

---

# Energy-Efficient Task Allocation for Distributed Applications in Wireless Sensor Networks

Virginia Pilloni  
DIEE, University of Cagliari  
09123, Cagliari, Italy  
Email: virginia.pilloni@diee.unica.it

Luigi Atzori  
DIEE, University of Cagliari  
09123, Cagliari, Italy  
Email: l.atzori@diee.unica.it

**Abstract**—The increase in the computation and sensing capabilities as well as in battery duration of commercial available Wireless Sensors Network (WSN) nodes are making the paradigm of an horizontal ambient intelligence infrastructure feasible. Accordingly, 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. In this scenario, we face the problem of setting up the desired application in complex scenarios with hundreds of nodes, which consists in identifying which actions should be performed by each of the nodes so as to satisfy the ambient needs while minimizing the application impact on the infrastructure battery lifetime. Accordingly, we approach the problem by considering every possible decomposition of the application’s sensing and computing operations into tasks to be assigned to the each infrastructure component. The contribution of energy consumption due to the performance of each task is then considered to compute a cost function, allowing us to evaluate the viability of each deployment solution. Simulation results show that our framework results in considerable energy conservation with respect to sink-oriented or cluster-oriented deployment approaches, particularly for networks with high node densities, non-uniform energy consumption and initial energy, and complex actions. **Index Terms**—Wireless Sensor Networks, network lifetime.

## I. INTRODUCTION

Over the past decade, Wireless Sensor Networks (WSN) have consistently evolved into more complex distributed monitoring and control systems. In the beginning, the intelligence of the network was concentrated on the Coordinator, while all other nodes had basic processing and memory capacity.

WSNs are now becoming more and more complex: according to the information gathered by the sensors, the network is capable of making decisions and acting upon them. Reduction in the cost of the devices has increased nodes capacity, thus they can perform simple 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. In this case, we focus on the need for a logic that, starting from the desired application, can be set up in complex scenarios, evaluate the possible deployment solutions and decide which task should be performed by each of the nodes so as to satisfy the ambient needs while minimizing the application impact on the network lifetime. Accordingly, we approach the problem by assigning a cost to the execution of each possible task by every single node, which adds up to the total cost function. Solving the cost minimization problem, the assignment of tasks which weights less on the network will be found, and the network lifetime will be maximized.

This paper is organized as follows. The second and third sections introduce the problem and how we have approached it. The following section defines the algorithm used by the proposed framework to maximize the network lifetime. In the last section, some simulation results on the effectiveness of the framework are presented.

## II. BACKGROUND

### A. Past Studies

Due to their scarce resources, minimization of energy consumption has 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.

Routing is probably the most immediate issue addressed to accomplish this goal: a convenient choice of the paths to route data, may result in a significant energy conservation. In [1], an energy-efficient metric for finding routes was proposed. Other routing techniques are shown in many other studies, such as [2].

Some studies build on the assumption that transmission energy consumption is related to the square of the distance between two communicating nodes. Therefore, it might be more energy efficient to send data over many short hops, rather than fewer long hops. This issue was handled in [3]. This approach intends to maximize the network lifetime by minimizing overall energy consumption. However, it does not resolve the problem of unbalanced energy consumption among the nodes, which may lead to an early death of some nodes.

In order for some nodes not to die much earlier than others, energy consumption in the network should be as balanced as

possible. Relay nodes might be used for this purpose, as shown in [4] and [5]. Energy load distribution can also be achieved conveniently deploying the network nodes, as in [6]. In [7], the nodes are spaced non-uniformly as a function of their distance. Taking into account that nodes near the sink feel the effects of higher traffic more than other nodes, spacing are adjusted in such a way that nodes with higher traffic have a shorter hop distance than nodes with less traffic.

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 [8], 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 this paper, a framework that determines which of the nodes should process the data in order to maximize the network lifetime is presented. To the best of the authors' knowledge, no similar frameworks for WSN have been proposed before. An example of an overlaying framework that handles an architecture for an integration of the Internet of Things (IoT) in enterprise services might be found in [9]. However, this is not conceived to minimize the energy consumption, unlike the framework hereinafter described.

