sensors

Article

Deployment of Distributed Applications in Wireless Sensor Networks

Virginia Pilloni and Luigi Atzori [*]

Department of Electric and Electronic Engineering, University of Cagliari, Cagliari 09123, Italy;
E-Mail: virginia.pilloni@diee.unica.it

[*] Author to whom correspondence should be addressed; E-Mail: l.atzori@diee.unica.it.

Abstract: The increase in computation and sensing capabilities as well as in battery duration of commercially 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 of 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 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.

Keywords: Wireless Sensor Networks; energy consumption; network lifetime

Energy Consumption

P_T	Transmitting radio frequency power consumption
P_R	Receiving radio frequency power consumption
P_{T0}	Transmitting circuit power consumption
P_{R0}	Receiving circuit power consumption
P_A	Power Amplifier power consumption
P_{Tx}	Output power at the antenna
P_{Rx}	Input power at the antenna
δ	Distance between transmitting and receiving nodes
η	Drain efficiency of the Power Amplifier
A	Parameter determined by the characteristics of the antennas such as gain and efficiency
α	Path loss exponent
ε	Parameter proportional to P_{Rx} and A
e^{tx}	Transmission energy consumption
R	Data rate
e_T	Energy consumed to send one bit
e_R	Energy consumed to receive one bit
e^{sens}	Sensing energy consumption
e^{proc}	Processing energy consumption
e^{instr}	Average energy consumption per instruction executed
M^{task}	Number of instruction to process $task$
smp^{in}	Number of samples to be processed

Problem Formulation

$X = \{x_1, ..., x_i, ..., x_N\}$	Set of N nodes in the WSN
$\mathbf{A} = (a_{ij}) \in \mathbb{R}^{N \times N}$	Adjacency matrix
$\mathbf{\Delta} = (\delta_{ij}) \in \mathbb{R}^{N \times N}$	Distances matrix
$\mathbf{E} = (\varepsilon_{ij}) \in \mathbb{R}^{N \times N}$	Matrix of the parameters ε
$V_i = \{P_{R0i}, P_{T0i}, \eta_i\}$	Set of characteristic parameters of the node x_i
$F = \{f_0, ..., f_w, ..., f_W\}$	Set of W tasks that can be performed by any node in the network
$D_i = \{d_{i1}, ..., d_{im}, ..., d_{il_i}\}$	Set of l_i tasks that the node x_i is able to perform
O	Operation performed by the WSN
E^{tot}	Total cost value associated to O
$C = \{c_1, ..., c_l, ..., c_L\}$	Set of L sub-operations which must be executed by the nodes to perform operation O
$S = \{s_1, ..., s_i, ..., s_N\}$	Set of N statuses, where s_i corresponds to the status of the node x_i

Traffic Flows

$\mathbf{T}_i^{out} = (t_{ih}^{out}) \in \mathbb{R}^H$	Vector of the output traffic flows generated by node x_i
$t_{ih}^{out} = \{k_{ih}^{out}, f_{ih}^{out}\}$	Traffic flow, corresponding to the h-th element of $\mathbf{T}_i^{out}$, where each sample of k_{ih}^{out} bits is transmitted at the frequency f_{ih}^{out}
$\mathbf{T}_i^{in}$	Vector of the input traffic flows, corresponding to the union of the traffic flows received by node x_i
$p(\mathbf{T}_i^{in}, s_i)$	Function which generates the output traffic $\mathbf{T}_i^{out}$, based on the input traffic and the status of the node x_i

Virtual Nodes

G_i	Set of sub-operations that a single node x_i can perform consecutively for the implementation of the operation O
L	Set of sub-operations that the nodes cannot execute with the others
$X^{vir} = \{x_1^{vir}, ..., x_v^{vir}, ..., x_{N^{vir}}^{vir}\}$	Set of N^{vir} nodes of the WSN with virtual nodes
$\mathbf{A}^{vir} \in \mathbb{R}^{N^{vir} \times N^{vir}}$	Adjacency matrix for the WSN with virtual nodes
$\mathbf{\Delta}^{vir} \in \mathbb{R}^{N^{vir} \times N^{vir}}$	Distances matrix for the WSN with virtual nodes
$\mathbf{E}^{vir} \in \mathbb{R}^{N^{vir} \times N^{vir}}$	Matrix of the parameters ε for the WSN with virtual nodes
V_v^{vir}	Set of characteristic parameters of the node x_v^{vir} of the WSN with virtual nodes
D_v^{vir}	Set of tasks that the node x_v^{vir} of the WSN with virtual nodes is able to perform

Cost Functions

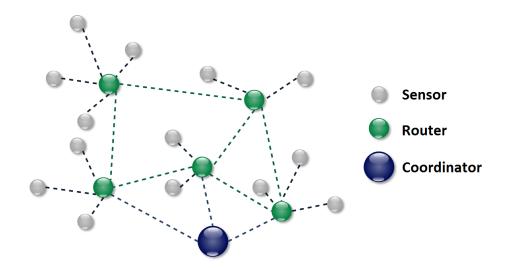|	
E_i^{sens}	Sensing cost function for node x_i
γ_i	Coefficient in inverse proportion with the residual energy of the node x_i
E_i^{proc}	Processing cost function for the node x_i
E_i^{tx}	Transmission cost function for the node x_i
Q_l^{min}	Minimum number of nodes that have to perform the sub-operation c_l
Q_l^{max}	Maximum number of nodes that have to perform the sub-operation c_l

1. Introduction

Over the past decade, Wireless Sensor Networks (WSN) have consistently evolved into more complex distributed monitoring and control systems. In the beginning, the only goal of a WSN was to monitor a given environment. As depicted in Figure 1, sensors gathered the required information mostly according to a fixed temporal schedule and sent it to the routers, whose primary functions were to both perform the sensing and conveniently transmit data received from sensors. Finally, the Coordinator would collect the data transmitted by the router. The Coordinator has much higher processing and memory capacity

than the other nodes in the network. It initializes and manages the WSN: it is responsible for routing and storing the routing tables as well as information about the network and security keys. The Coordinator usually interfaces with a server or a computer. In this way, data from sensors could be processed and stored. The Coordinator possessed all the intelligence of the network. Given the cost of more advanced devices, all other nodes only had basic processing and memory capacity [1].

A WSN is characterized by the presence of one or more sinks. A sink is a node which gathers and controls data collected by different sensor nodes. In every network there is at least one sink present, corresponding to the Coordinator.

Figure 1. Structure of a WSN.

