New Energy Efficient and Fault Tolerant Methodology based on a Multi-agent Architecture in Reconfigurable Wireless Sensor Networks

Hanene Rouainia[1][a], Hanen Grichi[2,4][b], Laid Kahloul[3][c] and Mohamed Khalgui[4,5][d]

[1]*Faculty of Sciences of Tunis, El-Manar University, Tunis, Tunisia*

[2]*Faculty of Sciences of Bizerte (FSB) - University of Carthage, Bizerte, Tunisia*

[3]*LINFI Laboratory, Computer Science Department, Biskra University, Biskra, Algeria*

[4]*School of Electrical and Information Engineering, Jinan University, Zhuhai, China*

[5]*INSAT Institute, University of Carthage, Tunis, Tunisia*

hanene.rouainia@fst.utm.tn, {hanen.grichi, laid.k.b, khalgui.mohamed}@gmail.com

Keywords: Reconfigurable Wireless Sensor Networks, Multi-agent Architecture, Mobility, Resizing, Mobile Sink Node, Hardware/Software Failures, Test Packet Technique, Harvesting Energy.

Abstract: Reconfigurable wireless sensor networks became more complex and dynamic systems. Their importance increases with time and more challenges appear. The most important challenges in RWSNs are the energy and software/hardware failure problems. In this paper, we propose a new methodology composed of a set of solutions summarized in the application of the mobility, resizing, and mobile sink nodes using a multi-agent architecture and an energy efficient routing protocol. It contains also a test packet technique to detect the malfunctioning entities and isolate them. Moreover, we develop a simulator named *RWSNSim* which allows simulating WSNs and RWSNs with and without application of the proposed methodology. It permits also to compare the different results using line charts. Finally, we simulate a case study with *RWSNSim* in a 3D environment to evaluate the performance of the proposed methodology.

1 INTRODUCTION

Wireless sensor networking is an important wireless technology that has a wide variety of applications from small-size to large-scale systems and provides unlimited future potentials. A wireless sensor network (WSN) is composed of a set of very small battery-operated devices (SNs). It can be used in many areas such as medical, military, environmental monitoring, forecasting, and intelligent home (Vijayalakshmi and Muruganand, 2018). Sensor nodes (SNs) are the principal devices in WSNs. They communicate with each other through wireless communication to transmit a volume of data to a central station and to execute a set of tasks that may involve data processing (Agrawal, 2017).

In WSNs, we have several challenges such as lack of energy, real-time and hardware/software failures (Hafidi et al., 2020), (Allouch et al., 2019). These problems occur because of several factors such as WSNs work under many types of renewable energy resources which are not frequently available, SNs use limited energy resources, communication volume, SNs are fragile and prone to failures, harsh environmental conditions, and human effect (Ghribi et al., 2018), (Rouainia et al., 2020). Many solutions are proposed to resolve these problems such as reconfiguration, mobile sink nodes, mobility, resizing, energy efficient routing protocols, and fault tolerant routing protocols (Salem, 2017), (Housseyni et al., 2018). A reconfigurable wireless sensor network (RWSN) is a WSN allowing reconfiguration scenarios (Grichi et al., 2018). In WSNs, all sensor nodes are fixed. However, sensor nodes in RWSNs can be fixed or mobile (Rouainia et al., 2020).

The originality of this paper lies in the use of a set of solutions in a unified methodology according to a set of rules in energy and time-saving manner. These solutions are effective and inexpensive in terms of time and cost such as reconfiguration, mobile sink nodes, mobility, and resizing. Each solution has proven its effectiveness in previous related works

[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

(Rouainia et al., 2020), (Grichi et al., 2018). We also added a test packet technique to treat the hardware/software failures. It has proven its effectiveness in detecting failures and treating them as quickly as possible. Besides the previous novelties in this paper, we provide also a new tool called *RWSNSim*. It allows simulating our case studies using the proposed methodology applying the two well-known routing protocols: LEACH and WBM-TEEN (Zagrouba and Kardi, 2021). Our developed simulator provides for the use of a comprehensive and accurate view of the proposed methodology's impact on the network, which is estimated by a 625% increase in network lifetime and the failures are discovered and treated when they occur.

This paper is structured as below. After the introduction section, we present the related works in section 2. Then, we resume the background of WSNs, RWSNs, and the energy issue in Section 3. A new energy efficient and fault tolerant methodology in RWSNs is reported in Section 4. Section 5 presents the experiment simulated using *RWSNSim* to validate and evaluate the performance of the proposed methodology. Finally, the conclusion is drawn in section 6.

2 RELATED WORKS

The authors of (Housseyni et al., 2018) propose an intelligent multi-agent distributed architecture using the three forms of reconfiguration (software, hardware, and protocol reconfiguration). Moreover, in (Grichi et al., 2018), the reconfiguration is considered as an efficient solution to the energy problem in WSNs because it makes the WSN satisfy the real-time and energy constraints taking into consideration the system performance optimization.

The paper (Chao et al., 2019) considers the unmanned aerial vehicles (UAVs) as mobile sink nodes in WSN water monitoring. It proposes a mobile data collection scheme based on the high maneuverability of UAVs. While the paper (Wang et al., 2019) presents an energy efficient routing schema combined with clustering and sink mobility technologies in WSNs. Finally, the paper (Zhong and Ruan, 2018) studies the energy efficient routing method with support for multiple mobile sink nodes to effectively alleviate the hot spot problem.

The papers (Raj et al., 2021) and (Nguyen and Nguyen, 2020) use the mobility to minimize the total distance between sensor nodes in the network and consequently decrease the energy consumption to keep the network alive as long as possible. Otherwise, the works in (Ji et al., 2014) and (Erman et al., 2012) take the geographic resizing of zones as a solution to energy and coverage problems in WSNs. While the paper (Grichi et al., 2018) proposes 3D mobility and dynamic resizing of zones into a new run-time power-oriented methodology to reduce the energy consumption by sensor nodes.

Several researches suggest energy efficient routing protocols as effective solutions to the energy problem in WSNs. Indeed, we have many energy efficient routing protocols such as e-NL BEENISH, IQAR, MBC, and WBM-TEEN (Shalini and Vasudevan, 2021), (Khediri et al., 2021). Many fault tolerant protocols are also proposed to treat the software/hardware failures such as HDMRP, PFTP, and FTCP-MWSN (Moussa et al., 2020), (Zagrouba and Kardi, 2021).

3 BACKGROUND

We present in this section a semi-formal description of WSNs and RWSNs components and architectures. We also describe the energy issue detailing the energy model and problem. Finally, we discuss and detail the software/hardware problems in WSNs.

3.1 Wireless Sensor Networks

We consider that W is a wireless sensor network in a 3D environment. It contains a base station BS and a set of zones $S_Z(W) = \{ \bigcup_{k=1}^{Nb_Z(W)} \{Z_k\}\}$, where Z_k is a zone in W and $Nb_Z(W)$ is the total number of zones in W. Each zone Z_k contains a gateway $\{G_k, k \in [1..Nb_Z]\}$ and a set of fixed sensor nodes formalized by $S_N(Z_k) = \{ \bigcup_{i=1}^{Nb_N(Z_k)} \{N_{i,k}\}\}$, where $N_{i,k}$ is a node in Z_k and $Nb_N(Z_k)$ is the total number of nodes in Z_k.