### B. Energy Consumption

Energy consumption in WSNs is determined most of all by transmission and reception. As mentioned in [10]

$$\begin{cases} P_T(\delta) = P_{T0} + \frac{P_{Rx} \times A \times \delta^\alpha}{\eta} \\ P_R = P_{R0} \end{cases} \quad (1)$$

where:  $P_T$  and  $P_R$  are radio frequency power consumptions for transmitting and receiving;  $P_{T0}$  and  $P_{R0}$  are the components of power consumption of the transmitting and receiving circuitry;  $P_{Rx}$  is the power to be received;  $A$  is a parameter determined by the characteristics of the antennas;  $\delta$  is the distance between transmitter and receiver;  $\alpha$  denotes the path loss exponent;  $\eta$  is the drain efficiency of the Power Amplifier. In this model we considered a channel in which the path loss component is predominant, and thus secondary effects such as multipath and Doppler can be neglected. Of course, the model might be extended to account for other fading effects.

Considering  $\varepsilon = P_{Rx_{min}} \times A$ , where  $P_{Rx_{min}}$  is the minimum reception power for a reliable communication, the total power consumption for communicating between a transmitting node  $A$  and a receiving node  $B$  could be written as

$$P_{AB} = P_{R0B} + P_{T0A} + \frac{\varepsilon_{AB} \times \delta_{AB}^\alpha}{\eta}$$

Therefore, the energy consumption of the network to transmit a packet of  $k$  bits from  $A$  to  $B$  with a constant data rate  $R$  is

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

The model described does not take into account mechanisms such as sleeping schedule and route discovery, which may produce overhead. It could thus be necessary to also consider the energy consumption due to the overhead.

Besides transmission and reception, the other two main causes of energy consumption are due to the sensing activity and to the processing. 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 instruction executed, which can be determined on the basis of the device datasheet used

$$e^{proc}(task, data^{in}) = smp^{in} \times M^{task} \times e^{instr} \quad (3)$$

### III. PROBLEM FORMULATION

The goal of a WSN is to accomplish a given number of operations, mostly based on measurements performed on the relevant environment. Given the status of the network in terms of nodes capacity, topology and energy distribution, the problem addressed is to assign to each node the tasks that, combined together, contribute to the target network operations while minimizing the impact on the network lifetime.

In our modelling,  $X = \{x_1, \dots, x_i, \dots, x_N\}$  is the set of nodes in the WSN, where  $x_i$  can be a sensing node, a router or an actuator (or node with a combination of these roles). The node  $x_N$  refers to the sink (we assume to have only one sink in the network). The network can be described by:

- the  $N \times N$  adjacency matrix  $\mathbf{A} = (a_{ij})$ : an element  $a_{ij}$  of  $\mathbf{A}$  is equal to 1 when a link connects node  $i$  to node  $j$  and the sink is closer to  $j$  than to  $i$ ;
- the distance matrix  $\mathbf{\Delta} = (\delta_{ij})$ , which contains the pairwise distances (in meters) between adjacent nodes. If  $\delta_{ij} = \infty$ , nodes  $i$  and  $j$  are not adjacent;
- the matrix  $\mathbf{E} = (\varepsilon_{ij})$ , with the parameters  $\varepsilon_{ij}$  introduced in Section II-B, calculated for each couple of adjacent nodes  $i$  and  $j$ . If  $\varepsilon_{ij} = \infty$ , nodes  $i$  and  $j$  are not adjacent;
- 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;
- the set  $F = \{f_0, \dots, f_w, \dots, f_W\}$  of tasks, which encompasses all the tasks that can be performed by any node in the network for any kind of application. For instance, 0 might correspond to the “temperature sensing in the

area 1”, 1 to the “temperature sensing in the area 2”, 2 to the “spatial averaging on received data”, 3 to the “only transmission”, 4 to “no actions”. Each of the tasks in an  $F$  set entails a transmission of data, with the exception, of course, of “no actions”;

- the set  $D_i = \{d_{i1}, \dots, d_{im}, \dots, d_{il_i}\}$ , with  $D_i \subseteq F$ , where the elements of  $D_i$  are the tasks that the node  $x_i$  is able to perform.