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. Indeed, they are expected to be one of the pillars of the Internet of Things (IoT) paradigm [2], which foster the introduction of key applications, including but not limited to domotics, assisted living, e-health, enhanced learning automation and industrial manufacturing logistics, business/process management, and intelligent transportation of people and goods. Reduction in the cost of the devices has increased the nodes' capacity, thus they can perform some processing before sending the data to a sink. They can aggregate data coming from different sensors, perform temporal and spatial averaging as well as data filtering so as to reduce the burden of transmitting large amounts of data to the Coordinator and increase the network lifetime. Indeed, devices in a WSN are typically powered by batteries that can be difficult to replace, such as in the case of subterranean or underwater nodes.

These considerations contribute to the vision of an horizontal ambient intelligence 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, for example: monitoring a given geographical area and alerting when something is happening therein; activating the heating system when the ambient is getting cold; tracking the processing chain in industrial plants to prevent hazardous scenarios. In this case, we focus on the need for a logic that, starting from the desired application, can be set up in complex scenarios with hundreds of nodes, evaluate the possible deployment solutions and decide which action should be performed

by each of the nodes so as to satisfy the ambient needs while minimizing the application's impact on the infrastructure's 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 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.

This paper is organized as follows. The second Section describes how other authors have dealt with the problem of maximizing of the network lifetime, and the model of energy consumption considered in this work. The third Section introduces the problem and how we have approached it. The following Section defines the algorithm used by the framework to maximize the network lifetime. In the fifth Section, some simulation results on the effectiveness of the framework are presented. The last Section draws final conclusions. Finally, in the Appendix we provide the list of symbols used to make it easier to follow the description of the proposed algorithm.

2. Background

2.1. 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 that have focused on the maximization of the network lifetime, each taking into account a different approach to achieve it.

Routing is probably the most immediate issue addressed in order to accomplish this goal. A convenient choice of paths to route data may result in significant energy conservation. In [3], an energy-efficient metric for finding routes was proposed. In [4], nodes energy reserves were taken into account to route the traffic so that the nodes' drain-out is maximized. Other routing techniques are shown in many other studies, such as [5] and [6].

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 [7] and, later, in [8]. 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. This uneven energy dissipation may lead to an early death of some nodes, indeed resulting in a reduction of the network lifetime rather than an increase.

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 [9] and [10]. Energy load distribution can also be achieved by conveniently deploying the network nodes, as in [11]. In [12], the nodes are spaced non-uniformly as a function of their distance. Taking into account that nodes near the base station feel the effects of higher traffic more than other nodes, spacing is 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 the 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 [13], 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.

In [14], Cluster Heads for a mobile WSN are chosen according to a selection algorithm which takes into account characteristics of the nodes such as power energy and transmission rate. The clustering algorithm presented is periodically re-run to possibly find new nodes which might improve the network lifetime.

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 IoT in enterprise services might be found in [15]. However, this framework is not conceived to minimize the energy consumption, unlike the framework hereafter described.

2.2. Energy Consumption

Energy consumption in Wireless Sensor Networks is determined most of all by transmission and reception. As mentioned in [16]

$$
\begin{cases}
P_T(\delta) = P_{T0} + P_A(\delta) = P_{T0} + P_{Tx}(\delta)/\eta \\
P_R = P_{R0}
\end{cases}
\tag{1}
$$

where P_T and P_R are radio frequency power consumptions for transmitting and receiving respectively; P_A is the power consumption of the Power Amplifier (PA); δ is the distance between the transmitter and the receiver; P_{T0} and P_{R0} are the components of power consumption of the transmitting and receiving circuitry respectively; P_{Tx} is the output power at the antenna which, for reliable transmissions, depends on the distance δ; η is the drain efficiency of the PA.

Considering a channel in which the path loss component is predominant, and thus secondary effects such as multipath and Doppler can be neglected, the received power P_{Rx} can be expressed as

$$
P_{Rx} = \frac{P_{Tx}}{A \times \delta^\alpha}
\tag{2}
$$

where A is a parameter determined by the characteristics of the antennas (such as gain and efficiency) and α denotes the path-loss exponent, which is about 2 for free space. This kind of modelling is typical of free space propagation. Of course, the model might be extended to account for other fading effects.

From Equations (1) and (2)

$$
P_T(\delta) = P_{T0} + \frac{P_{Rx} \times A \times \delta^\alpha}{\eta}
$$

Considering $\varepsilon = P_{Rx_{min}} \times A$, where $P_{Rx_{min}}$ is the minimum reception power for a reliable communication

$$
P_T(\delta) = P_{T0} + \frac{\varepsilon \times \delta^\alpha}{\eta}
$$

This implies that the total power consumption for communicating between a transmitting node A and a receiving node B of a WSN could be written as

$$P_{AB} = P_{R0_B} + P_{T0_A} + \frac{\varepsilon_{AB} \times \delta_{AB}^\alpha}{\eta_A}$$

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

$$
\begin{aligned}
e^{tx}(k, P_{T0_A}, \eta_A, P_{R0_B}, \varepsilon_{AB}, \delta_{AB}) &= \frac{P_{AB} \times k}{R} \\
&= \frac{k}{R}\left(P_{R0_B} + P_{T0_A} + \frac{\varepsilon_{AB} \times \delta_{AB}^\alpha}{\eta_A}\right) \\
&= k \times (e_{R_B} + e_{T_A}(\delta))
\end{aligned}
\tag{3}
$$

where e_{T_A} is the energy to send one bit over a distance δ_{AB} and e_{R_B} is the energy to receive one bit.

The model described does not take into account mechanisms such as sleep schedule and route discovery, which may produce overhead. Thus it could thus be necessary to consider not just the single packet transmission, but also 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, which is given by the used hardware.

The processing energy consumption $e^{proc}(task, data^{in})$ is proportional to the complexity of the $task$—that is, the number of instructions needed to complete the $task$—and to the ingress data $data^{in}$—the higher the number of samples involved in the processing, the higher the energy consumption. This function $e^{proc}(.)$ can be determined on the basis of the device datasheet used. 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

$$e^{proc}(task, data^{in}) = smp^{in} \times M^{task} \times e^{instr} \tag{4}$$

A summary of the notation used throughout this paper is provided at the beginning of this document for quick reference.
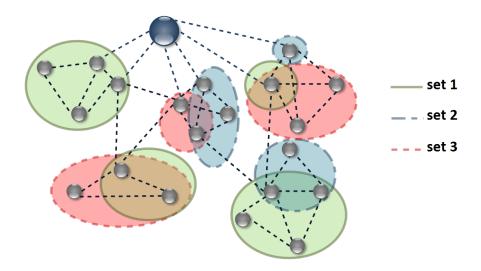
3. Problem Formulation

The goal of a WSN is to accomplish a given number of operations mostly based on some measurements performed on the relevant environment. In our scenario not all the nodes have the same capacities, as represented in Figure 2, where three sets of possible tasks have been considered, which could be: data processing, temperature measurement and video monitoring. Given the status of the network in terms of node capacities, 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 application impact on the infrastructure battery lifetime.