Each sensor node has a sensing unit contains a set of sensors formalized by $S_{Sens}^{N_{i,k}} = \{Sens_{j,N_{i,k}} | \ i \in [1..Nb_N(Z_k)], \ k \in [1..Nb_Z(W)] \ and \ j \in [1..Nb_{Sens}(N_{i,k})]\}$, where $Sens_{j,N_{i,k}}$ is a sensor in $N_{i,k}$ and $Nb_{Sens}(N_{i,k})$ is the total number of sensors in $N_{i,k}$. These sensors are designed for sensing the chemical and physical conditions in the surrounding environment such as temperature, gases, humidity, pressure, etc. It contains also a power unit composed of 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 (Ramasamy, 2017).

3.2 Reconfigurable Wireless Sensor Networks

We consider that R is a reconfigurable wireless sensor network in a 3D environment. Firstly, R is based on a multi-agent architecture used to handle the execution of the reconfiguration scenarios. We have four types of agents with different roles: *i)* The controller agent Ag_{Ctrl} which is designed to control the whole network. *ii)* The zone agents $\{Ag_k \mid k \in [1..NbZ]\}$ which control the zones. These two types of agents are servers with high charge level batteries allowing them to make important and crucial decisions in the network such as applying the resizing of zones, processing the sensing data, and applying the mobility. *iii)* The sink agents $\{SA_{m,k} \mid m \in [1..Nb_{SN}(Z_k)]\ k \in [1..NbZ]\}$ which control the sinks. *iv)* The node agents $\{Ag_{i,k} \mid i \in [1..NbN(Z_k)]\ k \in [1..NbZ]\}$ which control the sensor nodes. These two types of agents are software agents installed on the sensor nodes and mobile sink nodes themselves. They can make a set of decisions like activation/deactivation and mobility of sinks and mobile nodes (Grichi et al., 2018), (Rouainia et al., 2020).

Indeed, R contains a base station BS, a controller agent Ag_{Ctrl}, and a set of zones formalized by $S_Z(R) = \{\bigcup_{k=1}^{Nb_Z(R)} \{Z_k\}\}$, where Z_k is a zone in R and $Nb_Z(R)$ is the total number of zones in R. Each zone Z_k contains a zone agent Ag_k, a set of mobile sink nodes formalized by $S_{SN}(Z_k) = \{SN_{m,k} |\ k \in [1..Nb_Z]\ \ m \in [1..Nb_{SN}(Z_k)]\}$, where $SN_{m,k}$ is a mobile sink node in Z_k and $Nb_{SN}(Z_k)$ is the number of mobile sink nodes in Z_k, and a set of sensor nodes formalized by $S_N(Z_k) = \{N_{i,k} |\ k \in [1..Nb_Z]\ \ i \in [1..Nb_N(Z_k)]\}$. We have two types of sensor nodes: fixed and mobile ones. The fixed sensor nodes are formalized by $S_{FN}(Z_k) = \{N_{i,k} |\ k \in [1..Nb_Z]\ \ i \in [1..Nb_N(Z_k)]\}$. The mobile sensor nodes are formalized by $S_{MN}(Z_k) = \{N_{i,k} |\ k \in [1..Nb_Z]\ \ i \in [1..Nb_N(Z_k)]\}$. We denote that $S_{MN} \cup S_{FN} = S_N\ \ and\ \ S_{MN} \cap S_{FN} = \varnothing\}$.

Otherwise, in R the fixed sensor nodes are the same in W, but the mobile sensor nodes differ in terms of the existence of a mobilizer and a location finder. The mobilizer permits to mobile sensor node to move easily in the network. The location finder is used to identify its position (Tzounis et al., 2017), (Robinson et al., 2017). The mobile sink nodes play the role of gateways between sensor nodes and zone agents. They have the same components of mobile sensor nodes without the sensing unit and with two high-charge level batteries (Ramasamy, 2017).

Each entity in W and R has three coordinates representing its position. These coordinates are formalized by $\{(x_E, y_E, z_E) | E \in \{S_N, S_{SN}, S_G, S_{Ag}, BS\}\}$, where S_G is the set of gateways in W.

The total charge of each entity E in W and R is formalized by $C(E) = C(B_{pr}(E)) + C(B_{add}(E))$. While the charge capacity of each entity E in W and R is formalized by $capacity(E) = capacity(B_{pr}(E)) + capacity(B_{add}(E))$.

3.3 Energy Issue

In the following, we present an energy model, the most energy-consuming tasks and we formalize the energy problem in R. We define also the software and hardware failure problems in R.

3.3.1 Energy Model

We consider that in each period ρ, each entity in R can execute a set of tasks. These tasks are formalized by $T = \{\tau_1, \tau_2, ..., \tau_{Nb_\tau(E)}\} | E \in \{S_N, S_{SN}, S_{Ag}\}$, where $Nb_\tau(E)$ is the total number of tasks that can be executed by E in ρ. Each task τ_a is linked to a trilogy formalized by $Tr_{E,\rho} = (t_{exec}(\tau_a(E)), e_c(\tau_a(E)), p_{\tau_a}(E))$, where $t_{exec}(\tau_a(E))$ is the execution time of $\tau_a(E)$, $e_c(\tau_a(E))$ is the energy consumed by E to execute τ_a and $p_{\tau_a}(E)$ is a function formalized by:

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

In order to predict the approximate value of the energy produced by the additional battery in each entity E in R in a period ρ, we formalize it in the following formula:

$$EP_{[\rho]}(E) = \int_{t_x}^{t_y} \sum_{t_i}^{t_j} [e_{prod} \times (t_j - t_i)] \quad (2)$$

where e_{prod} is the produced energy in each time unit, $t_x \le t_i \le t_y$,$t_x \le t_j \le t_y$ and $\rho = |t_y - t_x|$.

Table 1 presents a set of the most energy-consuming tasks executed by different entities in R.

In order to predict the approximate value of the energy consumed by each entity E in R in a period ρ, we formalize it by:

$$EC_{[\rho]}(E) = \int_{t_x}^{t_y} \sum_{a=1}^{Nb_\tau^E} (p_{\tau_a}(E) \times e_c(\tau_a(E)))dt + \varepsilon \quad (3)$$

such that $\rho = |t_y - t_x|$, Nb_τ^E is the number of tasks executed by E in the period ρ.

Table 1: Set of the most energy-consuming tasks executed by different entities in R.

E	$\tau_a(E)$	Description
$E \in \{S_{Ag}, S_{SN}, S_N\}$	$\tau_1(E) = ReceptFrom()$	Receiving packets from predecessors.
	$\tau_2(E) = SendTo()$	Sending packets to successors.
	$\tau_3(E) = Deactivate()$	Deactivate the entity E.
	$\tau_4(E) = Activate()$	Activate the entity E.
Ag_{Ctrl}	$\tau_5(Ag_{Ctrl}) = Resizing()$	Apply the resizing of zones.
	$\tau_6(Ag_{Ctrl}) = Isolate(Ag_k)$	Isolate the zone agent Ag_k.
Ag_k	$\tau_5(Ag_k) = ApplyMobility()$	Apply the mobility of mobile entities.
	$\tau_6(Ag_k) = Isolate(E)$	Isolate the entity E where $E \in \{S_N, S_{SN}\}$.
	$\tau_7(Ag_k) = OrganizeNodes()$	Organize the sensor nodes in Z_k into clusters in subzones.
$SN_{m,k}$	$\tau_5(SN_{m,k}) = MoveTo()$	Moving to another position in Z_k.
$N_{i,k}$	$\tau_5(N_{i,k}) = SensingBy(Sens_{j,N_{i,k}})$	Sensing the physical and chemical conditions of the surrounding environment by $Sens_{j,N_{i,k}}$.
	$\tau_6(N_{i,k}) = MoveTo()$	Moving to another position where $N_{i,k} \in S_{MN}$.

