3D Mobility, Resizing and Mobile Sink Nodes in Reconfigurable Wireless Sensor Networks based on Multi-agent Architecture under Energy Harvesting Constraints

Hanene Rouainia[1] [a], Hanen Grichi[2,3] [b], Laid Kahloul[4] [c] and Mohamed Khalgui[2,3] [d]
[1]*Faculty of Sciences, El-Manar University, Farhat Hached University Campus B.P. n ° 94 - ROMMANA, Tunis, 1068, Tunisia*
[2]*School of Electrical and Information Engineering, Jinan University, Zhuhai, 519070, China*
[3]*INSAT Institute, University of Carthage, 1080, Tunisia*
[4]*LINFI Laboratory, Computer Science Department, Biskra University, Biskra, Algeria*
hanene.rouainia@fst.utm.tn, {hanen.grichi, laid.k.b, khalgui.mohamed}@gmail.com

Keywords: Reconfigurable Wireless Sensor Network, Reconfiguration, Multi-agent Architecture, 3D Mobility, Resizing, Mobile Sink Node.

Abstract: This paper deals with reconfigurable wireless sensor networks (RWSNs) to be composed of a set of sensor nodes, which monitor the physical and chemical conditions of the environment. RWSNs adapt dynamically their behaviors to their environment. The main challenge in RWSN is to keep the network alive as long as possible. We apply a set of solutions for energy problems by using 3D mobility, resizing and mobile sink nodes. These solutions are based on a multi-agent architecture employing a wireless communication protocol. Moreover, we develop an application named *RWSNSim* that allows us to simulate an RWSN and apply the proposed solutions. The performance of the proposed approach is demonstrated through a case study. The case study consists of surveying of fire in a forest which is simulated with *RWSNSim* application.

1 INTRODUCTION

Wireless sensor networks (WSNs) are defined as large networks to be composed of very small battery-operated devices (sensor nodes). WSNs are usually designed for specific systems to provide a solution to a wide range of applications from small-size to large-scale systems. These systems are implemented in many areas such as medical, environmental and structural monitoring (Habibu et al., 2014), (Rault et al., 2014).

Sensor nodes (SNs) are the principal entities in WSN. SNs are connected with each other via a wireless communication. There are two types of sensor nodes: mobiles and fixed ones. The specific feature in mobile nodes is the ability to move to different positions in the network. SNs communicate with each other wirelessly to execute a set of tasks and process data using simple microprocessors with limited computing resources (Habibu et al., 2014), (Garcia et al., 2009).

In wireless sensor networks, all the data collected by SNs are forwarded to a sink node. The sink node plays the role of gateway between sensor nodes and the data processing center. The placement of the sink node has a great impact on energy consumption and WSNs lifetime.

Wireless communication is an indispensable area in scientific and industrial communities. This technology is the fastest-growing segment of the communications industry (Goldsmith, 2005). It can be used with a minimum supervision, a reasonable cost as a widely distributed and a high-resolution atmospheric observation network (Messer et al., 2006).

The reconfiguration is defined in (Housseyni et al., 2017) as any procedure that allows reconfiguring the system to be feasible. It is usually performed to update software and hardware components in response to user requirements and dynamic changes in its environment such as hardware or software failures, activation/deactivation of a sensor node and the addition of a new task (Ben Salem et al., 2016), (Grichi

---

[a] https://orcid.org/0000-0001-7544-988X
[b] https://orcid.org/0000-0002-4601-3574
[c] https://orcid.org/0000-0002-9739-7715
[d] https://orcid.org/0000-0001-6311-3588

et al., 2014b), (Housseyni et al., 2017). There are several implementations of reconfigurable systems such as reconfigurable wireless sensor networks (Grichi et al., 2016), software reconfiguration in Multi-Robot systems (Fang Tang and Parker, 2005), (Chen et al., 2016), manufacturing systems (Zhang et al., 2015) and real-time embedded systems (Wang et al., 2011). There are three reconfiguration levels. The first one is the software reconfiguration which is applied to the software architecture at run time by changing the behaviour of nodes (add, remove, modify the scheduling of tasks or change the data used by the tasks). The second one is the hardware reconfiguration that consists of activation or deactivation of nodes or sensors. The last one is the protocol reconfiguration (data-routing reconfiguration) which permits the modification of data routing. It allows us to optimize the protocol by the update, removal, and the addition of exchanging messages between sensor nodes and also their routing paths (Grichi et al., 2016), (Grichi et al., 2014a), (Gharsellaoui et al., 2012).

The mobility is defined as the movement of mobile nodes to minimize the distance between sensor nodes. It applies also to the mobile sink nodes to supply the minimization of the total distance between the network entities which allows to reduce the energy consumption in the network. The zones resizing is defined as the variation of the number of zones using a modification of their size with their sets of local nodes (Grichi et al., 2017). A multi-agent architecture is a set of agents (servers, processes, robots or humans), located in a certain environment and interacting according to certain relationships. An agent is an entity characterized by being at least partially autonomous, and each agent has a specific management and reconfiguration scenario roles in its environment. The multi-agent architecture is used in RWSNs to allow us to execute the reconfiguration scenarios with an orderly and an optimal manner.

Wireless sensor networks work under different types of renewable energy resources (solar, vibration, wind energy, etc.) which is not frequently available. Thus, the main challenge in WSNs is to keep the network alive for the longest time possible when energy is missed without human intervention. Another challenge that appears when trying to solve the energy problem is to find out solutions without expensive costs and a long time.

Several related works, try to solve the energy problem in WSNs such as in (Mahendrababu and Joshitha, 2014) which proposed a solution to the energy hole problem in WSNs using Witricity (wireless electricity) to overcome the power constraints in WSN by wirelessly charging the nodes and introducing the charging layer into the basic sensor network protocol stack. The article (Anastasi et al., 2009) deals with the energy consumption of a typical sensor node and discusses the main directions to energy conservation in WSNs. It presents a systematic and comprehensive taxonomy of the energy conservation schemes and techniques for energy-efficient data acquisition.

The reconfiguration is proposed as a solution to preserve the energy in WSNs such that in (Grichi et al., 2014b) which considers that it makes the wireless sensor network satisfying the real-time and energy constraints considering the system performance optimization. As well as in (Housseyni et al., 2017) which proposes an intelligent multi-agent distributed architecture, taking into consideration the three forms of reconfiguration. It uses two communication protocols: intra-subsystem and inter-subsystem communication protocols to ensure the effectiveness of the proposed reconfiguration strategy. The paper (Chniter et al., 2018) considers that the reconfiguration scenarios can cause the violation of real-time or energy constraints as a critical run-time problem. Therefore, the authors propose a multi-agent adaptive architecture to resolve this problem by handling the dynamic reconfigurations and ensure the proper execution of concurrent real-time distributed tasks under energy constraints. While the paper (Gasmi et al., 2016) proposes a new pipelined approach with five steps to figure out the reconfiguration scenarios that need to be applied without altering the performance of the pipeline. Finally, the paper (Grichi et al., 2017) proposes a new run-time power-oriented methodology for dynamic resizing and 3D mobility to keep the network alive as much as possible at any time before the next recharging operation. The mobility and resizing of zones are used also as solutions for the energy problem in WSNs. The papers in (Son and Ahn, 2014) and (Ahn, 2009) deal with mobility which permits the minimizing of the total distance between sensor nodes in order to decrease the energy consumed by them. While the papers in (Ji et al., 2013) and (Erman et al., 2012) deal with the geographic resizing of zones which allows the merging of zones to ensure comprehensive coverage in the network with preserving the energy.