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.

- the total cost value  $E^{tot}$ , which takes into account the energy consumption;
- the sequence  $C = \{c_1, \dots, c_l, \dots, c_L\}$  of the sub-operations which must be executed by the nodes to perform the operation  $O$ . If, for instance,  $O$  is a measurement of temperature from a geographical areas that has to be spatially averaged, some of the sub-operations in  $C$  will certainly be “measurement of temperature” and “spatial averaging”. Each sub-operation is equivalent to one of the  $f_w$  tasks defined above:  $C \subseteq F$ . The terms sub-operation and task are used interchangeably, but we refer to  $c_l$  as a sub-operation to emphasise its link with its respective operation  $O$ . Sub-operations are listed in  $C$  in priority order: if a node has to execute both  $c_1$  and  $c_2$ , the former must be executed before the latter;
- the set  $S = \{s_1, \dots, s_i, \dots, s_N\}$ , where  $s_i$  is the status of node  $i$  with respect to operation  $O$ . The value  $s_i$  defines which sub-operation  $c_l$  the node  $i$  is performing. Of course, the status of the node  $x_i$  has to be chosen among the set of tasks  $D_i$  that the node is able to perform. If the node is not involved in the operation  $O$ , its status must correspond to “no actions”. For this reason, “no actions” is necessarily included in the set  $F$ . It has to be always verified that:

- if a node  $x_j$  receives some data from a node  $x_i$ , which means that  $s_i$  cannot be “no actions”, node  $x_j$  must at least transmit the data

$$s_i \neq \text{“no actions”} \vee a_{ij} = 1 \Rightarrow s_j \neq \text{“no actions”} \quad (4)$$

- if a node  $x_j$  does not receive any data, which means that the status of all the nodes connected to it is “no actions”,  $s_j$  must also be set to “no actions”

$$s_i \equiv \text{“no actions”} \forall i : a_{ij} = 1 \Rightarrow s_j = \text{“no actions”} \quad (5)$$

Thanks to the the greater processing power and storage capacity of modern sensors, contrary to the past the same operation  $O$  can be performed in several different ways: 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 problem addressed is defined as the set of statuses  $S$  that minimizes the impact of the operation  $O$  on the network, minimizing the cost function  $E^{tot}$ . Afterwards the framework elaborates and assigns among the nodes the most appropriate tasks to be performed. Hence it is evident that the cost function  $E^{tot}$  will vary depending on the status of each node, that is, how the operation  $O$  is performed. In the following, we elaborate the considered scenario by defining further constraints that solve the problem.

#### 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 concept of virtual nodes, which are duplicates of real nodes that are introduced to deal with nodes that perform more than one sub-operation in a deployment solution; the cost functions built on the basis of the energy consumption formulas; the network lifetime maximization procedure; a summary of the proposed framework.

##### 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  $x_i$  receives the traffic  $\mathbf{T}_i^{in}$  over which it performs the sub-operation corresponding to its assigned status  $s_i$ . The effect of this sub-operation is the generation of the output traffic  $\mathbf{T}_i^{out}$ , which is computed by function  $p$  as follows

$$\mathbf{T}_i^{out} = p(\mathbf{T}_i^{in}, s_i) \quad (6)$$

The output traffic is then sent to the next node.

The data generated by  $p$  in the node  $x_i$  is modelled by the  $H$ -dimensional vector  $\mathbf{T}_i^{out} = (t_{i1}^{out}, \dots, t_{ih}^{out}, \dots, t_{iH}^{out})$ , 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  $f_{ih}^{out}$ . Each sample described by  $t_{ih}^{out}$  results from a spatial processing or a sensing. The data  $\mathbf{T}_i^{out}$  is then sent to the following node  $x_j$  according to adjacency matrix  $\mathbf{A}$ .