3.3.2 Energy Problem

Considering that R operates under a renewable energy characterized by the oscillating presence in the surrounding environment. We assume that the energy consumption times can interfere with energy production times. However, many times the renewable energy is not available.

We consider that renewable energy is not available in the time interval $[t_a, t_b]$ which can be a long time. In the meantime, a set of entities in R still working and executing the most energy-consuming tasks mentioned in Table 1. So the energy produced in each entity is almost equal to zero (i.e., $EP_{[t_a,t_b]} \approx 0$). On the other side, the charge of entities will reach the β threshold (i.e., $C(E) \leq \beta$).

Therefore, as more time passes, more entities will be deactivated. As a result, the distance between entities that stay alive will be expanded. So, the consumed energy by these entities will be increased which speeds up their deactivation. Finally, the number of remaining active entities will be decreased and the network can stop working until harvesting energy returns or human intervention which is unpleasant.

3.4 Software & Hardware Failure Problems

We know that the sensor nodes are fragile and prone to software or hardware failures, especially if they were placed in dangerous environments. They may fail because of several reasons such as lack of energy, failure of one of the sensor node components. Because of environmental effects the sensor nodes may also sense and transmit incorrect data. Otherwise, some delays in data communication can occur because of link failures which can affect network topology. In RWSNs that do not use software/hardware failure detection techniques, every failure affects the efficiency of the network by disrupting the execution of reconfiguration scenarios and the data transmission process. In this paper, we treat the software/hardware failures that disrupt the receiving and sending tasks executed by the different network entities.

4 CONTRIBUTION: NEW ENERGY EFFICIENT AND FAULT TOLERANT METHODOLOGY IN RWSNs

In this section, we present our motivation which resumes the energy challenge and the hardware/software failures problem in RWSNs under harvesting energy constraints. We present also the formalization of the proposed methodology and the used algorithms to apply it.

4.1 Motivation

In WSNs, the principal challenge is to keep the network alive as long as possible without human intervention. In order to keep up with this challenge, it is important to reduce the consumed energy by network entities, detect the failures when they happened, treat them by isolation, and repair them as soon as possible. So, to do that, we propose a new methodology composed of a set of solutions and techniques. Firstly, we use a multi-agent architecture to manage the different reconfiguration scenarios. We use also mobile sink nodes in each zone, which collect the sensing data

Table 2: Variables and functions used to resolve the energy problem.

Var/Fun	Value	Definition
α	Constant* $0.15 \times capacity(N_{i,k}) \leq \alpha \leq 0.2 \times capacity(N_{i,k})$	is a threshold for the energy charge in sensor nodes. It is used by Ag_k to apply the mobility.
β	$EC_{[\rho]}(E) \mid E \in \{S_{Ag}, S_N, S_{SN}\}$	is a threshold for the energy charge in each entity E in the network. It is used to deactivate the entity E.
$state(E)$	$[0,1] \mid E \in \{S_N, S_{SN}, S_{Ag}\}$	if E is active it equals 1 else it equals 0.
$isFree(E)$	$[0,1] \mid E \in \{S_{MN}, S_{SN}\}$	if E has moved recently as close to a sensor node $N_{a,k}$ it equals 0 else it equals 1.
$EPres(N_{i,k})$	$[0,1]$	if the energy problem is resolved recently it equals 1 else it equals 0.
$Nb_{Act}(N_{i,k}, Z_k)$	$\sum_{i=1}^{Nb_N(Z_k)} state(N_{i,k})$	is the total number of active nodes in Z_k.
$Nb_{Act}(SN_{m,k}, Z_k)$	$\sum_{i=1}^{Nb_N(Z_k)} state(SN_{m,k})$	is the total number of active mobile sink nodes in Z_k
γ	Constant* $0.3 \times NbN(Z_k) \leq \gamma \leq 0.4 \times NbN(Z_k)$	is a threshold for the number of active sensor nodes in Z_k. It is used by Ag_k to apply the mobility.
λ	Constant* $0.15\times \leq NbN(Z_k)\lambda \leq 0.2 \times NbN(Z_k)$	is a threshold for the number of active sensor nodes in Z_k. It is used by Ag_{Ctrl} to apply the resizing.

from sensor nodes and send them to zone agents. We also apply the mobility and the resizing of zones to reduce the consumed energy and to guarantee the coverage of the largest possible area in the network. To resolve the hardware/software failures problem, we use a test packet technique. In order to prove the performance of the proposed methodology, we use two routing protocols which are LEACH and WBM-TEEN. The first one is characterized as an energy-wasting routing protocol compared with WBM-TEEN protocol, but it guarantees early failure detection using the test packet technique. The second one is an energy efficient protocol, but the failure detection may be delayed or not ensured. Finally, we develop a simulator named *RWSNSim* which permits simulating WSNs and RWSNs using the proposed methodology.

4.2 Formalization

Table 2 defines a set of variables and functions which are used to describe the energy problem in RWSN.

We present in the following a set of variables used to detect the malfunctioning entities and isolate them:

- $\delta \in [0..1]$: is a percentage factor used to fix the waiting time to receive the sensing data or the acknowledge messages.
- $t_{rep_1}(N_{i,k})$: is a response time of $N_{i,k}$ if it receives a request of sensing data from its predecessors. It is used by $SN_{m,k}$ to detect the cluster containing the malfunctioning node. It is formalized by:

$$\begin{aligned} t_{rep_1}(N_{i,k}) = Nb_{pred}(N_{i,k}) \times t_{exec}(\tau_1(N_{i,k})) \\ + Nb_{Sens}(N_{i,k}) \times t_{exec}(\tau_5(N_{i,k})) \\ + |N_{i,k}E| \times t_{exec}(\tau_2(N_{i,k}))] \end{aligned} \quad (4)$$

where $Nb_{pred}(N_{i,k})$ is the number of active predecessors of $N_{i,k}$, E is the successor of $N_{i,k}$ and $|N_{i,k}E|$ is the distance between $N_{i,k}$ and E.

- $dl_1(N_{i,k}, SN_{m,k})$: is a receiving task deadline used by $SN_{m,k}$ to receive the sensing data from $N_{i,k}$, where $N_{i,k}$ is a cluster head. It is formalized by:

$$\begin{aligned} dl_1(N_{i,k}, SN_{m,k}) = (1+\delta) \times (\sum_{i=1}^{Nb_{succ}(SN_{m,k})} \\ (t_{exec}(\tau_2(SN_{m,k})) \times |VSucc_{SN_{m,k}}[i] SN_{m,k}|)) \\ + \sum_{j=1}^{Nb_N(SZ_{m,k})} (t_{rep_1}(VNAct_{SZ_{m,k}}[j])) \\ + Nb_{pred}(SN_{m,k}) \times t_{exec}(\tau_1(SN_{m,k})) \end{aligned} \quad (5)$$

where $Nb_{succ}(SN_{m,k})$ is the number of active successors of $SN_{m,k}$, $VSucc_{SN_{m,k}}$ is the vector which contains them, and $VNAct_{SZ_{m,k}}$ is the list of active sensor nodes in $SZ_{m,k}$.