In our modelling, the number of nodes $X = \{x_1, ..., x_i, ..., x_N\}$ in the WSN is denoted with N, where the node 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 there is 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 $\boldsymbol{\Delta} = (\delta_{ij})$, 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 ε_{ij} introduced in Section 2.2, calculated for each combination of adjacent nodes i and j, where i reaches the sink though j. If $\varepsilon_{ij} = \infty$, nodes i and j are not adjacent;

- the set of characteristic parameters $V_i = \{P_{R0i}, P_{T0i}, \eta_i\}$ of every node x_i, which are useful to compute the transmission energy consumption as defined in Equation (3) in Section 2.2;

- the set $F = \{f_0, ..., f_w, ..., f_W\}$ of tasks, which encompasses all the tasks that can be performed by any node in the network. For instance, 0 might correspond to "temperature sensing in the area 1", 1 to the "temperature sensing in the area 2", 2 to the "pressure sensing in the area 3", 3 to the "spatial averaging" (which means performing the average of the sensed data arriving from different geographical areas), 4 to the 'temporal averaging" (which means performing the average of sample values sensed by the same sensor at different instants of time), 5 to the "only transmission", 6 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 $D_i = \{d_{i1}, ..., d_{im}, ..., d_{il_i}\}$, with $D_i \subseteq F$, where the elements of D_i are the tasks that the node x_i is able to perform.

Figure 2. Example of a WSN. Nodes belonging to set 1 might perform task f_1; nodes in set 2 might perform task f_2; nodes in set 3 might perform task f_3.

We assume that a given operation O has to be deployed in the network, which can be decomposed into a sequence of distributed tasks. This could represent diverse operation, including: computing the average of the temperature in certain geographical areas, measuring the light intensity in a room, video-surveillance of a specific geographical area, or a combination of these. In the following, we rely

on a specific reference application, which we name *spatial and temporal monitoring*: a spatial and temporal mean operation over an hour is performed on the temperature values sensed by the sensors every 10 min by the sensors from 3 different locations; the average values are stored in the sink. We use this example to better explain our modelling.

Three significant parameters can be associated with the operation to be deployed:

- the total cost value E^{tot}, which takes into account the energy consumption;

- the sequence $C = \{c_1, ..., c_l, ..., c_L\}$ of the sub-operations that must be executed by the nodes to perform the operation O. If, for instance, O is our reference example of *spatial and temporal monitoring*, some of the sub-operations in C will certainly be "temporal averaging" 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 order of priority: if a node has to execute both c_1 and c_2, the former must be executed before the latter. With reference to our *spatial and temporal monitoring* example, c_1, c_2 and c_3 are the sensing sub-operations: c_1 is "temperature sensing in area 1", c_2 is "temperature sensing in area 2" and c_3 is "temperature sensing in area 3"; because every sensor only belongs to one area, the order of the sensing sub-operations is irrelevant. Rather, it is important that data is gathered before any other sub-operation is performed on it; thus, the sensing sub-operations are the first elements in C. If a node has to calculate both the temporal and spatial mean of the received values, it has to first work out the temporal mean on the data received from each path, and then it has to perform the spatial mean on it. Therefore, c_4 is "temporal mean" and c_5 is "spatial mean". Finally, c_6 is "only transmission" and c_7 is "no actions". All the sub-operations for the example are summarized in Table 1;

- the set $S = \{s_1, ..., s_i, ..., s_N\}$, 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 corresponds to "no actions". For this reason, "no actions" is necessarily included in set F. The following must always be verified:

  - 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

  $$s_i \neq \text{"no actions"} \ \vee \ a_{ij} = 1 \Rightarrow s_j \neq \text{"no actions"} \tag{5}$$

  - 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"

  $$s_i \equiv \text{"no actions"} \ \forall i : a_{ij} = 1 \Rightarrow s_j = \text{"no actions"} \tag{6}$$

Table 1. Sub-operations for the *spatial and temporal monitoring example*.

c_i	Description
c_1	Temperature sensing in area 1
c_2	Temperature sensing in area 2
c_3	Temperature sensing in area 3
c_4	Temporal mean
c_5	Spatial mean
c_6	Only transmission
c_7	No actions

Thanks to 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, and therefore the transmission energy consumption would be lower as well; 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 framework described further takes a high-level code as input, evaluates which combination of statuses $S = \{s_1, ..., s_i, ..., s_N\}$ permits the operation O to be performed with the lowest possible energy consumption E^{tot}, and finally 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. The problem addressed is then defined as the set of statuses S that minimizes the impact of the application on the network lifetime. In the following, we elaborate the considered scenario by defining further constraints that solve the problem.

4. 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 a single action in a deployment solution; the cost functions built on the basis of the energy consumption formulae; the network lifetime maximization procedure; a summary of the proposed framework. Note that the modelling that we propose in this work and that we present in this Section is aimed at evaluating all the possible solutions of application deployment in terms of data transmission and processing. The parameters, constraints and cost functions are introduced for the sole aim of evaluating the viability of the solution.

4.1. Constraints on 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

$$\mathbf{T}_i^{out} = p(\mathbf{T}_i^{in}, s_i) \tag{7}$$

The output traffic is then sent to the next node towards the sink.

The data generated by p in the node x_i is modelled by the H-dimensional vector $\mathbf{T}_i^{out} = (t_{i1}^{out}, ..., t_{ih}^{out}, ..., t_{iH}^{out})$, 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

$$\mathbf{T}_j^{in} = \bigcup_{i=1}^{N} \mathbf{T}_i^{out} \times a_{ij} \times z_i, \qquad \text{with } z_i = \begin{cases} 0 & s_i \equiv \text{"no actions"} \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

As defined by Equation (7), 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 Equation (5);

- 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.
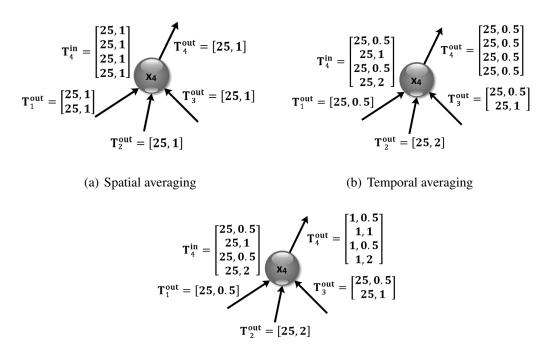
Referring to the *spatial and temporal monitoring* example, processing can be a spatial averaging, a temporal averaging, or a combination of both. In a spatial processing, the samples coming from different paths are processed together, as shown in an example in Figure 3(a). Here, four flows of 25 bits per second are received by node 4, which are then averaged to produce a single flow of 25 bits per second. Accordingly, 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} and the frequency f_{j1}^{out} are equal to those of each input flows. It must be noted that, in general, k_{j1}^{out} is not necessarily equal to the number of bits of each input flow, but it may be different according to the processing output.