The node  $x_j$  receives data from all adjacent nodes that reach the sink through  $x_j$ , with the exception of the nodes with a “no actions” status

$$\mathbf{T}_j^{in} = \bigcup_{i=1}^N \mathbf{T}_i^{out} \times a_{ij} \times z_i, \quad \text{with } z_i = \begin{cases} 0 & s_i \equiv \text{“no actions”} \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

As defined by (6), the data  $\mathbf{T}_j^{in}$  received by the node  $x_j$  is processed, according to the status of  $x_j$ :

- if  $s_j$  is a sensing status,  $p$  does not take any  $\mathbf{T}_j^{in}$  as input and the output is defined by the specific sensing operation;

- if  $s_j$  is an “only transmission” status, the output of  $p$  is exactly equal to  $\mathbf{T}_j^{in}$ ;
- if  $s_j$  is a “no action” status,  $p$  returns a  $\mathbf{T}_j^{out}$  with all fields set to 0 as output. This case is included for completeness, but it is not supposed to happen because of the (4);
- if  $s_j$  is a processing status,  $\mathbf{T}_j^{out}$  can be the most diverse depending on the specific processing objectives, which are coded in  $s_j$  and that control the specific function  $p$ . In the following we analyze certain cases.

There are many processing tasks that can be performed in a WSN and that are coded in  $F$  as described in Section III. 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. The three most common kinds of processing that we have identified, which are spatial, temporal and single sample processing, will be described below.

In a spatial processing, the samples coming from different paths are processed together. The resulting  $\mathbf{T}_j^{out}$  is made of only one element  $t_{j1}^{out} = \{k_{j1}^{out}, f_{j1}^{out}\}$ , where the number of bits per sample  $k_{j1}^{out}$  is set according to the processing output, while the frequency  $f_{j1}^{out}$  is equal to those of each input flows.

Differently, the temporal processing is performed on every traffic flow in  $\mathbf{T}_j^{in}$ . The resulting  $\mathbf{T}_j^{out}$  contains the same number of traffic flows in  $\mathbf{T}_j^{in}$ , where every element  $t_{jh}^{out}$  is characterized by the same number of bits per sample  $k_{jh}^{out}$  and the same frequency  $f_{jh}^{out}$  corresponding to the processing frequency associated to the node status  $s_j$ .

Other processing tasks can be performed on every single sample of each received traffic flow. This is the case, for instance, where one must evaluate whether the received values exceed a give threshold or not.  $\mathbf{T}_j^{out}$  contains the same number of traffic flows as in  $\mathbf{T}_j^{in}$  at the same frequency  $f_{jh}^{out}$ , but with different bits per sample  $k_{jh}^{out}$ .

### B. Virtual Nodes

It is possible that the same node has to perform more than one sub-operation  $c_l$ . For instance, if the operation  $O$  is a temporal and spatial average of the temperature values measured in different geographical areas, it may happen that one single node has to perform both the spatial average and the temporal average on the received data. Therefore, for every node that can perform more than one sub-operation, we introduce an appropriate number of virtual nodes, each of them can execute only one sub-operation at a time. For every operation  $O$ , the network must be redefined taking into account the virtual nodes. For space reasons, this topic cannot be described here further. Nevertheless, the introduction of the virtual nodes does not distort the algorithm in its whole.

### C. Cost Functions

The objective of the proposed algorithm is to evaluate the viability of each deployment solution on the basis of a cost function that is connected to energy consumption. Quite often in similar scenarios, past studies have proposed the evaluation of the network lifetime and have aimed at maximizing it. Since

the network lifetime is affected by applications which can be analysed one by one, there is no sense in computing the network lifetime. For this reason, we try to minimize a cost function associated to the operation under analysis.