The mobile sink node is also used in several related works to minimize energy consumption in WSNs. The work in (Thomas and Mathew, 2018) presents an intelligent method to discover the optimal path for a mobile sink using a modified travelling salesman problem (MTSP). The paper in (Chao et al., 2019) proposes a mobile data collection scheme based on the high manoeuvrability of unmanned aerial vehicles (UAVs). UAVs are used as a mobile sink node in WSN water monitoring and transmit data wirelessly to collect monitoring node data efficiently and flexibly. The study in (Kang et al., 2019) proposes a new method to support an efficient location service for a mobile sink using the surplus energy of a solar-powered WSN. Moreover, the paper (Wang et al., 2019) considers that energy-efficient routing is crucial for applications that introduce WSNs. Therefore the authors present an energy-efficient routing schema combined with clustering and sink mobility technology. Finally, the paper (Zhong and Ruan, 2018) studies the energy-efficient routing method with multiple mobile sinks support to effectively alleviate the hot spot problem since sink node can move along certain trajectories, causing hot spot nodes more evenly distributed.

In the mentioned related works, the authors tried to solve the energy problem in WSNs by applying only one or two solutions together such as mobility of nodes (2D/3D mobility), resizing of zones, multi-agent architecture, mobile sink nodes or other solutions. All of these proposed solutions by the related works are not sufficient to provide a large amount of energy and they increase the cost in WSNs. Therefore, they do not extend the network lifetime for a long time.

In this paper, we propose a new solution based on a hierarchical multi-agent architecture to execute and manage reconfiguration scenarios in the network. Indeed, we apply the 3D mobility, resizing of zones and using the mobile sink nodes in RWSNs to keep the network alive the longest time possible. The 3D mobility solution can decrease the energy consumption by SNs in the network by minimizing the distance between nodes, so consequently decreasing the energy consumption at the send task (considered as the most task consumer of the energy).The resizing of zones is an ambitious solution by the combination of zones according to some conditions and consequently decreasing the number of zones, changing the size of the zones and their number of local nodes at runtime. We use also mobile sink nodes, which minimize energy consumption by moving towards the lowest nodes in terms of energy stored. Moreover, we propose an energy model, construct a system equation, presents some protocols and algorithms to apply and manage the proposed solutions. Finally, we develop a Java application called *RWSNSim* which permits to simulate a RWSN and applying the different forms of reconfiguration, 3D mobility, resizing of zones using mobile sink nodes based on multi-agent architecture. It allows also to display the difference between the network lifetime with and without the proposed solutions which enables us to establish the evaluation of performance. These proposed solutions are applied to an experimentation that consists of a case study on a reconfigurable wireless sensor network where the SNs monitor the temperature and CO2 values in a forest to avoid fire risks. The experimentation shows the performance and the reliability of our contribution where the earnings are estimated by a 248% increase in network lifetime. The originality of this paper is the unification of a set of solutions in the same contribution to increase the performance of these solutions.

This paper is structured as follows. After the introduction section that presents the main challenges in WSNs, the state of the art and the contribution of this paper. The background is presented in section 2. Section 3 describes the contribution which is the application of 3D mobility and resizing under energy constraints using mobile sink nodes based on a multi-agent architecture. An experimentation is presented in section 4 showing a case study, an evaluation of performance and a discussion. Finally, the conclusion is drawn in section 5.

2 BACKGROUND: FORMALIZATION OF WSNs

We define in this section the different components of the RWSNs such as sensor nodes, mobile sink nodes and agents. Moreover, we present the energy model that consists of the formalization of energy production and consumption in each sensor node. These formalizations is presented in formula (2) and (3). Finally, we display the energy problem.

2.1 Reconfigurable WSNs

We consider that R is a reconfigurable wireless sensor network composed of a set of zones $S_Z = \{ \bigcup_{k=1}^{Nb_Z} \{Z_k\}\}$ where Z_k is a zone in R and Nb_Z is the total number of zones in R. Each zone is composed of a set of nodes $S_N(Z_k) = \{ \bigcup_{i=1}^{Nb_N^k} \{N_i^k\}\}$ where N_i^k is a node in Z_k and Nb_N^k is the total number of nodes in Z_k. We have two types of nodes: *i)* mobile nodes $S_{MN} = \{N_i^k| \;\; k \in [1..Nb_Z], \;\; i \in [1..Nb_N^k]\}$ and *ii)* fixed nodes $S_{FN} = \{N_j^k| \;\; k \in [1..Nb_Z], \;\; j \in [1..Nb_N^k],$ $S_{MN} \cup S_{FN} = S_N \;\; and \;\; S_{MN} \cap S_{FN} = \varnothing\}$, where S_N is the set of nodes in R.

Each node N_i^k has a set of sensors $S_{Sens}^{N_i^k}$ that monitor the physical and chemical conditions of the environment such as (temperature, the density of CO2 gas in the air, the humidity in the environment, etc.). Thus, $S_{Sens}^{N_i^k} = \{Sens_j^{N_i^k} \mid\ i \in [1..Nb_N^k],\ k \in [1..Nb_Z]\ \ and\ \ j \in\ [1..Nb_{Sens}]\}$ where $Sens_j^{N_i^k}$ is a sensor in N_i^k and Nb_{Sens} is the total number of sensors in each node in R. A sensor node has two batteries, the first one is the principal battery $B_{pr}(N_i^k)$ and the second one is the additional battery $B_{add}(N_i^k)$. The principal battery is rechargeable by the additional battery and this last one is rechargeable from the harvesting energy. Each node N_i^k in the 3D WSN has three coordinates $(x_{N_i^k}, y_{N_i^k}, z_{N_i^k})$ representing its position in R, it has also a capture field represented by a sphere with radius r.

In each zone Z_k, we have a set of mobile sink nodes $S_{SN}(Z_k) = \{SN_m^k \mid\ k \in [1..Nb_Z]\ \ m \in [1..Nb_{SN}^k]\}$ where Nb_{SN}^k is the total number of mobile sink nodes in Z_k. The paper (Zhong and Ruan, 2018) proves that three sink nodes in each zone are more suitable because of its expensive cost, and their impact is decreased when their number increases.

In RWSNs, a multi-agent architecture is used to execute the reconfiguration scenarios. It composes of three types of agents with various roles and responsibilities. The first type is the controller agent Ag_{Ctrl} that controls the whole network and makes the important decisions in the network such as the application of zones resizing or the processing of the captured values. The second type of agents are deployed as a set of zone agents $\{Ag_k, k \in [1..Nb_Z]\}$. The roles of zone agents consist of a set of tasks such as the application of 3D mobility and the demand of the controlled values from the sink nodes in its zone. The last type of agent is a set of node agents or slave agents in each zone Z_k $\{Ag_i^k, i \in [1..Nb_N^k]\ \ k \in [1..Nb_Z]\}$. Each node agent controls a node and makes a set of decisions like the activation/deactivation of the node. The controller agent and zone agents are servers equipped with large charge batteries and the node agent is installed on the node itself. The zone agent equipped also by a set of sensors $S_{Sens}^{Ag_k} = \{Sens_j^{Ag_k} \mid\ k \in [1..Nb_Z]\ \ j \in [1..Nb_{Sens}^{Ag_k}]$ where $Nb_{Sens}^{Ag_k}$ is the total number of sensors in Ag_k that monitor the physical and chemical conditions of the environment (Grichi et al., 2017).