- $t_{rep_1}(SN_{m,k})$: is a response time of $SN_{m,k}$ if it receives a request of sensing data from Ag_k. It is formalized by:

Table 3: Variables used to send alert messages to Ag_{Ctrl} and BS to calling a human intervention.

Var	Value	Definition
$Nb_{flrN}(Z_k)$	$[0..Nb_N(Z_k)]$	is the number of malfunctioning sensor nodes in Z_k
$Nb_{flrSN}(Z_k)$	$[0..Nb_{SN}(Z_k)]$	is the number of malfunctioning sink nodes in Z_k
$Nb_{flrAg}(R)$	$[0..Nb_Z(R)]$	is the number of malfunctioning zone agents in R
$flrN$	Constant* $flrN \geq 0.1 \times NbN(Z_k)$	is a threshold for the number of malfunctioning sensor nodes in each zone.
$flrSN$	Constant* $flrSN \geq 0.2 \times NbSN(Z_k)$	is a threshold for the number of malfunctioning sink nodes in each zone.
$flrAg$	Constant* $flrAg \geq 0.1 \times NbZ$	is a threshold for the number of malfunctioning zone agents in R.

$$t_{rep_1}(SN_{m,k}) = t_{exec}(\tau_1(SN_{m,k})) + \sum_{i=1}^{Nb_{succ}(SN_{m,k})} (t_{exec}(\tau_2(SN_{m,k})) \times |VSucc_{SN_{m,k}}[i]SN_{m,k}|) + \sum_{j=1}^{Nb_{NAct}(SZ_{m,k})} (t_{rep_1}(VNAct_{SZ_{m,k}}[j])) + Nb_{pred}(SN_{m,k}) \times t_{exec}(\tau_1(SN_{m,k})) + |SN_{m,k}Ag_k| \times t_{exec}(\tau_2(SN_{m,k})) \quad (6)$$

where $Nb_{NAct}(SZ_{m,k})$ is the number of active sensor nodes in the subzone $SZ_{m,k}$, $VN_{SZ_{m,k}}$ is the vector which contains them and $Nb_{pred}(SN_{m,k})$ is the number of active predecessors of $SN_{m,k}$.

- $dl_1(SN_{m,k}, Ag_k)$: is a receiving task deadline used by Ag_k to receive sensing data from $SN_{m,k}$. It is formalized by:

$$dl_1(SN_{m,k}, Ag_k) = (1+\delta) \times (|Ag_k SN_{m,k}| \times t_{exec}(\tau_2(Ag_k)) + t_{rep_1}(SN_{m,k}) + t_{exec}(\tau_1(Ag_k))) \quad (7)$$

- $t_{rep_1}(Ag_k)$: is a response time of Ag_k if it receives a request of sensing data from Ag_{Ctrl}. It is formalized by:

$$t_{rep_1}(Ag_k) = t_{exec}(\tau_1(Ag_k)) + \sum_{m=1}^{Nb_{SN}(Z_k)} (t_{rep_1}(SN_{m,k}) + (|Ag_k SN_{m,k}| \times t_{exec}(\tau_2(Ag_k)))) + Nb_{SNAct}(Z_k) \times t_{exec}(\tau_1(Ag_k)) + |Ag_k Ag_{Ctrl}| \times t_{exec}(\tau_2(Ag_k)) \quad (8)$$

where $Nb_{SNAct}(Z_k)$ is the number of active mobile sink nodes in Z_k.

- $dl_1(Ag_k, Ag_{Ctrl})$: is a receiving task deadline used by Ag_{Ctrl} to receive sensing data from Ag_k. It is formalized by:

$$dl_1(Ag_k, Ag_{Ctrl}) = (1+\delta) \times (|Ag_{Ctrl} Ag_k| \times t_{exec}(\tau_2(Ag_{Ctrl})) + t_{rep_1}(Ag_k) + t_{exec}(\tau_1(Ag_{Ctrl}))) \quad (9)$$

- $t_{rep_2}(E_1)$: is a response time of E_1 if it receives a test packet from E_2. It is formalized by:

$$t_{rep_2}(E_1) = t_{exec}(\tau_1(E_1)) + |E_1E_2| \times t_{exec}(\tau_2(E_1)) \quad (10)$$

- $dl_2(E_1, E_2)$: is a receiving task deadline used by E_2 to receive acknowledge messages from E_1. It is formalized by:

$$dl_2(E_1, E_2) = (1+\delta) \times (|E_1E_2| \times t_{exec}(\tau_2(E_2)) + t_{rep_2}(E_1) + t_{exec}(\tau_1(E_2))) \quad (11)$$

where $E_1 \in \{S_N, S_{SN}, Ag_k\}$ and $E_2 \in \{S_{SN}, S_{Ag}\}$

Table 3 defines the variables used to send alert messages to Ag_{Ctrl} and BS to calling a human intervention.

(*) All the constant values of thresholds must be defined by the administrator of the network.

We propose a set of rules to regulate the application of different reconfiguration scenarios, the proposed techniques and to distribute the different responsibilities in the network. **Rule 1** tunes the application of the mobility. **Rule 2** regulates the application of the resizing of zones. On the other hand, **Rule 3** is used to detect the malfunctioning entities. While we use **Rule 4** to isolate them. Figures 1, 2, 3 and 4 show the illustration diagrams of these rules.

Rule 1: Application of the mobility of mobile entities (mobile sensor nodes and mobile sink nodes).

Cdt 1: if $(state(N_{i,k}) = 1 \,\&\, C(N_{i,k}) \leq \alpha \,\&\, EPres(N_{i,k}) = 0 \,\&\, Nb_{Act}(N_{i,k}, Z_k) > \gamma \,\&\, N_{i,k} \in SZ_{m,k})$ then Ag_k decides to apply the mobility of $SN_{m,k}$ considering the following subconditions:

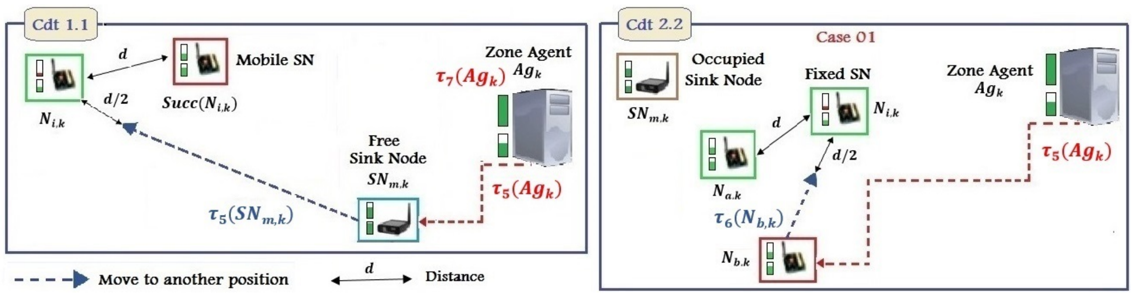

Figure 1: **Rule 1** illustration diagram.