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  $x_i$  is expressed as

$$E_i^{sens} = f_i^{out} \times \gamma_i \times e_i^{sens} \times y_i, \quad (8)$$

$$\text{with } y_i = \begin{cases} 1 & \text{if } s_i \equiv \text{sensing code} \\ 0 & \text{otherwise} \end{cases}$$

with  $e_i^{sens}$  representing the sensing energy consumption as defined in Section II-B. Recall that  $f_i^{out}$  is the node output traffic frequency, which also represents the sensing frequency. The parameter  $\gamma_i$  is a coefficient in inverse proportion to the residual energy of the node. When performing our experiments, we supposed that  $\gamma_i$  could change linearly from 1 (battery fully loaded) to 5 (battery level lower than 20%).

We define processing cost function as follows

$$E_i^{proc} = \sum_{h=1}^H f_{ih}^{out} \times \gamma_i \times e_{ih}^{proc}(c_{s_i}, \mathbf{T}_i^{in}) \times v_i \quad (9)$$

$$\text{with } v_i = \begin{cases} 1 & \text{if } s_i \equiv \text{processing code} \\ 0 & \text{otherwise} \end{cases}$$

where  $\gamma_i$  is the coefficient defined above, and  $e_{ih}^{proc}$  is the processing energy consumption defined in (3) which depends on the sub-operation  $c_{s_i}$  that has to be executed, which in turn depends on the status  $s_i$  of the node, and the received data  $\mathbf{T}_i^{in}$  described in (7). Because the processing cost depends on the number of processing per second performed by the same node  $x_i$ , 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. Therefore, unless the node is in a “no actions” status, it has to transmit the output data. Because, for (4), if a node receives data it cannot be in “no actions” status, every node involved in the operation  $O$  has to transmit data. The related cost function is

$$E_i^{tx} = f_i \times \gamma_i \times e^{tx}(\mathbf{T}_i^{out}, V_i, \bigcup_{j=1}^N V_j \times a_{ij}, \bigcup_{j=1}^N \varepsilon_{ij}, \bigcup_{j=1}^N \delta_{ij}) \quad (10)$$

with  $f_i$  transmission frequency,  $\gamma_i$  residual energy coefficient, and  $e^{tx}$  transmission energy consumption defined by (10) depending on: the data to be transmitted  $\mathbf{T}_i^{out}$ ; the characteristic parameters  $V_i$  of the node  $x_i$ ; the characteristic parameters  $V_j$  of all the  $x_j$  nodes that will receive the data from  $x_i$  which, for a connected graph, is just one; the parameter  $\varepsilon_{ij}$  concerning nodes  $x_i$  and  $x_j$ ; the distance  $\delta_{ij}$  between  $x_i$  and  $x_j$ .

Given (8), (9) and (10), the overall cost function for any operation  $O$  is

$$E^{tot} = \sum_{i=1}^N (E_i^{sens} + E_i^{proc} + E_i^{tx}) \quad (11)$$

#### D. Maximization of Network Lifetime

The goal is to find the set of statuses  $S$  of the nodes that minimize the network energy cost function. The optimization problem becomes

$$\text{minimize } E^{tot} = \sum_{i=1}^N (E_i^{sens} + E_i^{proc} + E_i^{tx}) \quad (12a)$$

$$\text{subject to } \sum_{i=1}^N q_{il} \leq Q_l^{max} \quad \text{with } q_{il} = \begin{cases} 1 & s_i \equiv c_l \\ 0 & \text{otherwise} \end{cases} \quad (12b)$$

$$\bigcup \{s_N\} = D_N \cap C \quad (12c)$$

The condition in (12b) is a constraint on the maximum  $Q_l^{max}$  number of nodes that must perform the sub-operation  $c_l$ . This could be necessary, for instance, when a given geographical area is monitored by a certain number of nodes, but the required information is not needed from all of them. In this case, the algorithm would assign  $c_l$  to the  $Q_l^{max}$  sensors which weight less on the network lifetime, among all of those which are able to perform the required sub-operation. When this constraint is not needed for  $c_l$ ,  $Q_l^{max}$  is set to  $N$ .

The condition (12c) shows that the set of statuses of the sink node  $x_N$  must correspond to all the sub-operations that the sink is able to perform, which means that, if there is any data still to be processed, the sink has to process them.