2.2 Energy Model

We consider that each sensor node executes a set of tasks $T = \{\tau_1, \tau_2, ..., \tau_{Nb_\tau^{N_i^k}}\}$ where $Nb_\tau^{N_i^k}$ is the total number of tasks that can be executed by N_i^k. We associate to each $\{\tau_a \mid\ a \in [1..Nb_\tau^{N_i^k}]\}$ a trilogy formalized by $(t_{exec}(\tau_a), e_c(\tau_a), p_{\tau_a}(N_i^k))$ where $t_{exec}(\tau_a)$ is the execution time of τ_a, $e_c(\tau_a)$ is the energy consumption by τ_a and $p_{\tau_a}(N_i^k)$ is a boolean function formalized by:

$$\begin{cases} p_{\tau_a}(N_i^k) = n\ \ if\ \ \tau_a\ \ is\ \ executed\ \ n\ \ times \\ \qquad\qquad\qquad p_{\tau_a}(N_i^k) = 0\ \ if\ \ not \end{cases} \quad (1)$$

The Table 1 summarizes the most energy-consuming tasks that can be executed by N_i^k.

Table 1: Set of the most energy-consuming tasks.

τ_a	Description
$\tau_1 = Deactivate(N_i^k)$	Deactivate node N_i^k
$\tau_2 = Activate(N_i^k)$	Activate node N_i^k
$\tau_3 = Capture(Sens_j^{N_i^k})$	Capture the physical and chemical conditions of the environment by sensor $Sens_j^{N_i^k}$
$\tau_4 = Send(N_i^k)$	Send a captured values from N_i^k to its successor.
$\tau_5 = Recept(N_i^k)$	Recept a message by N_i^k from its predecessor.
$\tau_6 = Move(N_i^k)$	Moving to another position where $N_i^k \in S_{MN}$.
$\tau_7 = Resize(N_i^k)$	if N_i^k is in a zone included in a resizing task which is applied by Ag_{Ctrl}.

We assume that we can predict the approximate value of the instanteneous energy consumption by N_i^k for near future depending on the set of most-consuming tasks which are mentioned in Table 1 and its trilogy $(t_{exec}(\tau_a), e_c(\tau_a), p_{\tau_a}(N_i^k))$. As a result, the instanteneous energy consumption by each node N_i^k in the time interval $[t_1, t_2]$ is $EC_{dt}(N_i^k)$ which is expressed in watt and formalized as:

$$EC_{dt}(N_i^k) = \int_{t_1}^{t_2} \sum_{a=1}^{Nb_\tau^{N_i^k}} (p_{\tau_a}(N_i^k) \times e_c(\tau_a))dt + \varepsilon \quad (2)$$

such that $i \in [1..Nb_N^k]$ and $k \in [1..Nb_Z]$

We assume also that we can predict the approximate value of the instanteneous energy production $E_{prod}(t_1, t_2)$ by the additional battery in each node $B_{pr}(N_i^k)$ for a near future in the time interval $[t_1, t_2]$ depending on the following formula:

$$E_{prod}(t_1, t_2) = \int_{t_1}^{t_2} \sum_{t_i}^{t_j} [e_{prod} \times (t_j - t_i)] \quad (3)$$

where e_{prod} is the produced energy in each time unit, $t_1 \leq t_i \leq t_2$ and $t_1 \leq t_j \leq t_2$.

2.3 Energy Problem

We assume that the energy consumption times can interfere with energy production times. We assume also that the harvesting energy can not be available many times according to its special characteristics.

We consider that in the interval time $[t_3, t_4]$ the harvesting energy is not available, while a set of nodes execute the most energy-consuming tasks which are mentioned in table 1. Therefore, the energy produced by nodes is negligeable (4) and the frequency of energy consumption is rising.

$$E_{prod}(t_3, t_4) \approx 0 \tag{4}$$

Then, the charge of nodes will reach the β threshold (5).

$$C(N_i^k) = \beta \tag{5}$$

Thus, deactivating a number of nodes results in an increase in energy consumption by the remaining active nodes due to the increase in the distance between them. Therefore, the number of active nodes in the network will decrease in the event of a lack of harvesting energy. Indeed, with the absence of harvesting energy, the network can stop working until harvesting energy returns or human intervention which is unpleasant in RWSNs and should be avoided.

3 CONTRIBUTION: A NEW PROTOCOL FOR MINIMIZING ENERGY CONSUMPTION

We present in this section the paper's contribution. We start with a motivation that defines the RWSNs challenges and cites the proposed solutions. Then, we cite a set of 3D mobility and resizing rules. Finally, we describe the different algorithms used in the paper's contribution.

3.1 Motivation

The major challenge in RWSNs is to keep the network alive as long as possible by the minimization of energy consumption by sensor nodes until the execution of reconfiguration scenarios. In this paper, we try to increase the execution time in RWSNs by using a new methodology based on a multi-agent architecture which allows decreasing the energy consumption in the network using: *i)* 3D mobility which means the change of position of a mobile node or a mobile sink node to minimize the distance between nodes to profit an amount of energy, *ii)* resizing of zones is the merge of neighboring zones which have a reduced number of active nodes, and *iii)* mobile sink nodes that consist of specific mobile nodes equipped with high charge level batteries. We present in this part the global equation system which permits to decrease the energy consumption.

3.2 Formalization

We propose a set of rules to tune the application of 3D mobility of mobile entities and resizing of zones. According to the 3D mobility rules (**Rule 1** and **Rule 2**), we can minimize the energy consumption by sensor nodes in the network by minimizing the total distance between the network entities. According to the resizing rule (**Rule 3**), we can extend the life of the remaining active nodes and ensure comprehensive coverage in the network. Table 2 resumes the definition of thresholds and variables used in these rules. Indeed, we have three rules:

Rule 1: if $C(N_i^k) \leq \alpha$ *and* $C(N_i^k) \geq \beta$ *and* $Nb_{Act}(Z_k) > \gamma$ then zone agent Ag_k applies 3D mobility of a mobile sink node SN_m^k according to the following sub-rules:

Sub-rule 1.1: if SN_m^k is free, i.e., it has not moved recently as close to a node N_a^k such that $C(N_a^k) \leq \alpha$, the following tasks will be applied in Z_k:

$$\begin{cases} \tau_{j_1}^{Ag_k} = Apply3DMobility() \\ \tau_{j_2}^{SN_m^k} = MovingTo(x', y', z') \end{cases}$$

such that $j_1 \in [1..Nb_\tau^{Ag_k}]$, $j_2 \in [1..Nb_\tau^{SN_m^k}]$, $k \in [1..Nb_Z]$ and $m \in [1..Nb_{SN}^k]$.

where $\tau_{j_1}^{Ag_k}$ is the task number j_1 executed by the zone agent Ag_k. By executing this task, Ag_k will be apply the 3D mobility in Z_k. $\tau_{j_2}^{SN_m^k}$ is the task number j_2 executed by sink node SN_m^k. By executing this task, SN_m^k will be moved to another position x', y', z'.