Differently, the temporal averaging is performed on every traffic flow in $\mathbf{T}_j^{in}$. The resulting $\mathbf{T}_j^{out}$ contains the same number of traffic flows as 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 averaging frequency associated to the node status s_j. Indeed, we may have different status codes associated to the temporal averaging, each one distinguished by a different processing frequency. Figure 3(b) shows an example for

this kind of processing, where, in this case, we assume the node status corresponds to temporal averaging with frequency 0.5 Hz.

Other processing tasks can be performed on every single sample of each received traffic flow without involving other samples. This is the case, for instance, where one must evaluate whether the received values exceed a given threshold or not, consequently transmitting a boolean output value. The only thing that changes in the output traffic flows is the number of bits per sample; therefore, $\mathbf{T}_j^{out}$ 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}. Figure 3(c) shows the traffic flows for the described processing.

Figure 3. Examples of data processing in node 4 that receives input traffic from nodes 1–3. In these sketches we show the input and output traffic for: spatial averaging (**a**); temporal averaging (**b**); and single sample processing (**c**).

(a) Spatial averaging

(b) Temporal averaging

(c) Single sample processing

There are many other processing tasks that can be performed in a given network and they are coded in F as described in the previous Section. 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.

4.2. Virtual Nodes

It is possible that a single node has to perform more than one task. For instance, if operation O is a temporal and spatial average of the temperature values measured in different geographical areas, it may happen that a single node has to compute both spatial and temporal average values on the received data. To take into account this type of scenario, we rely on the concept of virtual nodes. These are copies of real nodes, each one able to perform only a specific sub-operation and sending the resulting data at

zero-energy cost to the next virtual node (except the last one that sends the data to the next node). We define the set of sub-operations that a single node x_i can perform consecutively for the implementation of the operation O

$$G_i = D_i \cap C \setminus L \tag{9}$$

where L is the set of sub-operations that the node cannot execute with the others, which are "only transmission" and "no actions". These two are kept outside the set G_i because it cannot happen that a sequence of sensing/processing tasks are followed by any of the tasks in L. If $|G_i| > 1$, the node x_i is divided into $|G_i|$ virtual nodes. Each virtual node is created so as to be able to execute only one of the sub-operations in G_i, to which the sub-operations in L are added. Each virtual node can then be assigned to perform only one of these three sub-operations (the one taken from G_i plus the two in L). Hence, the new network will have a number N^{vir} of total nodes x_v^{vir}

$$N^{vir} = \sum_{i=1}^{N} \Gamma_i, \qquad \text{with } \Gamma_i = \begin{cases} |G_i| & \text{if } |G_i| > 0 \\ 1 & \text{otherwise} \end{cases}$$

The set of possible sub-operations for x_v^{vir} is D_v^{vir}. Figure 4 draws an example of sequence of virtual nodes for node x_4, which is substituted by nodes x_4^{vir}, x_5^{vir} and x_6^{vir}. Let us refer to the *spatial and temporal monitoring* example and the associated set of sub-operations C listed in Table 1. Additionally, let us assume that $\{c_1, c_4, c_5, c_6, c_7\} \subset D_4$, that is: the node x_4 can sense the area 1, perform a temporal averaging, a spatial averaging, only transmit the received data and perform no actions (these last two tasks are always included). Because $L = \{c_6, c_7\}$
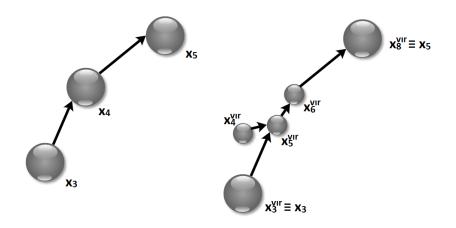
$$\begin{aligned} D_4^{vir} &= \{c_1, c_6, c_7\} \\ D_5^{vir} &= \{c_4, c_6, c_7\} \\ D_6^{vir} &= \{c_5, c_6, c_7\} \end{aligned} \tag{10}$$

A new adjacency matrix $\mathbf{A}^{vir}$ is defined to incorporate additional virtual nodes. Such matrix is built simply by substituting the real node with the sequence of virtual nodes, so that the first virtual node is connected to the nodes from which x_i received the data, while the last virtual node is connected to the node to which x_i sent the data. The other nodes are connected in sequence. An exception happens if the real node can also perform some sensing functions. In this case, the corresponding virtual node is kept outside this sequence and it merely sends the data to the subsequent virtual node. This rule has been introduced because the sensing operation does not need any data from other nodes. This scenario is what happens in the example shown in Figure 4, where the temperature sensing function can be performed by node x_4^{vir}.

As defined in Section 3, sub-operations are in priority order. For this reason, virtual nodes have to be placed such that the first performs the sub-operation with the highest priority, and the last executes the sub-operation with the lowest priority and sends the data to the rest of the network. With reference to Figure 4, the highest priority is given to the sensing action, because whether there are any data to be sensed, the information should be gathered before processing it. For this reason, sub-operation c_1 is included in D_4^{vir}. If data received by node x_5^{vir} has to be processed, temporal averaging should be performed before spatial averaging. Thus, priority of the former must be higher than the latter, and this

is the reason why temporal averaging c_4 is an element of set D_5^{vir}, while spatial averaging c_5 is an element of set D_6^{vir}. The adjacency matrix $\mathbf{A}^{vir}$ has to be defined accordingly.

Figure 4. Example of virtual nodes substituting node x_4. Node x_4 can perform a sensing action.

A set of characteristic parameters $V_i^{vir} = \{P_{R0i}^{vir}, P_{T0i}^{vir}, \eta_i^{vir}\}$ has to be associated with each node. Taking into account the virtual nodes substituting node x_i, the transmission cost from a virtual node to the other must be null. Therefore, only the last virtual node, that has to transmit the data to the network, has the same characteristic parameters V_i^{vir} as x_i: the virtual nodes before it must have the parameters P_{R0i}^{vir} and P_{T0i}^{vir} set to 0.