The problem defined in (12) is an Binary Integer Programming (BIP) problem: the unknown status of a node can be defined as a  $|C|$ -dimensional binary array, where  $C$  is the set of sub-operations. Since every node can only have one status, which means that it can perform only one sub-operation, only one element of this array can be equal to 1, and it corresponds to one of the sub-operations that the relating node  $x_i$  is able to perform, according to  $D_i$ . The elements of the array represent the weights to the contributions (8) and (9) of the node to  $E^{tot}$ . BIP problems are usually solved with efficient branch-and-bound algorithms of linear programming solvers such as CPLEX [11] or Xpress Optimization Suite [12]. Since BIP problems complexity scales exponentially with the problem size, finding the exact solution can be time consuming. Therefore, heuristic algorithms are used to obtain sub-optimal solutions which may be considered sufficient in most cases.

#### E. The Proposed Framework

In a WSN the Coordinator, which is responsible for initiating and maintaining the network, is the device on which the deployment algorithm is performed. The proposed algorithm needs to know the exact topology of the network, that is, how the nodes are connected to each other and what is the distance between any two of them, as well as the routing table. In order to compute the cost function (11), further information will be needed, such as the parameters to model the radio channel, the transmission, reception, sensing and processing

energy consumption of each node, the residual energy of each node, the working frequency and the data rate.

First, the algorithm defines the sets and matrices described in Section III; then, the network with the necessary virtual nodes is defined; the problem (12) is solved with a linear programming solver, in order to find the set  $S$ ; finally, a low level code describing which tasks each node has to perform will be developed and distributed to the appropriate nodes.

The solving algorithm has been implemented in Mosel language, and the solution has been found using Xpress Optimization Suite.

## V. PERFORMANCE ANALYSIS

### A. Test Cases and Simulations Setup

To evaluate the effectiveness of the algorithm on a realistic WSN, three test cases have been taken into account, according to some of the most significant realistic scenarios considered in past works, such as in [13]: uniform energy consumption and uniform initial energy at each node (UC-UE); non uniform energy consumption and uniform initial energy (NUC-UE) 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); uniform energy consumption and non uniform initial energy (UC-NUE) at each node (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, divided into areas of  $25 \text{ m}^2$ , where the nodes have been deployed with densities of 0.2, 0.3 and  $0.4 \text{ nodes/m}^2$ .

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 two operations: calculation of the mean values of gathered information over an hour, starting from the values gathered every 10 minutes from every area (OpA); aggregation of traffic coming from different areas of the network, carrying gathered information to the Coordinator, for later analysis by qualified staff (OpB).

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.

### B. Analysis of Case Studies

The optimization algorithm has been applied to each of the cases mentioned in V-A. The resulting cost value has been compared with:

- i) the cost value that is obtained if data are processed only by the Coordinator (comparison C);
- ii) the mean cost value for all the possible solutions that might be detected (comparison M). This is introduced to

TABLE I  
PERCENTAGE VALUES OF ENERGY CONSERVATION USING THE  
FRAMEWORK, FOR COMPARISONS C AND M

	Node density [nodes/m <sup>2</sup> ]	UC-UE [%]		NUC-UE [%]		UC-NUE [%]	
		C	M	C	M	C	M
OpA	0.2	19.5	11.7	50.0	42.2	47.5	33.6
	0.3	20.9	12.7	55.2	43.2	50.3	35.6
	0.4	25.6	16.5	58.0	46.2	56.7	43.7
OpB	0.2	64.7	46.4	70.4	52.4	70.4	52.6
	0.3	66.4	48.8	75.2	59.1	72.3	55.7
	0.4	70.9	55.4	77.9	63.8	76.1	61.7

make a comparison with solutions where the processing of the data is performed on fixed nodes, which is expected to bring results corresponding to the median solution.

These comparisons are expressed as a percentage of the energy conservation that would result using the proposed algorithm, with respect to the alternative ones.

Tab.I shows the results for the two operations.