Sub-rule 1.2: if SN_m^k is not free, i.e., it has moved recently as close to a node N_a^k such that $C(N_i^k) \leq \alpha$, zone agent Ag_k has two cases:

Case 1: if $C(N_a^k) \leq \beta$ then the following task will be applied:

$$\tau_j^{Ag_a^k} = Deactivate(N_a^k)$$

such that $k \in [1..Nb_Z]$, $j \in [1..Nb_\tau^{Ag_k}]$ and $a \in [1..Nb_N^k]$.

where $\tau_j^{Ag_a^k}$ is the task number j executed by a node agent Ag_a^k. By executing this task, Ag_a^k will deactivate the node N_a^k.

Table 2: Thresholds and variables definition.

	Definition
α	Node charge threshold. When $C(N_i^k) \leq \alpha$ zone agent Ag_k decides to apply 3D mobility of mobile sink nodes or of mobile nodes in some cases.
β	Node charge threshold. When $C(N_i^k) \leq \beta$, the node agent Ag_i^k decides to deactivate the node N_i^k temporarily in some situations.
$Nb_{Act}(Z_k)$	Number of active nodes in the zone Z_k.
γ	Number of active nodes in zone Z_k threshold. When $Nb_{Act}(Z_k) \leq \gamma$ zone agent Ag_k decides to apply 3D mobility of mobile nodes or of mobile sink nodes in some situations.
λ	Number of active nodes in zone Z_k threshold. When $Nb_{Act}(Z_k) \leq \lambda$ controller agent Ag_{Ctrl} decides to apply resizing of the zones.

Therefore, SN_m^k will be free, zone agent Ag_k will be return to **Sub-rule 1.1**.

Case 2: if $C(N_a^k) \geq \beta$ then zone agent Ag_k will not apply 3D mobility in this case, and node N_i^k will be complete his work at the same pace.

Rule 2: if $C(N_i^k) \leq \alpha$ *and* $C(N_i^k) \geq \beta$ *and* $Nb_{Act}(Z_k) \leq \gamma$ then the zone agent Ag_k applies 3D mobility of a mobile sink node SN_m^k or of mobile node according to the following sub rules:

Sub-rule 2.1: if $|N_i^k SN_m^k| \leq |N_i^k E|$ where $E \in SN_m^k, N_b^k$ is the successor of N_i^k, $|N_i^k SN_m^k|$ is the distance between node N_i^k and sink node SN_m^k and $|N_i^k E|$ is the distance between node N_i^k and entity E. In this situation zone agent Ag_k has two cases:

Case 1: if SN_m^k is free, zone agent Ag_k will be return to **Sub-rule 1.1** without considering the conditions.

Case 2: if SN_m^k is not free, zone agent Ag_k will be pass to **Sub-rule 2.2** without considering the conditions.

Sub-rule 2.2: if $|N_i^k SN_m^k| \geq |N_i^k N_b^k|$ where N_b^k is the successor of N_i^k, $|N_i^k SN_m^k|$ is the distance between node N_i^k and sink node SN_m^k and $|N_i^k N_b^k|$ is the distance between node N_i^k and node N_b^k. In this situation the zone agent Ag_k has two cases:

Case 1: if $N_b^k \in S_{MN}$ then zone agent Ag_k sends an order to node agent Ag_i^k to ask N_b^k to getting close to N_i^k as possible without consuming a large amount of energy.

Therefore the following tasks will be executed:

$$\begin{cases} \tau_{j_1}^{Ag_k} = Apply3DMobility() \\ \tau_{j_2}^{N_b^k} = MovingTo(x", y", z") \end{cases}$$

such that $j_1 \in [1..Nb_\tau^{Ag_k}]$, $j_2 \in [1..Nb_\tau^{N_b^k}]$, $k \in [1..Nb_Z]$ and $b \in [1..Nb_N^k]$.
where $\tau_{j_1}^{Ag_k}$ is the task number j_1 executed by the zone agent Ag_k. By executing this task, Ag_k will be apply the 3D mobility in Z_k. $\tau_{j_2}^{N_b^k}$ is the task number j_2 executed by the mobile node N_b^k. By executing this task, N_b^k will be moved to another position $x", y", z"$.

Case 2: if $N_b^k \in S_{FN}$ then zone agent Ag_k will be return to **Rule 1** without considering the conditions.

Rule 3: if $Nb_{Act}(Z_k) \leq \lambda$ the controller agent Ag_{Ctrl} decides to apply the resizing of zones and it chooses the neighbor zone of Z_k with the minimum number of active nodes. The following task will be executed:

$$\tau_j^{Ag_{Ctrl}} = ApplyResizing()$$

such that $j \in [1..Nb_\tau^{Ag_{Ctrl}}]$.
where $\tau_j^{Ag_{Ctrl}}$ is the task number j executed by the controller agent Ag_{Ctrl}. By executing this task, Ag_{Ctrl} will be apply the resizing of zones and some changes will be carried out in R such as the number of active zones, number of nodes and sink nodes in zones affected by resizing.

Figure 1 presents the logic of the 3D mobility rules which is defined previously (**Rule 1** and **Rule 2**). These rules will be implemented in the Algorithm 1.

3.3 Modeling

In order to figure out the architecture of reconfigurable wireless sensor networks with their components and relations between them, we construct a class diagram which is shown in Figure 2. The proposed class diagram is composed of 12 classes:

- *Entity:* is the class model of all physical entities in RWSN. It contains the common properties (Ex: the entity state and coordinates properties) and operations (Ex: *activate()* and *deactivate()* operations) between these entities.
- *AgCtrl:* is a class which models the controller agent Ag_{Ctrl}. It extends from the entity class and