New matrices $\boldsymbol{\Delta}^{vir}$ and $\mathbf{E}^{vir}$ are defined, where the values of their elements for adjacent virtual nodes from the same original node are null. From now on, we will be referring only to a network with virtual nodes; however, to make the presentation clearer, we will skip subscript "vir", as it is unnecessary.

4.3. 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 in our framework we assume the network may be employed to perform more than one operation simultaneously, there is no sense in computing the network lifetime since it is affected by other applications which are not considered in the same analysis. For this reason we try to minimize the energy consumption for the operation under analysis, allowing the network administrator to also include a parameter that takes into account the current node battery energy level, as it is shown in the following.

We consider three cost functions: one for the sensing, one for the processing and one for the transmission. The sensing cost function for node x_i is expressed as

$$E_i^{sens} = f_i^{out} \times \gamma_i \times e_i^{sens} \times y_i, \qquad \text{with } y_i = \begin{cases} 1 & \text{if } s_i \equiv \text{sensing code} \\ 0 & \text{otherwise} \end{cases} \qquad (11)$$

with e_i^{sens} representing the sensing energy consumption as defined in Section 2.2. Recall that f_i^{out} is the node output traffic frequency, which also represents the sensing frequency. The parameter γ_i is

a coefficient in inverse proportion to the residual energy of the node which can be set to drive the deployment of the application towards nodes with higher residual energy levels, as anticipated above. When performing our experiments, we have set γ_i to 1 when the battery is fully loaded, while we set γ_i to 5 when the battery level is lower than 20% of the total charge. From 1 to 5, γ_i changes linearly.

We defined the processing cost function as follows

$$E_i^{proc} = \sum_{h=1}^{H} f_{ih}^{out} \times \gamma_i \times e_{ih}^{proc}(c_{s_i}, \mathbf{T}_i^{in}) \times v_i \qquad \text{with } v_i = \begin{cases} 1 & \text{if } s_i \equiv \text{processing code} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where γ_i is the coefficient defined above, and e_{ih}^{proc} is the processing energy consumption defined by Equation (4), which depends on the characteristics of the node x_i, 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 Equation (8). 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 4.1. The number of samples to calculate e_i^{proc} is defined differently for each kind of processing detected in Section 4.1. For a spatial processing, the number of processed samples is equal to the number of ingress traffic flows; for a temporal processing, the number of processed samples for each traffic flow t_{ih}^{out} is the ratio between the frequency of arrival of the samples f_{ih}^{in} and the processing frequency f_{ih}^{out}; by definition, for a single sample processing the number of processed samples is 1.

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 Equation (5) 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

$$E_i^{tx} = f_i \times \gamma_i \times e^{tx}\left(\mathbf{T}_i^{out}, V_i, \bigcup_{j=1}^{N} V_j \times a_{ij}, \bigcup_{j=1}^{N} \varepsilon_{ij}, \bigcup_{j=1}^{N} \delta_{ij}\right) \tag{13}$$

where f_i is the transmission frequency, γ_i is the residual energy coefficient, and e^{tx} is the transmission energy consumption defined by Equation (3), 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 ε_{ij} concerning nodes x_i and x_j; the distance δ_{ij} between x_i and x_j.

Given Equations (11–13), the overall cost function for any operation O is

$$E^{tot} = \sum_{i=1}^{N} \left(E_i^{sens} + E_i^{proc} + E_i^{tx}\right) \tag{14}$$

4.4. Maximization of Network Lifetime

The goal is to find the set of the statuses $S = \{s_1, ..., s_i, ..., s_N\}$ of the nodes that minimize the network energy cost function.

Therefore, the optimization problem becomes

$$\text{minimize} \quad E^{tot} = \sum_{i=1}^{N} \left(E_i^{sens} + E_i^{proc} + E_i^{tx} \right) \tag{15a}$$

$$\text{subject to} \quad Q_l^{min} \leq \sum_{i=1}^{N} q_{il} \leq Q_l^{max} \qquad \text{with } q_{il} = \begin{cases} 1 & s_i \equiv c_l \\ 0 & \text{otherwise} \end{cases} \tag{15b}$$

$$\bigcup_v \{s_v\} = G_N \qquad \forall v : x_v \mapsto x_N \tag{15c}$$

The condition in Equation (15b) is a constraint on the minimum (Q_l^{min}) and the maximum (Q_l^{max}) number of nodes that have to perform the sub-operation c_l. This could be necessary, for example, when a given geographical area is monitored by a certain number of nodes, but the required information is not needed from all of them. If, for instance, the mean temperature value of an area monitored by 30 sensors is needed, we may prefer the temperature information to be gathered just by 10 of those sensors, in order to consume less energy. In this case, both Q_l^{min} and Q_l^{max} would be equal to 10, and the algorithm would choose the 10 sensors which weight less on the network lifetime, among the 30 sensors which are able to sense temperature in the required area. When this constraint is not needed for a sub-operation l, Q_l^{min} is null and Q_l^{max} is set to N.

The condition in Equation (15c) shows that the set of statuses of the virtual nodes corresponding to the original sink node x_N must correspond to the set $|G_N|$.

This implies that none of the virtual nodes corresponding to the original sink node can be in "only transmission" or "no actions" status, but they are in a processing status; therefore, if there is any data still to be processed, those virtual nodes have to process them.

The problem defined in Equation (15) is a Binary Integer Programming (BIP) problem: the unknown status of a node x_i can be defined as a $|C|$-dimensional binary array, where C is the set of sub-operations as defined in Section 3. Because every node can only have one status, which means that it can perform only one sub-operation among those that it is able to perform, 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 in Equations (11,12) of the node to the cost function.

BIP problems are classified as NP-hard. Their exact solution is usually found using branch-and-bound algorithms. The worst case complexity of branch-and-bound algorithms is the same as the complexity of exhaustive search, which means that its complexity scales exponentially with the problem size. In the case under consideration, the problem size is dominated by the number of sub-operations $|C|$ and the number of virtual nodes N^{vir}. Therefore, the worst case complexity would be $O(2^{|C| \times N^{vir}})$. Nevertheless, in most cases branch-and-bound is more efficient compared to exhaustive search. Furthermore, our problem's structure is such that only one element of the $|C|$-dimensional array representing the status of each node is nonzero. This condition allows to reduce the search space to $O(N^{vir|C|})$. It has to be noted that commercial mathematical programming solvers such as CPLEX [17] or Xpress Optimization Suite [18] are claimed to use optimized branch-and-bound algorithms whose complexity scales linearly with the problem size.