The results show an average improvement of 49.6% of the proposed strategy with respect to the alternative ones. Limited benefits are observed in case UC-UE for both operations. This is because for networks with heterogeneous parameters, which are the most common in real scenarios, in cases NUC-UE and UC-NUE devices' energy consumption does not weight the same on the entire network. This means that, 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 cases NUC-UE and UC-NUE, when the number of nodes in an area increases, it is more likely that among neighbouring nodes there are nodes where the processing cost is lower; 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.

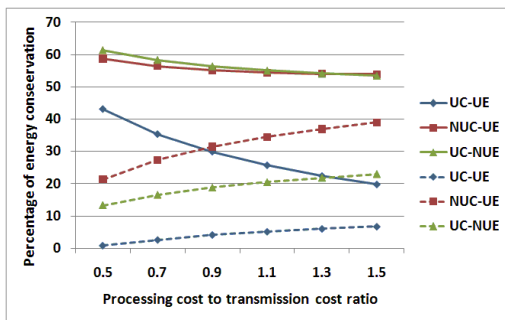


Fig. 1. Percentage of energy conservation with respect to the ratio between processing cost and transmission cost, for cases UC-UE, NUC-UE and UC-NUE, for a density of 0.3 nodes per m<sup>2</sup>. Solid lines show energy conservation with respect to data processed only by the Coordinator; dashed lines show energy conservation with respect to data processed by every Cluster Head

The results of the framework have been compared with another typical case: data processed by every Cluster Head found in the path to the destination. Fig.1 depicts the percentage of energy conservation while the ratio between the processing cost and the cost to transmit 137 bytes of data increases. Comparison has been made both in the case data are processed only by the Coordinator and data are processed by every Cluster Head. In the former, energy conservation decreases when processing cost to transmission cost ratio increases. In fact, when the processing cost increases with respect to the transmission cost, transmitting data becomes more convenient than processing them. On the other hand, in the latter energy conservation increases when processing cost to transmission cost ratio increases, consistently with the fact that, when the processing cost increases, it is more convenient to accurately choose the nodes where processing might be performed rather than processing data every time that it is possible to do so.

## VI. CONCLUSIONS

In this paper we have studied the deployment of distributed applications in WSNs and proposed a solution aimed at minimizing the impact of energy consumption on the network lifetime. The resulting algorithm has been implemented to perform simulations in different scenarios with heterogeneous parameters, and the results have been compared with alternative solutions. We may infer from the results that using the framework described would be particularly energy conserving when the application encompasses the processing of data coming from different nodes, the processing is pretty complex, and the energy consumption of nodes as well as battery energy is not uniform over the entire network.

## REFERENCES

- [1] S. Singh, M. Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. ACM MobiCom*, 1998.
- [2] 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.
- [3] 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.
- [4] S. C. Ergen and P. Varaiya, "Optimal placement of relay nodes for energy efficiency in sensor networks," in *Proc. IEEE ICC*, 2006.
- [5] 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.
- [6] Q. Gao, K. Blow, D. Holding, and Q. R. Iqbal, "Radio range adjustment for energy efficient wireless sensor networks," in *Proc. SpringSim07*, 2007.
- [7] I. Howitt and J. Wang, "Energy balanced chain in wireless sensor networks," in *Proc. IEEE WCNC*, 2004.
- [8] 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.
- [9] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. M. de Souza, and V. Trifa, "Soa-based integration of the internet of things in enterprise services," in *Proc. IEEE IWCS*, 2009.
- [10] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *Proc. IEEE SECON*, 2006.

- 
- [11] ILOG, “Cplex user’s manual,” 2008. [Online]. Available: [http://yalma.fime.uanl.mx/cplex11-manual/Content/Optimization/Documentation/CPLEX/\\_pubskel/XPlatform/data.html](http://yalma.fime.uanl.mx/cplex11-manual/Content/Optimization/Documentation/CPLEX/_pubskel/XPlatform/data.html)
- [12] FICO, “Xpress optimization suite.” [Online]. Available: <http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx>
- [13] 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.