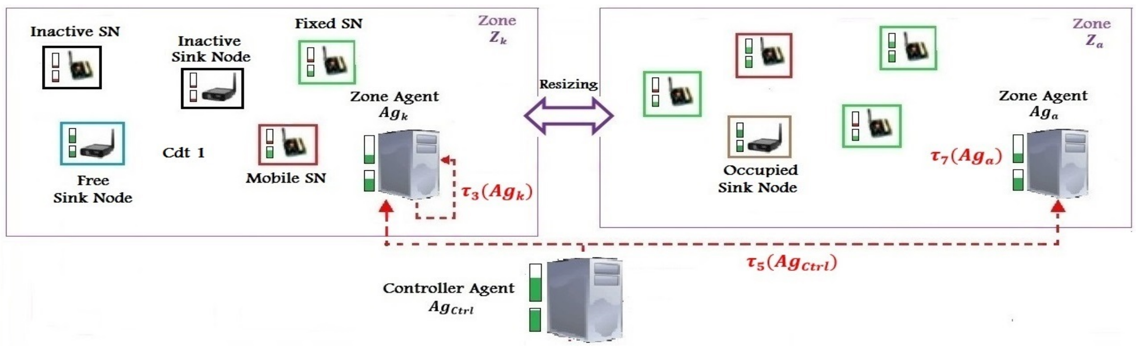

Figure 2: **Rule 2** illustration diagram.

Cdt 1.1: if $(isFree(SN_{m,k}) = 1)$ then the following tasks will be executed by Ag_k and $SN_{m,k}$:

$$\tau_5(Ag_k) = ApplyMobility()$$
$$\tau_5(SN_{m,k}) = MoveTo()$$

Cdt 1.2: if $(isFree(SN_{m,k}) = 0)$ then $N_{i,k}$ must continue its execution at the same pace until the energy problem is solved or until the deactivation.

Cdt 2: if $(state(N_{i,k}) = 1 \,\&\, C(N_{i,k}) \leq \alpha \,\&\, EPres(N_{i,k}) = 0 \,\&\, Nb_{Act}(N_{i,k}, Z_k) \leq \gamma \,\&\, N_{i,k} \in SZ_{m,k})$ then Ag_k decides to apply the mobility of $SN_{m,k}$ or of the closest mobile sensor node to $N_{i,k}$ in $SZ_{m,k}$ considering the following subconditions:

Cdt 2.1: if $(|N_{i,k}SN_{m,k}| \leq |N_{i,k}E|$ where $E \in \{SN_{m,k}, N_{b,k}\}$ such that $N_{b,k}$ is the closest mobile sensor node to $N_{i,k})$ then Ag_k has the following two cases:

case 1: if $(isFree(SN_{m,k}) = 1)$ then Ag_k decides to apply the mobility of $SN_{m,k}$ (**Cdt 1.1** without considering the **Cdt 1** conditions).

case 2: if $(isFree(SN_{m,k}) = 0)$ then Ag_k decides to apply the mobility of $N_{b,k}$ (**Cdt 2.2** without considering the **Cdt 2** conditions).

Cdt 2.2: if $(|N_{i,k}SN_{m,k}| > |N_{i,k}N_{b,k}|)$ then Ag_k has the following two cases:

case 1: if $(isFree(N_{b,k} = 1) \;\&\; C(N_{b,k}) > \alpha + [\int_{t_x}^{t_y}(e_c(\tau_6(N_{b,k}))dt + \varepsilon)])$ where:

$$e_c(\tau_6(N_{b,k})) = (|N_{i,k}N_{b,k}| - |N_{i,k}N_{a,k}|/2) \times e_{Mob}$$

where $N_{a,k}$ is the successor of $N_{i,k}$ and e_{Mob} is the energy consumed by $N_{b,k}$ to move one meter.

In this case, Ag_k decides to apply the mobility of $N_{b,k}$. Therefore the following tasks will be executed by Ag_k and $N_{b,k}$:

$$\tau_5(Ag_k) = ApplyMobility()$$
$$\tau_6(N_{b,k}) = MoveTo()$$
$$\tau_7(Ag_k) = OrganizeNodes()$$

Case 2: if $(isFree(N_{b,k} = 0)$ or $C(N_{b,k}) \leq \alpha + [\int_{t_x}^{t_y}(e_c(\tau_6(N_{b,k}))dt + \varepsilon)])$ then Ag_k decides to apply the mobility of $SN_{m,k}$ (**Cdt 1** without considering the **Cdt 1** conditions).

Rule 2: Application of the resizing of zones in R.

Cdt 1: if $(Nb_{Act}(N_{i,k}, Z_k) \leq \lambda)$ then Ag_{Ctrl} decides to apply the resizing between Z_k and the neighbor zone Z_a which contains the minimum number of active sensor nodes and active sink nodes, or only the zone which contains the minimum number of active sink nodes in case of equality of active sensor nodes and vice versa. Therefore, the following tasks will be executed by Ag_{Ctrl} and Ag_k:

$$\tau_5(Ag_{Ctrl}) = Resizing()$$
$$\tau_3(Ag_k) = Deactivate()$$
$$\tau_7(Ag_k) = OrganizeNodes()$$

Rule 3: Detecting the malfunctioning entities in R.

Cdt 1: if $((t_{rep_1}(N_{i,k}) > dl_1(N_{c,k}, SN_{m,k}))$ where $N_{c,k}$ is a cluster head) then $SN_{m,k}$ sends a test packet to

$N_{c,k}$ cluster sensor nodes from the closest one to the malfunctioning one to detect it.

Cdt 1.1: if$((t_{rep_2}(N_{i,k}) > dl_2(N_{i,k}, SN_{m,k})))$ then $SN_{m,k}$ sends an alert message to Ag_k to inform it of a failure in $N_{i,k}$ to isolate it. Therefore the following tasks will be executed by Ag_k:

$$\tau_6(Ag_k) = Isolate(N_{i,k})$$
$$\tau_7(Ag_k) = OrganizeNodes()$$

Cdt 2: if $((t_{rep_1}(SN_{m,k}) > dl_1(SN_{m,k}, Ag_k)))$ then Ag_k decides to send a test packet to $SN_{m,k}$ to detect if it is a malfunctioning sink node.

Cdt 2.1: if $((t_{rep_2}(SN_{m,k}) > dl_2(SN_{m,k}, Ag_k)))$ then Ag_k decides to isolate $SN_{m,k}$. Therefore the following tasks will be executed by Ag_k:

$$\tau_6(Ag_k) = Isolate(SN_{m,k})$$
$$\tau_7(Ag_k) = OrganizeNodes()$$

Cdt 3: if $((t_{rep_1}(Ag_k) > dl_1(Ag_k, Ag_{Ctrl})))$ then Ag_{Ctrl} decides to send a test packet to Ag_k to detect if it is a malfunctioning zone agent.

Cdt 3.1: if $((t_{rep_2}(Ag_k) > dl_2(Ag_k, Ag_{Ctrl})))$ then Ag_{Ctrl} decides to isolate Ag_k and apply the resizing of zones (**Rule 2**). Therefore the following tasks will be executed by Ag_{Ctrl}:

$$\tau_6(Ag_{Ctrl}) = Isolate(Ag_k)$$
$$\tau_5(Ag_{Ctrl}) = Resizing()$$

Rule 4: Resolve the hardware/software failures problem in R.

Cdt 1: if $(Nb_{flrN}(Z_k) \geq flrN \,||\, Nb_{flrSN}(Z_k) \geq flrSN)$ then Ag_k sends an alert message to Ag_{Ctrl} that sends it to the base station BS for human intervention.

Cdt 2: if $Nb_{flrAg}(R) \geq flrAg$ then Ag_{Ctrl} sends an alert message to the base station BS for human intervention.