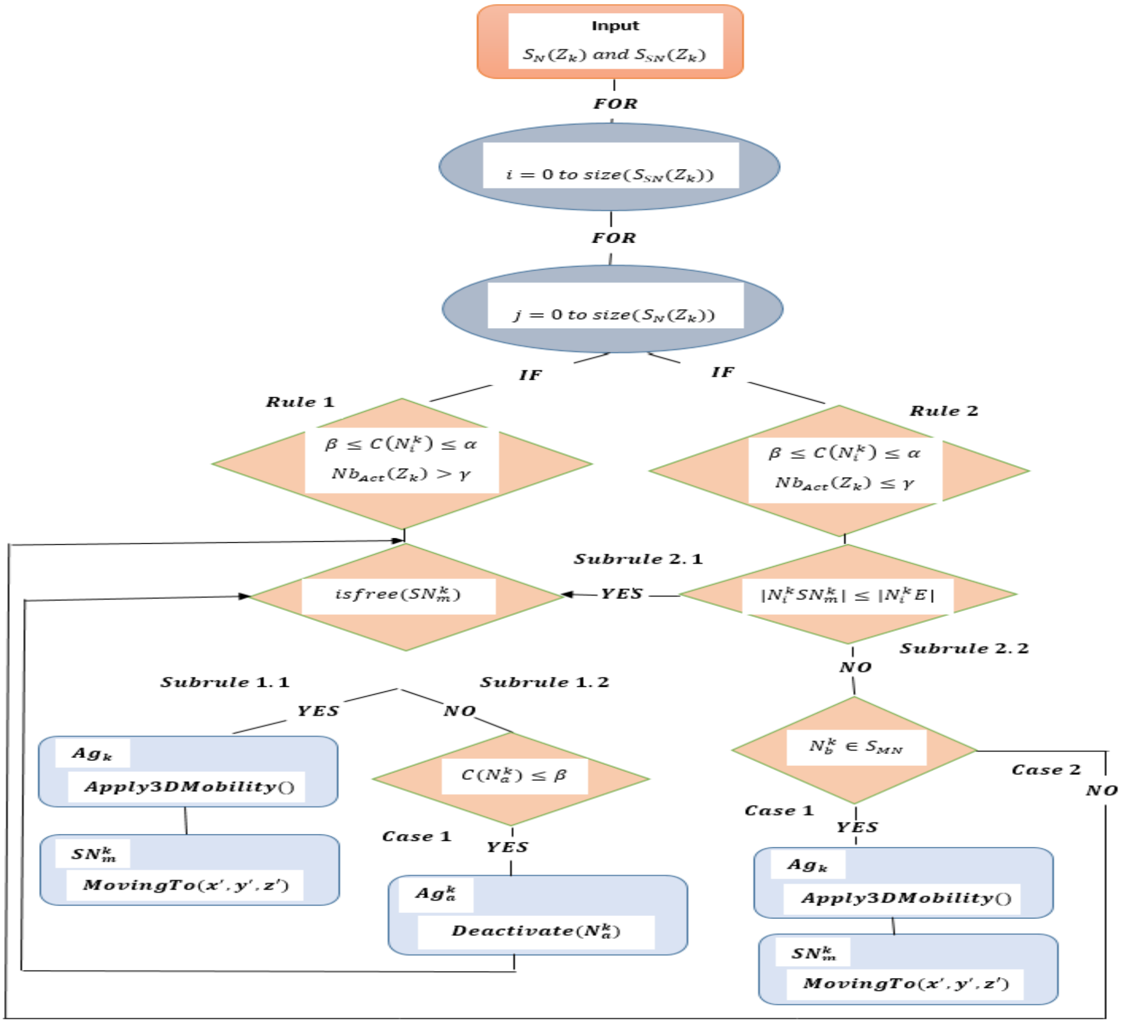

Figure 1: Logic of 3D mobility rules.

contains the Ag_{Ctrl} principal properties (Ex: a vector of zones *VZ* and a vector of captured values *VCV*) and operations (Ex: applying the resizing task *resizing()* and sending an order to a zone agent *sendOrderTo(Ag: ZoneAgent)*)

- *Zone:* is a class which models each zone Z_k. It contains the Z_k properties (Ex: an identifiant *IDj*, a state of Z_k *state* and a vector of neighbor zones *VNeighbors*)
- *ZoneAgent:* is a class which models each zone agent Ag_k. It extends from the entity class and contains the Ag_k principal properties (Ex: an identifiant *IDj* and a vector of nodes *VN*) and operations (Ex: organizing the nodes in Z_k *organizeNodes()* and applying the 3D mobility task *apply3DMobility()*).
- *SubZone:* is a class which models each subzone SZ_m^k in Z_k. It contains the SZ_m^k principal properties (Ex: an identifiant composed of two properties (*IDj* and *IDk*) and a vector of nodes *VN*).
- *Sink:* is a class which models each mobile sink node SN_m^k in SZ_m^k. It extends from the entity class and contains the SN_m^k properties (Ex: an identifiant composed of two properties (*IDj* and *IDk*) and a boolean property *free*) and operations (Ex: moving to another position *movTo(x2: Integer, y2: Integer, z2: Integer)* and sending an order to a node N_i^k *sendOrderTo(N: Node)*).
- *Node:* is a class which models each node N_i^k in Z_k. It extends from the entity class and contains the N_i^k principal properties (Ex: an identifiant composed of two properties (*IDi* and *IDj*) and a vector of sensors *VSens*) and operations (Ex: monitoring a physical or chemical values in the environment *monitoring()* and sending a vector of captured values to its successor *sendVCVTo(VCV: CapVal, E:*

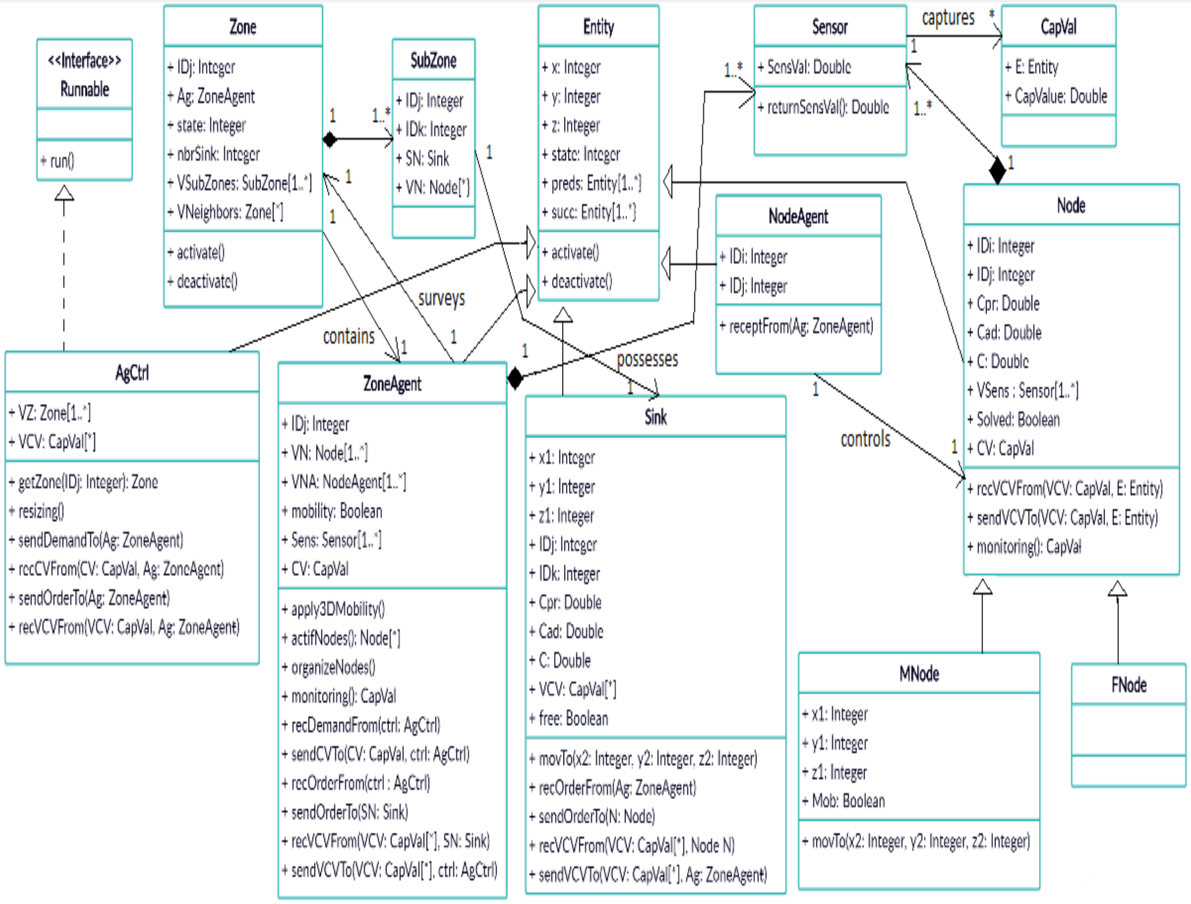

Figure 2: Class diagram for RWSNs.

Entity)).

