) data processed by every Cluster Head found in the path to the Coordinator (mechanism CH);
- iii) centralized optimization algorithm described in [?] (mechanism CO).

In the following we present a comparison between the results obtained using the DLMA algorithm and those obtained using with mechanisms C, CH and CO. Specifically, we focus on the percentage of the energy conservation gained when using the proposed algorithm with respect to the alternative methods. We refer to these results with: DLMA-C, DLMA-CH, and DLMA-CO. Tab.?? shows the results for the operation defined in Section ???. The results show an average improvement of 57.7% of the proposed strategy for comparisons DLMA-C and DLMA-CH, while, as it should be expected, we observed an average decrement of 15.6% for comparison DLMA-CO. As a matter of fact, this is the drawback of the DLMA with respect to the centralized optimization algorithm which finds an optimal solution.

As for the simulation results of the centralized optimization algorithm, the best results are obtained for heterogeneous

TABLE I  
PERCENTAGE VALUES OF ENERGY CONSERVATION USING DLMA, FOR  
COMPARISONS DLMA-C, DLMA-CH AND DLMA-CO

Node density [nodes/m <sup>2</sup> ]	UC-UE [%]		
	DLMA-C	DLMA-CH	DLMA-CO
0.2	65.9	33.7	-22.0
0.3	69.9	32.1	-19.4
0.4	72.2	36.5	-20.0
	NUC-NUE [%]		
	DLMA-C	DLMA-CH	DLMA-CO
0.2	72.2	49.1	-11.2
0.3	76.8	49.7	-10.3
0.4	78.9	55.5	-10.6

networks, which are the most common in real scenarios. This is because, unlike other mechanisms where the processing is performed on fixed nodes regardless the energy consumed by the single nodes, the nodes chosen by the algorithm to perform the processing are those weighting less on the network.

The tendency of an improving energy conservation when node density increases is due to two factors: in case NUC-NUE, when the number of nodes in an area increases, it is more likely that among neighboring nodes there are nodes where the processing cost is lower due to their higher battery level or energy consumption; the higher the number of nodes in the same area, the larger the clusters formed, the bigger the amount of data that can be processed before they arrive to the Coordinator, reducing the energy cost.

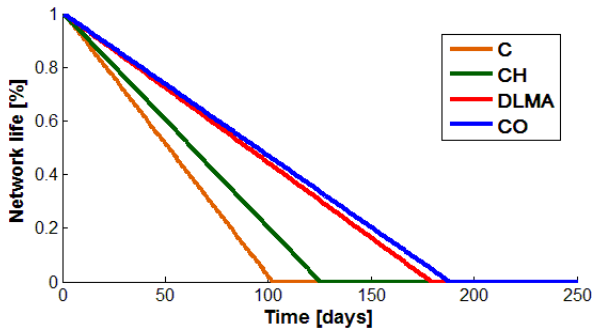


Fig. 1. Network life percentage decrement with respect to time expressed in days, for mechanisms C, CH, CO and DLMA, for a node density of 0.3 nodes/m<sup>2</sup> in case NUC-NUE

Fig.?? shows the percentage decrease of the network life expressed as the minimum residual battery capacity among all nodes in the network, with respect to time expressed in days, for the analysed mechanisms, for a node density of 0.3 nodes/m<sup>2</sup> in case NUC-NUE. Starting from the same initial battery charge, lifetime is shown to be much shorter if data are processed by fixed nodes, than in the solutions found by centralized optimization algorithm and DLMA. Furthermore, early death of nodes could be avoided programming nodes to run DLMA again when they reach a threshold battery level, so

that they can reduce their burden and extend their lifetime. Of course, running centralized optimization algorithm again when a node's battery level is critical should be much more difficult due to its higher complexity and to its centralized nature.

## VI. CONCLUSIONS

In this paper we have studied the deployment of distributed applications in WSNs and proposed a framework of decentralized lifetime maximization for WSNs which minimizes the impact of the applications on the network lifetime. The resulting DLMA algorithm based on gossip has been described and implemented to perform simulation in realistic scenarios. The results have been compared with alternative solutions, showing an improvement with respect to fixed nodes mechanisms such as data processed only by the Coordinator or by Cluster Heads. DLMA algorithm is outperformed by the global optimization algorithm, which however presents a higher computational complexity and is unable to react quickly to network changes.

## REFERENCES

- [1] V. Pilloni and L. Atzori, "Deployment of distributed applications in wireless sensor networks," *Sensors*, vol. 11, no. 8, pp. 7395–7419, 2011.
- [2] S. Singh, M. Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. ACM MobiCom*, 1998.
- [3] F. Liu, C. Tsui, and Y. J. Zhang, "Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 7, pp. 2258–2267, July 2010.
- [4] B. Calhoun, D. Daly, N. Verma, D. Finchelstein, D. Wentzloff, A. Wang, S. Cho, and A. Chandrakasan, "Design considerations for ultra-low energy wireless microsensor nodes," *IEEE Transactions Computers*, vol. 54, no. 6, pp. 727–740, June 2005.
- [5] S. C. Ergen and P. Varaiya, "Optimal placement of relay nodes for energy efficiency in sensor networks," in *Proc. IEEE ICC*, 2006.
- [6] J. Tang, B. Hao, and A. Sen, "Relay node placement in large scale wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 490–501, Feb. 2006.
- [7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, October 2002.
- [8] D. Shah, "Gossip algorithms," *Foundations and Trends® in Networking*, vol. 3, no. 1, pp. 1–125, 2009.
- [9] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [10] M. Franceschelli and A. Gasparri, "On agreement problems with gossip algorithms in absence of common reference frames," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4481–4486.
- [11] M. Franceschelli, A. Giua, and C. Seatzu, "Load balancing over heterogeneous networks with gossip-based algorithms," in *2009 American Control Conference*, St. Louis, Missouri, USA, Jun. 2009.
- [12] J. Lu, C. Tang, P. Regier, and T. Bow, "A gossip algorithm for convex consensus optimization over networks," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 301–308.
- [13] R. Van Renesse, Y. Minsky, and M. Hayden, "A gossip-style failure detection service," in *Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer-Verlag, 2009, pp. 55–70.
- [14] Z. Cheng, M. Perillo, and W. Heinzelman, "General network lifetime and cost models for evaluating sensor network deployment strategies," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 484–497, April 2008.