4.3 Algorithms

To apply the mobility in Z_k, Ag_k must execute **Algorithm 1** according to the **Rule 1** conditions. Otherwise, the controller agent Ag_{Ctrl} executes **Algorithm 2** to apply the resizing task according to the **Rule 2** conditions. Finally, each zone agent Ag_k must execute **Algorithm 3 & 4** to detect and isolate the malfunctioning sinks.

5 EXPERIMENTATION

In the following, we will describe the *RWSNSim* simulator mentioning the services it provides. Then, we will present a case study of WSN and RWSN designated to protect the forests against fires to preserve the lives of animals and humans. Finally, we will evaluate the performance of the proposed methodology.

Algorithm 1: Apply the mobility in Z_k.

```
Input: Set of sensor nodes and mobile sink nodes.
Output: Minimize the total distance between sensor nodes and mobile sink nodes.
  for $i = 0$ to $VSZ.size()$ do
    for $j = 0$ to $VSZ.get(i).vnactive.size()$ do
      $N \leftarrow VSZ.get(i).vnactive.get(j)$
      if (Rule 1.Cdt 1.Cdt 1.1) || (Rule 1.Cdt 2.Cdt 2.1.case 1) || (Rule 1.Cdt 2.Cdt 2.2.case 2) & (Rule 1.!(Cdt 1.Cdt 1.1)) then
        $S \leftarrow VSZ.get(i).getSink()$
        $S.moveascloseTo(N)$
      end if
      if ((Rule 1.Cdt 2.Cdt 2.1.case 2) & (Rule 1.Cdt 2.!(Cdt 2.2).case 1)) || (Rule 1.Cdt 2.Cdt 2.2.case 1) then
        $M \leftarrow N.getclosestMN()$
        $M.moveascloseTo(N)$
      end if
    end for
  end for
```

Algorithm 2: Resizing of zones in R.

```
Input: Set of active zones.
Output: Cover the possible largest zones in R.
  for $i = 0$ to $NbZ$ do
    $Z \leftarrow VZ.get(i)$
    if Rule 2.Cdt 1 then
      $Z1 \leftarrow findAppropriateNeigh()$
    end if
    if $Z \neq null$ AND $Z1 \neq null$ then
      $Z.Ag.deactivate()$
      for $j = 0$ to $Z.Ag.NbNAct$ do
        $Z1.Ag.VN.add(Z.Ag.VNAct.get(j))$
        for $k = 0$ to $Z.NbSink$ do
          $S \leftarrow Z.Ag.VSZ.get(k).getSink()$
          if $S.state = 1$ then
            $Z1.Ag.VSZ.add(Z.Ag.VSZ.get(k))$
          end if
        end for
      end for
      $findNeigh(Z1)$
    end if
  end for
```

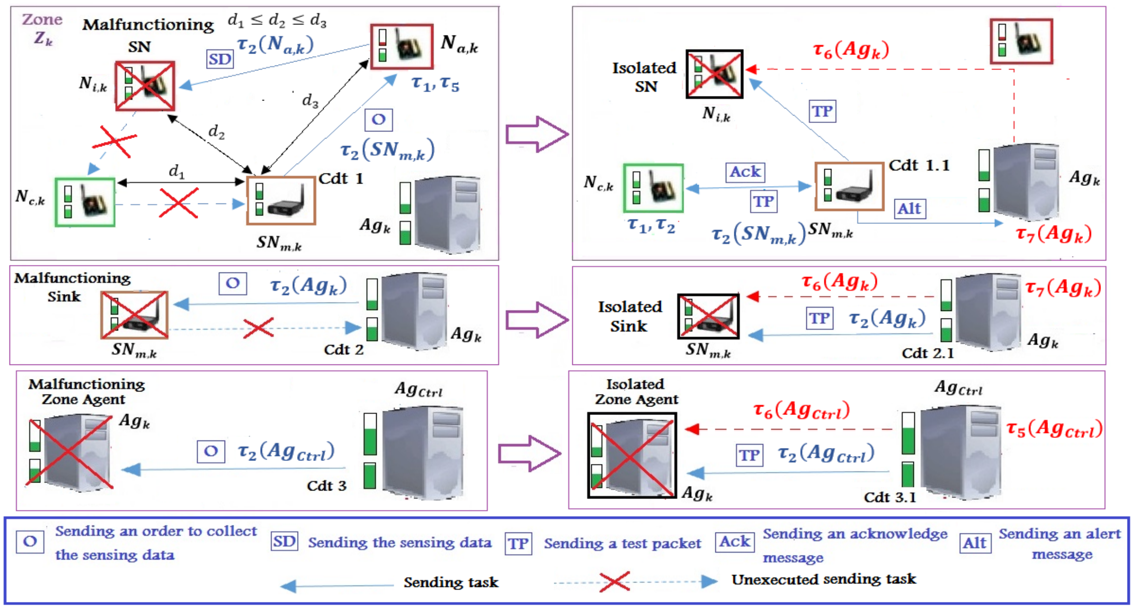

Figure 3: **Rule 3** illustration diagram.

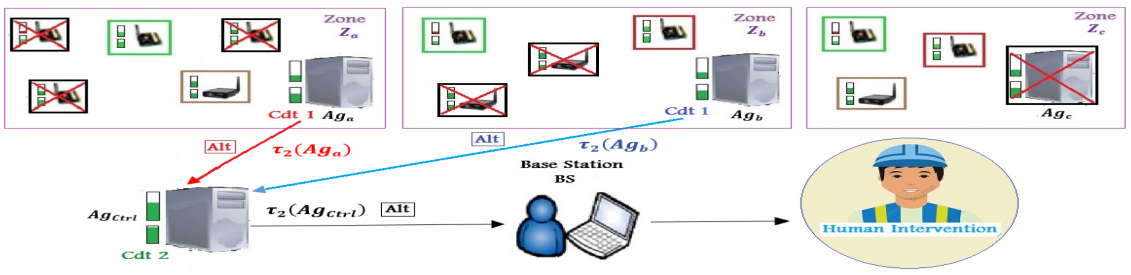

Figure 4: **Rule 4** illustration diagram.

Algorithm 3: Detect the malfunctioning sinks by Ag_k.

Input: Set of active sink nodes.
Output: Detect the malfunctioning sinks.

```
$sendTstPck \leftarrow false$
$AllSinksVerif \leftarrow false$
$startTime \leftarrow currentTime()$
while $AllSinksVerif = false$ do
  $endTime \leftarrow currentTime()$
  $waitTime \leftarrow endTime - startTime$
  for $i = 0$ to $VS.size()$ do
    if $waitTime > dl1(VS.get(i), Ag)$ then
      $VTstPck.add(VS.get(i))$
      $sendTstPckTo(VS.get(i))$
      $sendTstPck \leftarrow true$
    end if
  end for
end while
```

Algorithm 4: isolate the malfunctioning sinks by Ag_k.