- *NodeAgent:* is a class which models each node agent Ag_i^k in Z_k. It extends from the entity class and contains the Ag_i^k principal properties (Ex: an identifiant composed of two properties (*IDi* and *IDj*)) and operation (receiving an order from zone agent Ag_k *receptFrom(Ag: ZoneAgent)*).
- *FNode:* is a class which models each fixed node $N_a^k \in S_{FN}$ in Z_k. It extends from the node class.
- *MNode:* is a class which models each mobile node $N_b^k \in S_{MN}$ in Z_k. It extends from the node class and contains the N_b^k properties (Ex: a boolean property *Mob*) and an operation (moving to another position *movTo(x2: Integer, y2: Integer, z2: Integer)*).
- *Sensor:* is a class which models each sensor $Sens_j^{N_i^k}$ in a node N_i^k or $Sens_j^{Ag_k}$ in a zone agent Ag_k. It extends from the entity class and contains the $Sens_j^{N_i^k}$ property (a captured value *SensVal*) and operation (return the captured value *returnSensValue(): Double*).
- *CapVal:* is a class which models each captured values *CV* that captured by a sensor $Sens_j^{N_i^k}$ or $Sens_j^{Ag_k}$.

3.4 Algorithms

In the proposed solutions we use a communication protocol based on a multi-agent architecture and a wireless communication between different network entities. In order to minimize the energy consumption in **R**, each zone agent Ag_k executes **Algorithm 1** to apply the 3D mobility according to its rules that mentioned in 3.2. While **Algorithm 2** implements the most used function in **Algorithm 1**, i.e., function $move(e, n, succ)$ that moves the entity $e \in \{S_{MN}, S_{SN}\}$ as close to node n taking into consideration the distance between it and its successor *succ*. Otherwise controller agent Ag_{Ctrl} executes **Algorithm 3** to apply the resizing task according to its rules that montioned in 3.2.

4 EXPERIMENTATION

In order to make a system reliable against fires in a forest, we build a reconfigurable wireless sensor network R.

Algorithm 1: Apply 3D Mobility in Z_k.

```
Require: Set of nodes and sink nodes
Ensure: Minimize the total distance between node
  $vsz \leftarrow getsubzones()$
  for $i = 0$ to $size(vsz)$ do
    variables affectation
    for $j = 0$ to $size(get(vn)$ do
      variables affectation
      if Rule 1 conditions then
        if Sub-rule 1.1 conditions then
          $move(s,n,succ)$
        else if Sub-rule 1.2 conditions then
      for $k = 0$ to $size(getpred(s))$ do
        $e \leftarrow get(getpred(s),k)$
        if $e \in S_N$ then
          if $getc(e) \leq \alpha \& issolved(e)$ then
            if $getc(e) \leq \beta$ then
              $deact(getag(get(n,i),get(n,k)))$
              $move(s,n,succ)$
            end if
          end if
        end if
      end for
      if Rule 2 conditions then
        if Sub-rule 2.1 conditions then
          if case 1 condition then
            $move(s,n,succ)$
          if case 2 condition then
      for $k = 0$ to $size(getpred(s))$ do
        $e1 \leftarrow get(getpred(s),k)$
          if Sub-rule 2.1 conditions then
            $deact(getag(get(n,i),get(n,k)))$
            $move(s,n,succ)$
            if case 1 conditions then
        if $getc(succ) \geq \alpha$ then
          $move(succ,n,succ)$
        end if
      end for
      if Sub-rule 2.2 conditions then
        $move(succ,n,succ)$
      end if
    end for
  end for
```

Algorithm 2: Moving node e as close to n.

```
Require: Entity $e \in \{S_{MN}, S_{SN}\}$, node $n$ and its successor $succ$
Ensure: moving $e$ as close to $n$
  $movingto(e, (getx(n) + getx(succ))/2, (gety(n) + gety(succ))/2, (getz(n) + getz(succ)/2))$
  if $e \in S_{SN}$ then
    $setfree(e, false)$
  end if
  $setsolved(n, true)$
```

Algorithm 3: Resizing of zones in R.

```
Require: Set of active zones
Ensure: Cover the possible largest space in $R$
  for $i = 0$ to $nbz$ do
    if $getnbact(get(vz,i)) \leq \lambda$ then
      $Z \leftarrow get(vz,i)$
      if $size(vneighbors(Z)) \geq 0$ then
        $Z_1 \leftarrow get(vneighbors, 0)$
        for $j = 0$ to $size(vneighbors(Z))$ do
          $X \leftarrow get(vz,i)$
          $Y \leftarrow vneighbors(X)$
          $nghbr \leftarrow get(Y, j)$
          $nbact \leftarrow getnbact(nghbr)$
          if $nbact \leq getnbact(Z_1)$ then
            $Z_1 \leftarrow nghbr$
          end if
        end for
      end if
      if $notempty(Z) \& notempty(Z_1)$ then
        $deact(Z)$
        $s \leftarrow size(vnactif(Z))$
        for $k = 0$ to $s$ do
          $add(vn(Z_1), get(vnactif(Z),k))$
          $add(vnactif(Z_1), get(vnactif(Z),k))$
        end for
        $resize \leftarrow true$
      end if
    end if
  end for
```

4.1 Case Study

4.1.1 Description

In this case study, we take a small part P of R composed of 4 zones. Each zone is composed of *i)* 40 nodes: 25 fixed nodes and 15 mobile nodes, *ii)* 3 mobile sink nodes and iii) one zone agent. Each node and zone agent in P has 2 sensors, a temperature value sensor, and a CO2 value sensor. Each node has a capture field represented as a cub (11m × 11m × 11m)

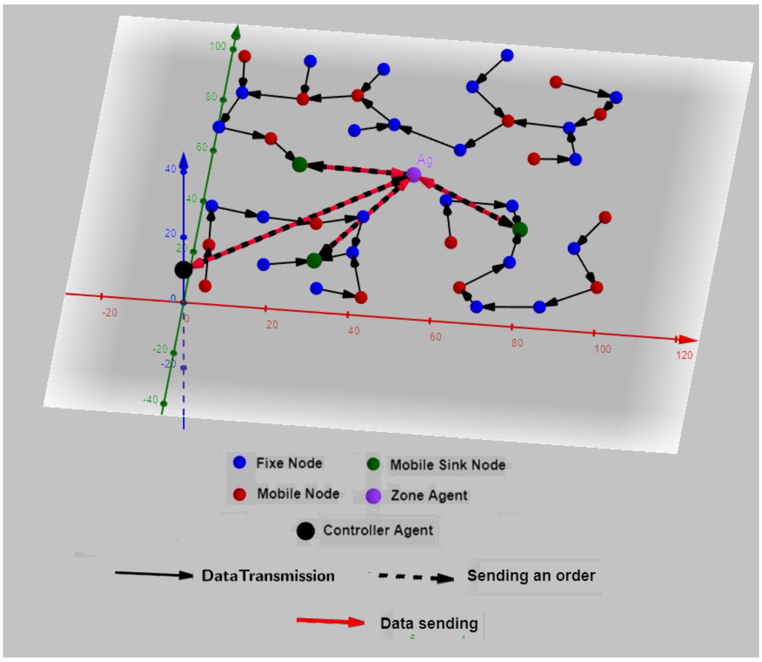

Figure 3: The entities position in zone Z_1 in 3D space.

and each zone agent has a capture field represented as a cub (22m × 22m × 22m). In order to find out the impact of the proposed solutions in RWSNs, we focused on zone Z_1. We propose that the position of controller agent is $Ag_{Ctrl}(0,0,10)$ and the position of zone agent Ag_1 is $(50,50,5)$. Figure 3 illustrates the initial nodes and mobile sink nodes positions in Z_1 in 3D Graphic.