In order to further reduce complexity, heuristic algorithms might be used as well, obtaining sub-optimal solutions which may be considered sufficient in most cases.

4.5. The Proposed Framework

Given a network similar to the one in Figure 2, 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 the distance between any two of them is, as well as the routing table. In order to compute the cost function in Equation (14), further information is 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.

In summary, the algorithm performs the following steps:

1. define N x_i, sets F, D_i, V_i and matrices $\mathbf{\Delta}$, $\mathbf{E}$ and $\mathbf{A}$;

2. define set O and the corresponding set C;

3. define sets G_i;

4. define the new network with N^{vir} virtual nodes x_i^{vir}, and new sets D_i^{vir}, V_i^{vir}, $\mathbf{\Delta}^{vir}$, $\mathbf{E}^{vir}$ and $\mathbf{A}^{vir}$;

5. given in Equation (15), solve it with a linear programming solver, in order to find set S.

The solving algorithm has been implemented in Mosel language, and the solution has been found using Xpress Optimization Suite. The binary array $\Sigma_v^{vir} = \{\sigma_{v1}^{vir}, ..., \sigma_{vl}^{vir}, ..., \sigma_{vL}^{vir}\}$ has been associated to each node x_v^{vir}, where L is the cardinality of C, that is, the number of sub-operations which must be executed to perform operation O, as described in Section 3. The elements of Σ_v^{vir} must satisfy the following constraints:

- the element σ_{vl}^{vir} can be equal to 1 if and only if x_v^{vir} is able to perform the sub-operation $c_l \in C$, which means that σ_{vl}^{vir} cannot be equal to 1 if the sub-operation c_l is not an element of the set of tasks D_v^{vir} that the node x_v^{vir} is able to perform

$$c_l \notin D_v^{vir} \Rightarrow \sigma_{vl}^{vir} \neq 1 \tag{16}$$

- only one element in Σ_v^{vir} can be equal to 1

$$\sum_{l=1}^{L} \sigma_{vl} = 1 \tag{17}$$

The elements of the array built this way are the weights y_i and v_i of the energy contributions in Equations (11,12) defined in Section 4.3.

The optimum way to perform the operation O at issue and spend the least amount of energy as possible will then be found. The node or the combination of nodes that are able to perform it and consume the minimum amount of energy will be chosen; then, a low level code describing which tasks each node has to perform will be developed and distributed to the appropriate nodes.

5. Performance Analysis

5.1. Test Cases and Simulations Setup

To evaluate the effectiveness of the proposed strategy, 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]:

1. **case1**: uniform energy consumption and uniform initial energy (UC-UE) at each node (equal characteristic parameters and battery life for every node);

2. **case2**: non-uniform energy consumption and uniform initial energy (NUC-UE) at each node (different characteristic parameters but same battery life for every node). The energy consumption of the nodes have been assigned according to a uniform distribution from 60% to 140% of the energy consumption for case 1;

3. **case3**: uniform energy consumption and non-uniform initial energy (UC-NUE) at each node (same characteristic parameters but different battery life for every node). The battery charge has been assigned randomly according to a uniform distribution from 20% to 100% of the total charge.

We monitored a rectangular-shaped outdoor environment (e.g., a vineyard, a seaport, a tourist plaza), divided into areas of 25 m^2, where the nodes have been deployed with different densities:

- 0.2 nodes per square meter;

- 0.3 nodes per square meter;

- 0.4 nodes per square meter.

The nodes have been positioned randomly, following a uniform distribution. Each node is equipped with sensors for the measurement of temperature, humidity, PH and light exposure. The data are sent to the Coordinator, which has identification number N.

We have focused our analysis on the following two operations:

(OpA) calculation and storage in the sink of the (temporal and spatial) mean values of temperature, humidity, PH and light exposure over an hour, starting from the values sensed every 10 min from every area;

(OpB) aggregation of traffic coming from different areas of the network, carrying temperature, humidity, PH and light exposure values to the Coordinator for later analysis by qualified staff.

We assumed that each sensed value is represented as a double numerical value, which is 64 bits long. Note that these two operations have been selected to compare the scenarios in which the network is required to perform significant data processing (OpA) and nodes have to perform only basic processing on the data, yet can significantly reduce the amount of transmitted data by aggregating the sensed samples (OpB).

For both OpA and OpB, the first sub-operations are the sensing ones: c_1 is the "temperature, humidity, PH and light exposure sensing for area 1", c_2 is the "temperature, humidity, PH and light exposure

sensing for area 2", and so on for every area. In order not to weigh down the text with alternatives, we suppose there are just 2 areas, which is not the case of the simulation scenario. Similarly to the *spatial and temporal monitoring example* in Section 3, for OpA c_3 is "temporal mean", c_4 is "spatial mean", c_5 is "only transmission" and c_6 is "no actions". In addition to c_1 and c_2, the sub-operations for OpB are: c_3 as "aggregation of samples", c_4 as "only transmission" and c_5 as "no actions". The sub-operations for the two operations are summarized in Table 2.

The nodes communicate using IEEE 802.15.4 radio interfaces on the 2.4 GHz frequency band. To keep things simple, any possible overhead has not been taken into account.

We have simulated the resulting scenarios in MatLab environment, where we have implemented the proposed algorithm together with alternative approaches as discussed in the following Subsection. The main setup parameters are listed in Table 3 ([16,19,20]).

Table 2. Sub-operations for OpA and OpB, for two monitored areas.

c_i	OpA	OpB
c_1	Temperature, humidity, PH and light exposure sensing for area 1	
c_2	Temperature, humidity, PH and light exposure sensing for area 2	
c_3	Temporal mean	Aggregation of samples
c_4	Spatial mean	Only transmissions
c_5	Only transmission	No actions
c_6	No actions	——

Table 3. Simulation setup parameters.