Input: Set of active sink nodes.
Output: Isolate the malfunctioning sink nodes.

```
$AllSinksVerif \leftarrow false$
if $sendTstPck = true$ then
  $startTime \leftarrow currentTime()$
  while $AllSinksVerif = false$ do
    $endTime \leftarrow currentTime()$
    $waitTime \leftarrow endTime - startTime$
    for $i = 0$ to $VTstPck.size()$ do
      $S \leftarrow VTstPck.get(i)$
      if $waitTime > dl2(S, Ag)$ then
        $isolate(S)$
      end if
    end for
  end while
end if
```

5.1 RWSNSim Simulator

To be able to evaluate the proposed methodology, we develop a simulator named *RWSNSim* using *Java*. It permits creating WSNs and RWSNs and save them in a database using *hsqldb*. It provides two routing protocols (LEACH and WBM-TEEN) and executes the simulation graph using *jgraph*, *jgraphx*, and *jgrapht* libraries. It provides also an execution report for each monitoring time, drawing the resulting line charts after the simulation using *jfreechart* library. Finally, It permits comparing the different networks and simulations.

5.2 Case Study

In order to clarify the performance of the proposed methodology, we use in this case study a small WSN (W) and a small RWSN (R) designated to protect the forests against fires.

(W) consists of *i)* a base station BS, *ii)* 4 zones $\{Z_1, Z_2, Z_3, Z_4\}$, *iii)* each zone is composed of a gateway G_k, where $k \in [1..4]$ and a set of 20 nodes defined by $S_N(Z_k) = \{N_{i,k} \quad i \in [1..20] \quad and \quad k \in [0..4]\}$, and *iv)* each node $N_{i,k}$ has a temperature and a CO_2 sensor.

(R) is composed of *i)* a base station BS, *ii)* a controller agent Ag_{Ctrl}, *iii)* 4 zones $\{Z_1, Z_2, Z_3, Z_4\}$, *iv)* each zone contains a zone agent Ag_k, where $k \in [1..4]$, a set of 3 mobile sink nodes defined by $S_{SN}(Z_k) = SN_{j,k} \quad j \in [1..3] \quad and \quad k \in [1..4]$ and a set of 20 nodes (12 fixed and 8 mobile ones) defined by $S_N(Z_k) = \{N_{i,k} \quad i \in [1..20] \quad and \quad k \in [1..4]\}$, and *v)* each node $N_{i,k}$ has a temperature and a CO_2 sensor.

5.3 Evaluation of Performance

To figure out the performance of the proposed methodology, we execute the proposed case study in *RWSNSim*. In the following, we will show the obtained results in the worst case. That is, during the months when the energy production by each entity E is negligible ($EP_{[\rho]}(E) \approx 0$) and the consumed energy is high ($EC_{[\rho]}(E) > 0$). Figure 5 shows the obtained line charts. Table 4 resumes the obtained results as percentages of success.

Table 4: The obtained results as percentages of success.

	(a) & (c)	(b) & (d)	% success
(a) & (b)	16 days	21 days	31.25%
(c) & (d)	88 days	116 days	31,82%
% success	450%	452.38%	625%

Through Figure 5 and Table 4, we remark that the effectiveness of the use of WBM-TEEN protocol achieves a success rate approx 32%. While, the effectiveness of the proposed methodology with the use of a same routing protocol achieves a success rate approx 450%. Otherwise, the proposed methodology achieves a success rate of 625% with the use of WBM-TEEN protocol compared to the case when we use LEACH protocol without the proposed methodology.

On the other side, when a software/hardware failure is committed in an entity in case **(a)** or **(b)**, the packets dropping problem will occur in W. The problem may reach a complete network shutdown. While in case **(c)**, the failure will be detected and isolated. Thus, eliminating the packets dropping problem and R will still work without problems.

According to the obtained results, we can come up with an important inference which is the choosing of the appropriate routing protocol. We can choose LEACH protocol as an appropriate routing protocol in two cases. The first one is if the network needs periodic monitoring and the sensor nodes are required to send the sensing data periodically. The second one is if the network needs to detect the failures quickly and isolate the malfunctioning entities when they happen. Otherwise, we can choose WBM-TEEN protocol as an appropriate routing protocol in two cases. The first one is if the network doesn't need periodic monitoring and the sensor nodes are required to send the sensing data only when their values exceed the thresholds. The second one is if the network needs more to reduce the energy consumption than the failure detection.

To show the benefits of the proposed methodology compared to related works, we make two comparison scenarios. In the first scenario, we compare between different simulations of R defined as follows:

- *Simulation 1:* with an energy efficient routing protocol.
- *Simulation 2:* with resizing of zones.
- *Simulation 3:* with mobility of mobile nodes.
- *Simulation 4:* with mobility of mobile entities.
- *Simulation 5:* with proposed methodology.

Figure 6 shows the results of these simulations.

Based on the first scenario, we conclude that the use of only one or two solutions to resolve the cited problems, as in some related works, can extend the lifetime of the network. In fact, this network lifetime extension is by a small percentage compared to the achieved extension using the proposed methodology.

In the second scenario, we compare the percentage of success of this work with related works. We have three contributions defined as follows:

- **Cont 1:** is the contribution of the paper (Grichi et al., 2018).

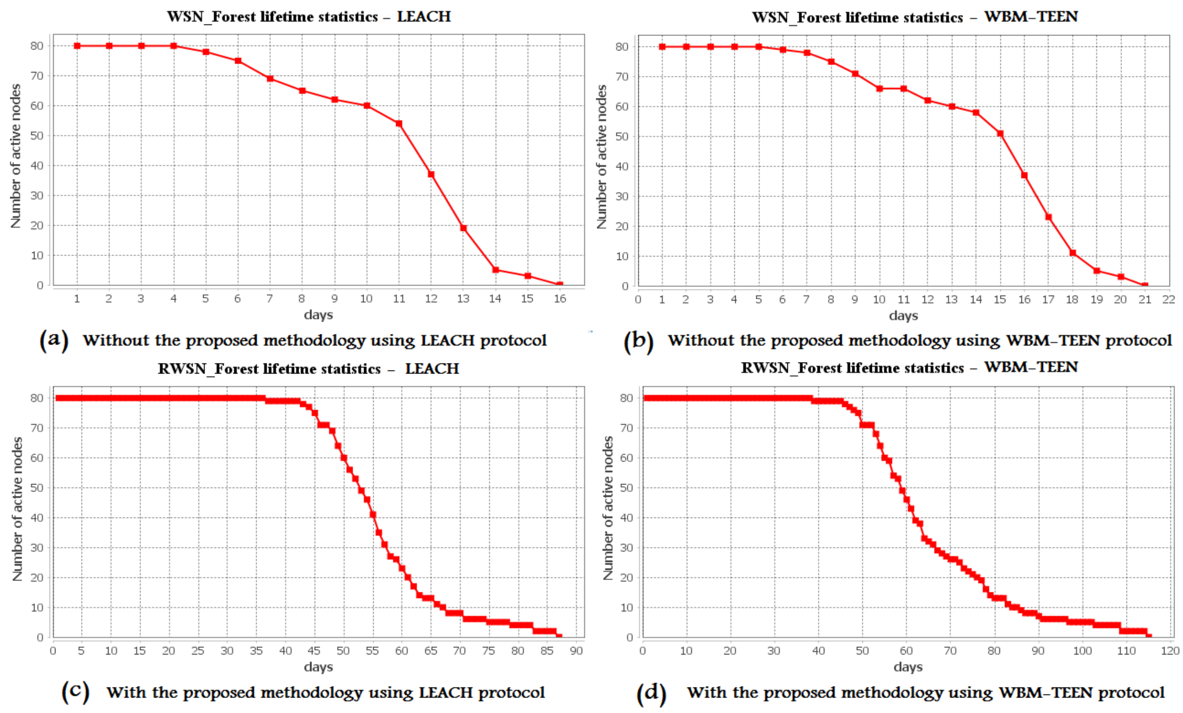

Figure 5: Simulation results.