Table 3: The number of controller zone demands per day in each season.

Months	Times/Day	Hours
Jan-Feb-Mar Oct-Nov-Dec	2	$12:00^h \& 14:00^h$
Apr-May-Jun Jul-Aug-Sep	6	$11:00^h \& 12:00^h$ $13:00^h \& 14:00^h$ $15:00^h \& 16:00^h$

4.1.2 Reconfiguration Scenarios

We propose the following reconfiguration scenarios which can be executed by the network entities:

1. Each period ρ, controller agent Ag_{Ctrl} executes a periodic task $\tau_1^{Ag_{Ctrl}}$ which Ag_{Ctrl} sends a request to each zone agent Ag_k to sense the temperature and CO_2 values:

$$\tau_1^{AgCtrl} = SendRequestTo(Ag_k)$$

such that $k \in [1..Nb_Z]$.

2. Each zone agent Ag_k which recepts a message from Ag_{Ctrl} will execute the following tasks:

$$\begin{cases} \tau_1^{Ag_k} = ReceptMsgFrom(AgCtrl) \\ \tau_2^{Ag_k} = CaptureCO2() \\ \tau_3^{Ag_k} = CaptureTemp() \\ \tau_4^{Ag_k} = SendCapValTo(Ag_{Ctrl}) \end{cases}$$

such that $k \in [1..Nb_Z]$.

3. Controller agent Ag_{Ctrl} will execute the following task:

$$\tau_2^{AgCtrl} = ReceptCapValFrom(CV, Ag_k)$$

such that $k \in [1..Nb_Z]$ and CV is the captured values.

4. Controller agent Ag_{Ctrl} tests if the incoming captured values from each zone agent Ag_k reach the thresholds.

4.a. if the incoming captured values from zone agent Ag_{k_1} reach the thresholds Ag_{Ctrl} will execute the following task:

$$\tau_3^{AgCtrl} = SendOrderTo(Ag_{k_1})$$

such that $k_1 \in [1..Nb_Z]$.

4.b. if the incoming captured values from all zone agents not reach the thresholds, Ag_{Ctrl} will wait for the next periodic task τ_1^{AgCtrl}.

5. Each zone agent Ag_k which received an order from controller agent Ag_{Ctrl} will execute these tasks:

$$\begin{cases} \tau_5^{Ag_k} = ReceptOrderFrom(Ag_{Ctrl}) \\ \tau_6^{Ag_k} = OrganizeNodes() \\ \tau_7^{Ag_k} = Apply3DMobility() \\ \tau_8^{Ag_k} = ApplyResizing() \\ \tau_9^{Ag_k} = SendOrderTo(SN_m^k) \end{cases}$$

such that $k \in [1..Nb_Z]$.

6. Each mobile sink node SN_m^k which received an order from zone agent Ag_k will start collecting of the captured values from SNs in the same subzone SZ_m^k. It will execute these tasks:

$$\begin{cases} \tau_1^{SN_m^k} = ReceptOrderFrom(Ag_k) \\ \tau_2^{SN_m^k} = SendOrderTo(Succ) \end{cases}$$

such that $m \in [1..Nb_{SN}^k]$, $k \in [1..Nb_Z]$ and $Succ \in S_N^k$.

7. Each node N_i^k which received a message from its predecessor will execute set of tasks $T = \{\tau_5, \tau_3, \tau_4\}$ in the same order, these tasks are described in Table 1.

8. Each mobile sink node SN_m^k which received a vector of all captured values VCV from SNs in the same subzone SZ_m^k will execute these tasks:

$$\begin{cases} \tau_3^{SN_m^k} = ReceptVCVFrom(Pred) \\ \tau_4^{SN_m^k} = SendVCVTo(Ag_k) \end{cases}$$

such that $m \in [1..Nb_{SN}^k]$, $k \in [1..Nb_Z]$ and $Pred \in S_N^k$.

9. Each zone agent which received a vector of all captured values from all sink nodes in Z_k will execute these tasks:

$$\begin{cases} \tau_{10}^{Ag_k} = ReceptVCVFrom(SN_m^k) \\ \tau_{11}^{Ag_k} = SendVCVTo(Ag_{Ctrl}) \end{cases}$$

such that $k \in [1..Nb_Z]$ and $m \in [1..Nb_{SN}^k]$.

10. Controller agent Ag_{Ctrl} which received a vector of all captured values from all zone agents in R will execute these tasks:

$$\begin{cases} \tau_4^{Ag_{Ctrl}} = ReceptVCVFrom(Ag_k) \\ \tau_5^{Ag_{Ctrl}} = Treatment(VCV) \end{cases}$$

such that $k \in [1..Nb_Z]$.

10.a if the incoming vector of captured values from zone agent Ag_k reach the thresholds, Ag_{Ctrl} will triggers an alert.

10.b if the incoming vector of captured values from zone agent Ag_k not reach the thresholds, Ag_{Ctrl} will wait for the next periodic task $\tau_1^{Ag_{Ctrl}}$.

4.2 Evaluation of Performance

In order to prove the originality of this work, we put the part P which composed of 4 zones in some special conditions. We consider that with every demand of sensory values from each zone agent Ag_k in P, it sends a value greater than the thresholds of temperature $TempThreshold = 35$ and of CO2 value $CO2Threshold = 5$. Therefore the controller agent Ag_{Ctrl} requests from Ag_k a vector of monitoring values VSV which is captured by sensor nodes. Thus, the exchange of messages abounds in each zone Z_k so, the sensor nodes will be consuming more energy. Figure 4 illustrates the change in the number of active nodes in P over time without the proposed solutions in the absence of harvesting energy.

We notice that the lifetime of the zone Z_1 is 92 days in Figure 4, this period is very short compared with the lifetime of the same zone in Figure 5 which is estimated by 229 days with a percentage equal to 248%.

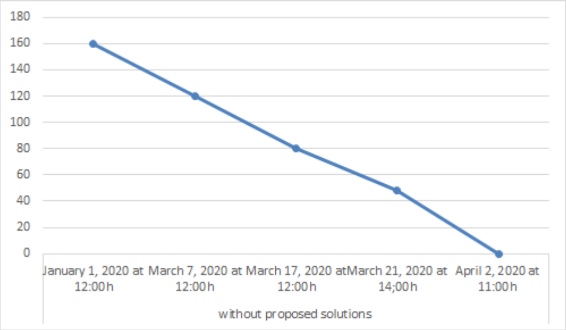

Figure 4: The number of active nodes in R over time without the proposed solutions in the absence of harvesting energy.

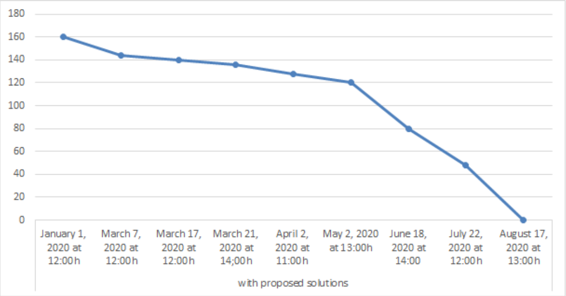

Figure 5: The number of active nodes in R over time with the proposed solutions in the absence of harvesting energy.

Table 4: Discussion.