Parameter	Value
RF frequency	2,400 MHz
Bit rate	250 kbps
Programmable output power range	Programmable in 8 steps from approximately -24 to 0 dBm
Receiver sensitivity	-94 dBm
P_{R0}	59.2 mW
P_{T0}	26.5 mW
η	50%
A	14 dB
e^{instr}	1 nJ
Packets header	12 bytes
Packets maximum payload	125 bytes

5.2. Analysis of Case Studies

The optimization algorithm has been applied to each of the cases mentioned in Section 5.1. The resulting cost value has been compared with:
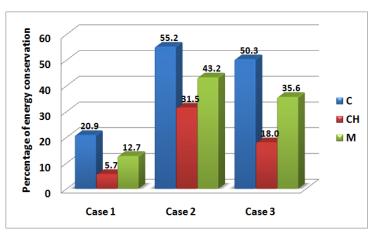
1. the cost value that is obtained if data are processed only by the Coordinator. This means that each traffic flow generated by the sensors is sent to the Coordinator without any processing at the intermediate nodes. We refer to this comparison with the letter "C";

2. the data are processed (whenever needed) by every Cluster Head (node receiving flows from different sensors) found in the path to the Coordinator. We refer to this comparison with the letters "CH";

3. the mean cost value for all possible solutions that might be detected. This is introduced to make a comparison with a possible solution where the processing of the data is performed on fixed nodes, which is expected to bring results corresponding to the median solution. We refer to this comparison with the letter "M".
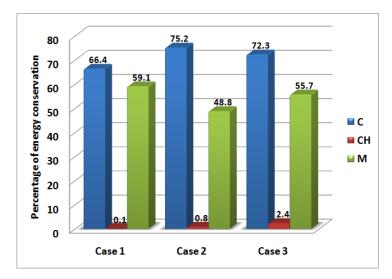
These comparisons are expressed as a percentage of the energy conservation that would result using the proposed technique with respect to the alternatives one.

Figure 5 shows the results for the two operations.

Figure 5. Percentage of energy conservation using the framework, for **(a)** OpA and **(b)** OpB, in cases UC-UE, NUC-UE and UC-NUE, comparisons C, CH and M, for a node density of 0.3 nodes per m^2.
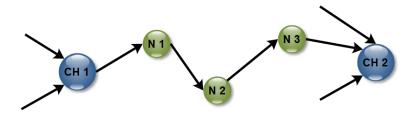

(a) OpA

(b) OpB

The two graphs show significant improvements of the proposed strategy with respect to the alternative ones with an average improvement of 36.3%. Limited benefits are observed in case UC-UE for both OpA and OpB. In fact, when all the nodes have the same parameter, and thus have a uniform energy consumption and the same initial energy, the choice of which node will perform the processing boils down to which Cluster Heads will do it. To illustrate the scenario, we refer to Figure 6, which depicts a Cluster Head CH 1 connected to the Cluster Head CH 2 by nodes 1, 2, and 3. Note that the Cluster Heads are just nodes that receive more than one traffic flow from different links. Because processing in CH 1 weights on the network as much as processing on node 1, node 2 or node 3, any processing of the data before arriving to CH 2 is more energy conserving. Processing the data on CH 1 ensures spending less transmission energy than processing data on nodes 1, 2 or 3. Accordingly, the CH approach allows for obtaining results similar to the ones we obtained with our approach in case UC-UE. We get slightly better results because when Cluster Heads are close to each other sometimes it is better to perform the processing only in the second Cluster Head rather than in both of them.

Figure 6. Example of a transmission from Cluster Head CH 1 to Cluster Head CH 2.

On the contrary, in cases NUC-UE and UC-NUE, devices' energy consumption does not weight the same amount on the entire network. This means that the nodes chosen by the proposed algorithm to perform the processing will be those that weight less on the network, regardless of whether they are Cluster Head or not. Therefore, the detection of the lower cost solution determines the best results, in terms of energy consumption, for networks with heterogeneous parameters, which are the most common type of networks in real scenarios.

The benefits with respect to the approach CH are lower than the case of using approaches C and M. In fact, literature dictates that the use of Cluster Heads is a convenient solution, because the aggregation of frames coming from different paths leads to a reduction in network energy consumption. For this reason, when using Cluster Heads the cost is much lower, compared to sending every single frame to the Coordinator, or to the average of the other possible solutions; this determines less difference from the optimization algorithm solution, and thus a lower energy conservation. However, this approach requires every node in the network to be able to perform data processing, which is not always the case. In any case, our approach is proven to always outperform the CH approach.

It could be noted that for OpA, in which processing is more elaborate and the number of instructions for every process greater, energy conservation is higher than it is for OpB. In fact, as could be expected, the lower the energy cost necessary for the processing, the more convenient it is to process the data in every Cluster Head encountered.
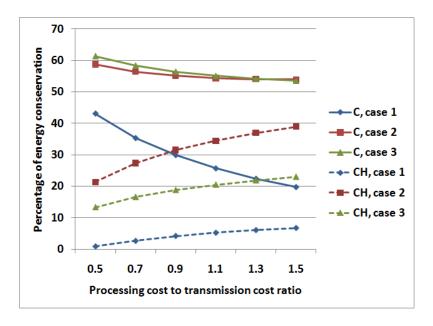
This fact is demonstrated by the results shown in Figure 7, which depicts the percentage of energy conservation while the ratio between the processing cost and the cost to transmit 137 bytes of data

increases. The distance considered is equal to the average distance of all the nodes from the Coordinator. Comparison has been made both in the case that data are processed only by the Coordinator (solid lines) and cases in which data are processed by every Cluster Head (dashed lines). In the former, energy conservation decreases when processing cost to transmission cost ratio increases. In fact, as could be expected, when the processing cost increases with respect to the transmission cost, it becomes more convenient to transmit data rather than process data. On the other hand, when compared with the CH approach, energy conservation increases when the processing cost to transmission cost ratio increases. In fact, 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. Table 4 shows the results for OpA and OpB, for different node densities of 0.2 and 0.4 nodes/m^2. The resulting tendency of an improved energy conservation when node density increases is basically due to two factors:

- in cases NUC-CE and UC-NUE, when the number of nodes in the same area increases, it is more likely that among neighbouring nodes there are node where the processing cost is lower;

- the higher the number of nodes in the same area, the higher the number of clusters formed, and therefore the bigger the amount of data that can be processed before they arrive to the Coordinator, reducing the energy cost.

It may be inferred from the results that using the framework would be particularly energy conserving when data from different nodes have to be processed together, the processing is pretty complex, and the energy consumption or the initial energy is not uniform for the network.

Figure 7. Percentage of energy conservation with respect to the ratio between processing cost and the cost to transmit 137 bytes of data, for cases UC-UE, NUC-UE and UC-NUE, for a density of 0.3 nodes per m^2. 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.
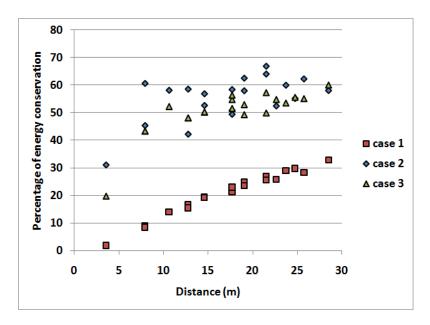
Table 4. Percentage values of energy conservation using the framework, for OpA and OpB, in cases UC-UE, NUC-UE and UC-NUE, comparisons C, CH and M, for a node density of 0.2 and 0.4 nodes/m^2.