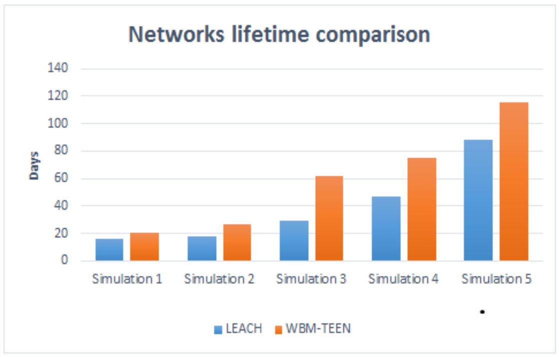

Figure 6: Networks lifetime comparison.

- **Cont 2:** is the contribution of the paper (Rouainia et al., 2020).
- **Cont 3:** is the proposed contribution of this paper.

To ensure an accurate comparison between these three contributions, we simulated the proposed case study using the proposed methodology in each contribution. Table 5 presents the obtained results.

Table 5: The obtained results of the three contributions.

	Cont 1	Cont 2	Cont 3
LEACH	136.36%	248%	450%
WBM-TEEN	172.09%	288.2%	452.38%

Looking at previous comparisons, we remark that by using the proposed methodology we can keep the network alive two to three times as long as compared to related works whatever the used routing protocol.

Finally, taking into consideration that this experimentation was executed by low batteries charged to avoid extending the simulation time. We conclude that the proposed methodology proves its efficacy in keeping the network alive for very long periods that may reach decades without human intervention.

6 CONCLUSION

In this paper, we propose a new methodology to resolve the energy and hardware/software failure problems in RWSNs to keep the network alive as long as possible. It consists of the use of mobile sink nodes, the application of mobility and resizing of zones, and using the test packet technique to detect the malfunctioning entities and isolate them. It is based on a multi-agent architecture and an energy efficient protocol taking into consideration a set of rules. Otherwise, we develop a simulator named *RWSNSim* which permits the construction of WSNs and RWSNs with and without using the proposed methodology. It also provides two routing protocols LEACH and WBM-TEEN. We have proven the efficacy of the proposed methodology, which reached a success rate of 625%. In the future, we will strive to improve the effectiveness of the proposed methodology. We aim also to improve the simulator *RWSNSim* and make it include more options and routing protocols.

REFERENCES

Agrawal, D. P. (2017). Sensor nodes (sns), camera sensor nodes (c-sns), and remote sensor nodes (rsns). In *Embedded Sensor Systems*, pages 181–194. Springer.

Allouch, A., Cheikhrouhou, O., Koubâa, A., Khalgui, M., and Abbes, T. (2019). Mavsec: Securing the mavlink protocol for ardupilot/px4 unmanned aerial systems. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 621–628.

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. *Complex.*, 2019:5781620:1–5781620:10.

Erman, A. T., Dilo, A., and Havinga, P. J. M. (2012). A virtual infrastructure based on honeycomb tessellation for data dissemination in multi-sink mobile wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.*, 17:1–27.

Ghribi, I., Abdallah, R. B., Khalgui, M., Li, Z., Alnowibet, K. A., and Platzner, M. (2018). R-codesign: Codesign methodology for real-time reconfigurable embedded systems under energy constraints. *IEEE Access*, 6:14078–14092.

Grichi, H., Mosbahi, O., Khalgui, M., and Li, Z. (2018). New power-oriented methodology for dynamic resizing and mobility of reconfigurable wireless sensor networks. *IEEE Trans. Syst. Man Cybern. Syst.*, 48(7):1120–1130.

Hafidi, Y., Kahloul, L., Khalgui, M., Li, Z., Alnowibet, K., and Qu, T. (2020). On methodology for the verification of reconfigurable timed net condition/event systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(10):3577–3591.

Housseyni, W., Mosbahi, O., Khalgui, M., Li, Z., and Yin, L. (2018). Multiagent architecture for distributed adaptive scheduling of reconfigurable real-time tasks with energy harvesting constraints. *IEEE Access*, 6:2068–2084.

Ji, S., Beyah, R. A., and Cai, Z. (2014). Snapshot and continuous data collection in probabilistic wireless sensor networks. *IEEE Trans. Mob. Comput.*, 13(3):626–637.

Khediri, S. E., Nasri, N., Khan, R. U., and Kachouri, A. (2021). An improved energy efficient clustering protocol for increasing the life time of wireless sensor networks. *Wirel. Pers. Commun.*, 116(1):539–558.

Moussa, N., Alaoui, A. E. B. E., and Chaudet, C. (2020). A novel approach of WSN routing protocols comparison for forest fire detection. *Wirel. Networks*, 26(3):1857–1867.

Nguyen, L. and Nguyen, H. T. (2020). Mobility based network lifetime in wireless sensor networks: A review. *Comput. Networks*, 174:107236.

Raj, S. N., Bhattacharyya, D., Midhunchakkaravarthy, D., and Kim, T. (2021). Multi-hop in clustering with mobility protocol to save the energy utilization in wireless sensor networks. *Wirel. Pers. Commun.*, 117(4):3381–3395.

Ramasamy, V. (2017). Mobile wireless sensor networks: An overview. *Wireless Sensor Networks—Insights and Innovations*, pages 1–12.

Robinson, Y. H., Julie, E. G., and Balaji, S. (2017). Bandwidth and delay aware routing protocol with scheduling algorithm for multi hop mobile ad hoc networks. *Academy of Science, Engineering and Technology*, 10(8):1512–1521.

Rouainia, H., Grichi, H., Kahloul, L., and Khalgui, M. (2020). 3d mobility, resizing and mobile sink nodes in reconfigurable wireless sensor networks based on multi-agent architecture under energy harvesting constraints. In *Proceedings of ICSOFT 2020*, volume 97, pages 394–406. ScitePress.

Salem, M. O. B. (2017). *BROMETH: Methodology to Develop Safe Reconfigurable Medical Robotic Systems: Application on Pediatric Supracondylar Humeral Fracture*. PhD thesis, Saarland University, Saarbrücken, Germany.

Shalini, V. B. and Vasudevan, V. (2021). e-nl beenish: extended-network lifetime balanced energy efficient network integrated super heterogeneous protocol for a wireless sensor network. *Int. J. Comput. Aided Eng. Technol.*, 15(2-3):317–327.

Tzounis, A., Katsoulas, N., Bartzanas, T., and Kittas, C. (2017). Internet of things in agriculture, recent advances and future challenges. *Biosystems engineering*, 164:31–48.

Vijayalakshmi, S. and Muruganand, S. (2018). *Wireless Sensor Network: Architecture - Applications - Advancements*. Mercury Learning & Information, 1 edition.

Wang, J., Gao, Y., Liu, W., Sangaiah, A. K., and Kim, H. (2019). Energy efficient routing algorithm with mobile sink support for wireless sensor networks. *Sensors*, 19(7):1494–1512.

Zagrouba, R. and Kardi, A. (2021). Comparative study of energy efficient routing techniques in wireless sensor networks. *Inf.*, 12(1):1–28.

Zhong, P. and Ruan, F. (2018). An energy efficient multiple mobile sinks based routing algorithm for wireless sensor networks. In *IOP Conference Series: Materials Science and Engineering*, volume 323, pages 1–4. IOP Publishing.