Paper	Reconfiguration	Mobility	Resizing	Mobile sink nodes	Multi-agent architecture	Other solutions
(Mahendrababu and Joshitha, 2014) (Anastasi et al., 2009)	✗	✗	✗	✗	✗	✓
(Gasmi et al., 2016)	✓	✗	✗	✗	✗	✓
(Grichi et al., 2014b)	✓	✗	✗	✗	✓	✗
(Housseyni et al., 2017) (Chniter et al., 2018)	✓	✗	✗	✗	✓	✗
(Grichi et al., 2017)	✓	✓	✓	✗	✓	✗
(Son and Ahn, 2014) (Ahn, 2009)	✗	✓	✗	✗	✗	✗
(Ji et al., 2013) (Erman et al., 2012)	✗	✗	✓	✗	✗	✗
(Thomas and Mathew, 2018) (Chao et al., 2019) (Kang et al., 2019) (Wang et al., 2019) (Zhong and Ruan, 2018)	✗	✗	✗	✓	✗	✗
This paper	✓	✓	✓	✓	✓	✗

4.3 Discussion

The originality of this paper is the combination between several solutions in order to solve the energy problem in WSNs which is mentioned in **2.3**. While the related works deal only with some solutions. The Table 4 presents the difference between this paper and the related works in terms of proposed solutions.

5 CONCLUSION

This paper proposes a new solution to energy problems in RWSNs which works under harvesting energy. This solution uses the 3D mobility, resizing and mobile sink nodes based on a multi-agent architecture which composed of a set of agents with different levels of responsibilities. The multi-agent architecture allows executing the reconfiguration scenarios more optimally. We also develop an application that permits the construction of RWSN using the proposed solution, and explains its effectiveness by comparing the results with and without applying this solution.

The perspectives of this paper is to increasing and improving the effectiveness of solutions while reducing cost and find an optimal path for the mobile nodes and sink nodes more useful to the network in terms of profit time and cost. Finally, we seek to take into consideration the constraints of energy, real-time and coverage area for each entity in RWSNs.

REFERENCES

Ahn, H.-S. (2009). Formation coordination for self-mobile localization: Framework. In *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation-(CIRA)*, pages 340–348. IEEE.

Anastasi, G., Conti, M., Francesco, M., and Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7:537–568.

Ben Salem, M. O., Mosbahi, O., Khalgui, M., Jlalia, Z., Frey, G., and Smida, M. (2016). Brometh: Methodology to design safe reconfigurable medical robotic systems. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 13.

Chao, F., He, Z., Pang, A., Zhou, H., and Ge, J. (2019). Path optimization of mobile sink node in wireless sensor network water monitoring system. *Complexity*, 2019:1–10.

Chen, Y., Mao, X., Hou, F., Wang, Q., and Yang, S. (2016). Combining re-allocating and re-scheduling for dynamic multi-robot task allocation. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 000395–000400. IEEE.

Chniter, H., Li, Y., Khalgui, M., Koubaa, A., Li, Z., and Jarray, F. (2018). Multi-agent adaptive architecture for flexible distributed real-time systems. *IEEE Access*, PP:1–1.

Erman, A. T., Dilo, A., and Havinga, P. (2012). A virtual infrastructure based on honeycomb tessellation for data dissemination in multi-sink mobile wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):17.

Fang Tang and Parker, L. E. (2005). Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1501–1508.

Garcia, R., Gordon-Ross, A., and George, A. D. (2009). Exploiting partially reconfigurable fpgas for situation-based reconfiguration in wireless sensor networks. In *2009 17th IEEE Symposium on Field Programmable Custom Computing Machines*, pages 243–246.

Gasmi, M., Mosbahi, O., Khalgui, M., Gomes, L., and Li, Z. (2016). R-node: New pipelined approach for an effective reconfigurable wireless sensor node. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP:1–14.

Gharsellaoui, H., Gharbi, A., Khalgui, M., and Ahmed, S. (2012). *Feasible Automatic Reconfigurations of Real-Time OS Tasks.*

Goldsmith, A. (2005). *Wireless Communications*. Cambridge University Press.

Grichi, H., Mosbahi, O., and Khalgui, M. (2014a). New flexible architectures for reconfigurable wireless sensor networks. In *International Conference on Software Technologies*, pages 151–169. Springer.

Grichi, H., Mosbahi, O., and Khalgui, M. (2014b). Reconfigurable wireless sensor networks new adaptive dynamic solutions for flexible architectures. In *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, pages 254–265. IEEE.

Grichi, H., Mosbahi, O., Khalgui, M., and Li, Z. (2016). Rwin: New methodology for the development of reconfigurable wsn. *IEEE Transactions on Automation Science and Engineering*, PP:1–17.

Grichi, H., Mosbahi, O., Khalgui, M., and Li, Z. (2017). New power-oriented methodology for dynamic resizing and mobility of reconfigurable wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, PP:1–11.

Habibu, H., Zungeru, A. M., Susan, A. A., and Gerald, I. (2014). Energy harvesting wireless sensor networks: Design and modeling. *Int. J. Wireless Mobile Netw.*, 6(5):17–31.

Housseyni, W., Mosbahi, O., Khalgui, M., Li, Z., and Yin, L. (2017). Multiagent architecture for distributed adaptive scheduling of reconfigurable real-time tasks with energy harvesting constraints. *IEEE Access*, PP:1–1.

Ji, S., Beyah, R., and Cai, Z. (2013). Snapshot and continuous data collection in probabilistic wireless sensor networks. *IEEE Transactions on Mobile Computing*, 13(3):626–637.

Kang, M.-J., Yoon, I., and Noh, D. K. (2019). Efficient location service for a mobile sink in solar-powered wireless sensor networks. In *Sensors*.

Mahendrababu, K. and Joshitha, K. L. (2014). A solution to energy hole problem in wireless sensor networks using witricity. In *International Conference on Information Communication and Embedded Systems (ICICES2014)*, pages 1–6.

Messer, H., Zinevich, A., and Alpert, P. (2006). Environmental monitoring by wireless communication networks. *Science*, 312(5774):713–713.

Rault, T., Bouabdallah, A., and Challal, Y. (2014). Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*, 67:104 – 122.

Son, J.-H. and Ahn, H.-S. (2014). Formation coordination for the propagation of a group of mobile agents via self-mobile localization. *IEEE Systems Journal*, 9(4):1285–1298.

Thomas, S. and Mathew, T. (2018). Intelligent path discovery for a mobile sink in wireless sensor network. *Procedia Computer Science*, 143:749 – 756. 8th International Conference on Advances in Computing & Communications (ICACC-2018).

Wang, J., Gao, Y., Liu, W., Kumar, A., and Kim, H.-J. (2019). Energy efficient routing algorithm with mobile sink support for wireless sensor networks. *Sensors*, 19:1494.

Wang, X., Khalgui, M., and Li, Z. (2011). Dynamic low power reconfigurations of real-time embedded systems. pages 415–420.

Zhang, J., Khalgui, M., Boussahel, W., Frey, G., Hon, C. T., Wu, N., and Li, Z. (2015). Modeling and verification of reconfigurable and energy-efficient manufacturing systems. *Discrete Dynamics in Nature and Society*, 2015:1–14.

Zhong, P. and Ruan, F. (2018). An energy efficient multiple mobile sinks based routing algorithm for wireless sensor networks. *IOP Conference Series: Materials Science and Engineering*, 323:012029.