	Node density [nodes/m^2]	Case 1 [%]			Case 2 [%]			Case 3 [%]		
		C	CH	M	C	CH	M	C	CH	M
OpA	0.2	19.5	5.5	11.7	50.0	29.3	42.2	47.5	16.4	33.6
	0.4	25.6	5.9	16.5	58.0	35.3	46.2	56.7	23.5	43.7
OpB	0.2	28.9	0.1	17.0	33.3	0.7	19.9	38.6	1.6	19.7
	0.4	30.8	0.1	18.2	37.8	1.0	21.4	40.9	3.5	21.1

In the considered framework we have not addressed the routing problem and we have assumed that the routing of the packets is working correctly. However, computation of the paths and re-computation in case of failure is for sure another cause of energy consumption. When the network experiences the failure of a node, that node must be bypassed and data addressed to it must be sent to the following node. To do it, an appropriate new routing path must be found. The number of packets exchanged to find a new routing path depends on the routing algorithm used by the network. If we consider a bad scenario where an average of 50 packets have to be sent among 10-hop distant nodes where each node is 2 m far from the other ones, in case of one node failure every hour, energy conservation would decrease by about 7.8%. It has been estimated that the decrease in energy conservation for OpA and for a density of 0.3 nodes/m^2 would be around 8.2% in case UC-UE, 12% in csae NUC-UE and 14.7% in case UC-NUE.

Figure 8. Percentage of energy conservation for every area of the network for OpA, comparison C, in cases UC-UE, NUC-UE and UC-NUE, for a density of 0.3 $nodes/m^2$, for an increasing distance of the area from the Coordinator.

A final observation may be made in regard to Figure 8, which depicts the percentage of energy conservation for OpA, comparison C, in cases UC-UE, NUC-UE and UC-NUE, for a density of 0.3 nodes/m^2, in relation to the distance of each area from the Coordinator. As could be expected, the greater the distance of the sources from the Coordinator, the more energy conservation is derived from the use of the framework.

Although they have not been reported, similar results have been obtained for all other cases.

6. Conclusions

In this paper we have studied the deployment of distributed applications in Wireless Sensor Networks and proposed a solution aimed at minimizing the impact of energy consumption on the network lifetime. We have considered a scenario where the nodes have dissimilar capacities in terms of sensing and processing. The resulting algorithm has been implemented to perform simulations in different scenarios and the results have been compared with alternative solutions. We observed significant improvements in terms of energy savings. We may therefore infer 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. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Networks* **2008**, *52*, 2292-2330.
2. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Networks* **2010**, *54*, 2787-2805.
3. Singh, S.; Woo, M.; Raghavendra, C. Power-aware routing in mobile *ad hoc* networks. In *Proceeding of the 4th ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, TX, USA, July 1998.
4. Chang, J.; Tassiulas, L. Energy conserving routing in wireless sensor networks. In *Proceedigns of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000*, Tel Aviv-Yafo, Israel, 26–30 March 2000; Volume 1, pp. 22-31.
5. Park, J.; Sahni, S. An online heuristic for maximum lifetime routing in wireless sensor networks. *IEEE Trans. Comput.* **2006**, *55*, 1048-1056.
6. Liu, F.; Tsui, C.; Zhang, Y.J. Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks. *IEEE Trans. Wireless Commun.* **2010**, *9*, 2258-2267.
7. Ramanathan, R.; Hain, R. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedigns of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000*, Tel Aviv-Yafo, Israel, 26–30 March 2000.
8. Calhoun, B.; Daly, D.; Verma, N.; Finchelstein, D.; Wentzloff, D.; Wang, A.; Cho, S.; Chandrakasan, A. Design considerations for ultra-low energy wireless microsensor nodes. *IEEE Trans. Comput.* **2005**, *54*, 727-740.

9. Ergen, S.C.; Varaiya, P. Optimal placement of relay nodes for energy efficiency in sensor networks. In *Proceedings of the IEEE International Conference on Communications, ICC '06*, Istanbul, Turkey, 1–15 June 2006.

10. Tang, J.; Hao, B.; Sen, A. Relay node placement in large scale wireless sensor networks. *Comput. Commun.* **2006**, *29*, 490-501.

11. Gao, Q.; Blow, K.; Holding, D.; Iqbal, Q.R. Radio range adjustment for energy efficient wireless sensor networks. In *Proceedings of SpringSim'07*, Norfolk, VA, USA, 25–29 March 2007.

12. Howitt, I.; Wang, J. Energy balanced chain in wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC 2004*, Atlanta, GA, USA, 21–25 March 2004.

13. Heinzelman, W.; Chandrakasan, A.; Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wireless Commun.* **2002**, *1*, 660-670.

14. Hong, T.; Wu, C. An improved weighted clustering algorithm for determination of application nodes in heterogeneous sensor networks. *J. Inf. Hiding Multimedia Signal Process.* **2011**, *2*, 173-184.

15. Spiess, P.; Karnouskos, S.; Guinard, D.; Savio, D.; Baecker, O.; de Souza, L.M.; Trifa, V. SOA-Based integration of the internet of things in enterprise services. In *Proceedings of the International Workshop on Cloud Services, IWCS 2009*, Los Angeles, CA, USA, 10 July 2009.

16. Wang, Q.; Hempstead, M.; Yang, W. A realistic power consumption model for wireless sensor network devices. In *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and* Ad Hoc *Communications and Networks*, Reston, VA, USA, 25–28 Sepember 2006.

17. ILOG. CPLEX User's Manual. Available online: http:// yalma.fime.uanl.mx/cplex11-manual/ Content/Optimization/Documentation/CPLEX/_pubskel/ XPlatform/User_man250.html (accessed on 9 June 2011).

18. FICO. Xpress Optimization Suite. Available online: http://www.fico.com/en/Products/DMTools/ Pages/FICO-Xpress-Optimization-Suite.aspx (accessed on 9 June 2011).

19. Chipcon. SmartRF CC2420 Datasheet. 2.4 GHz IEEE 802.15.4/ZigBee-Ready RF Transceiver. Available online: http://inst.eecs.berkeley.edu/ cs150/Documents/CC2420.pdf (accessed on 9 June 2011).

20. IEEE. Std 802.15.4™, 2006. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). Available online: http://www.google.com.hk/url?sa=t&source=web&cd=1&ved=0CB0QFjAA&url=http%3A%2F%2Fprofsite.um.ac.ir%2F hyaghmae%2FACN%2FWSNMAC1.pdf (accessed on 9 June 2011).