LEBANESE UNIVERSITY
FACULTY OF TECHNOLOGY

Thesis for Obtaining
Master's degree II
ENGINEERING OF INFORMATION SYSTEMS

Clustering Routing Protocol for Homogeneous Wireless Sensor Networks Powered by Renewable Energy Sources

Presented by:
Chirine Bassil

[/ /]

Jury of Defense:

Reader : Pr. Mohammad Hajjar
Reader : Dr. Haissam Hajjar
Supervisor : Dr. Hussein ElGhor

Promotion : 2019-2020

LEBANESE UNIVERSITY
FACULTY OF TECHNOLOGY
MASTER OF INFORMATION SYSTEM ENGINEERING
LABORATORY OF EMBERDDED AND NETWORKED SYSTEMS

Clustering Routing Protocol for Homogeneous Wireless Sensor Networks Powered by Renewable Energy Sources

MASTER THESIS

Chirine Bassil

SAIDA-2020

Advisor: Dr. Hussein EL GHOR

Acknowledgments

It is often difficult to thank people who help you accomplish the tasks given to you, yet we must express the full gratitude we feel towards them.

Words are powerless to express my gratitude to Dr. Hussein EL Ghor, a great professor, for his dedication, organization, enthusiasm and hard work. For being so encouraging and doing everything he can to keep me motivated and supported throughout my thesis.

Besides my advisor, how I can forget about the master committee and coordinator, especially the dean of the Faculty of Technology professor Mohamed Hajjar for giving me the wisdom.

Although it is difficult to list them, all my colleague , they always have been there with a good hand to encourage me in moments of weakness, and my best friends, they always know the right things to say when things weren't going my way, all the thanks for your never-ending support.

Last but not least, I would like also to thank my parents for infinite guidance and encouragement for a better future and to accomplish my thesis.

Abstract

The technology of wireless sensor networks (WSNs) is in constant development and it made great progress in many applications. One of the most popular problems in WSNs is the limited energy storage power at every sensor node. Hence, energy constraints have drawn an enormous attention for sensor nodes that harvest energy from different renewable energy sources. Cluster-based routing protocols is deemed a promising approach for the topology management, energy consumption, data transfer and stability in a distributed manner. This thesis aims to propose and develop a new distributed clustering algorithm for energy harvesting wireless sensor networks denoted by DEH-WSN (Energy Harvesting for Distributed Clustering Wireless Sensor Networks Protocol) that relies on matching between clustering and energy harvesting in a distributed topology. DEH-WSN uses initial and residual energy level of the nodes to choose cluster heads. Simulation results prove that the proposed method increases network lifetime and the effective throughput.

Keywords

WSN, Routing, Clustering, Energy harvesting EH-WSN, Network Lifetime.

Table of Contents

List of Figures

List of Tables

Chapter 1-General Introduction

- *Problem Statements.*
- *The main targets of this thesis.*
- *What this thesis includes.*

Chapter 1

General Introduction

Our work in this thesis is motivated by the rapid growth in wireless sensor networks (WSNs). WSNs are developed by the integration of sensing and wireless communication. They consist of low-power small sensor nodes that are able to sense, process and interact over unreliable short-range radio connections [26–27]. These sensor nodes are deployed in the target area to observe physical or environmental conditions like temperature, sound, vibration, or pressure. The essential subsystems of sensor nodes are used to acquire data for local processing and for sharing information by wireless communication.

WSNs are characterized by their potential applications in various fields [28], especially for disaster warning systems, ecological monitoring, Healthcare, intrusion detection, fire detection [29, 30] to name but few.

Problems Statement:

Sensor nodes can fail due to energy starvation since they rely on batteries with limited energy capacity. Hence, energy harvesting system that harvests energy from renewable energy sources (solar, wind, vibrations, etc.) is proposed as a solution to avoid energy shortage in the battery [31] where the harvested energy is greater than the energy consumption. The energy harvesting is used in multiple conditions to implement the power systems such as no energy storage, energy storage without battery, and rechargeable battery. However, since the energy consumption of sensor nodes is much higher than the charging rate, these nodes need to repose for some time to recharge, but this drives a modification in the network's topology.

The Main Targets Of This Thesis:

- Overcome the shortage of limited battery capacity by transforming the distributed energy-efficient protocol to a distributed energy Harvesting protocol [1, 2], based on the Solar Energy Harvested from the sun.
- The node that has a fewer delay time compared to its neighbors has higher possibilities to be a Cluster Head and it is determined according to their energy status and the amount of energy harvested. After creating a cluster and select a cluster head, each node from their cluster sends packets to the Base Station.

The Thesis Chapters Are Divided As Follows:

Chapter 2 Shows an overview of wireless sensor networks topic. We introduce concepts of WSNs including their advantages and their challenges aspects.

Chapter 3 introduces concepts and algorithms for Distributed Energy Efficient Clustering Protocol. Then, in a second step, we present the concepts of Energy Harvesting in WSNs by focusing on their problem statements, their categories, and then their classification.

Chapter 4 Targets the concept of our new protocol, where we develop the distributed Energy Management for Energy-Harvesting WSNs, named "Distributed energy harvesting in WSNs protocol (DEH-WSNs)", that focuses on the matching between the distributed protocol and the energy harvested. Therefore, we first present some related work to enhance the energy consumption by using clustering-based protocols, and then explain how these research works are related to the current work. In a second step, we start to introduce the process of our protocol including the aim of the algorithm and its steps, then an execution describes each step in the production of the experiment. Later, we demonstrate that the proposed algorithm improves energy constraints by increasing network life. The results showed an increase in the stability of the network in addition to the improvement of its efficiency.

Chapter 5 presents a summary of the study, including impact, limitations and future work.

Chapter 2-Wireless Sensor Networks (WSNs)

- *The concept of WSNs*
    - *Advantages*
    - *challenging aspects*
- *Clustering overview*
    - *Components of a clustered WSNs*
    - *Challenges Clustering Algorithm*
- *Conclusion*

Chapter 2

Wireless Sensor Networks (WSNs)

2.1. The concept of WSNs

Wireless sensor networks (WSNs) are one of the fastest-growing technologies; it's enabled by the rapid advances in micro-electro-mechanical-system (MEMS), computer networks, and wireless communication technology. WSNs have attracted enormous attention in diverse areas such as disaster warning systems, environment monitoring, safety, intruder detection, and others [3, 4, 5].

The most relevant advantages of WSNs are summarized as follows:

- Their ability to operate in harsh or unattended environments and to cover various areas including reachable and unreachable places such as in deep forests or underwater, etc [6].
- Their ability to work with complicated protocols to receive notifying messages from nodes.
- Reducing the number of required communications to perform operations or predefined computations whether centralized or distributed.
- Implementing complex modes of saving energy.

However, Sensor networks have a lot of challenging aspects, they are stated as follows:

- Limited and irreplaceable energy source: sensor nodes are operated by small and limited battery power, it is very hard or even impossible to change or recharge the batteries.
- Scalability: All the new algorithms or protocols for sensor networks must work efficiently and effectively on different network sizes.
- Fault tolerance: to hold operational WSNs, the protocols for WSNs should cope with the fault-tolerant features. Whilst the sensor nodes are quite prone to failure due to several factors such as environmental hazards and device failure, this failure can affect the overall network lifetime, it can limit the accessibility of the sensor nodes under their supervision.
- Self-organization: The network is expected to function in an unattended manner or be capable of self-organization in a harsh environment.
- Limited processing, storage, and communication: All algorithms or protocols for sensor networks should consider the energy limitations of

sensor nodes, and the processing, storage, and communication limitations.

2.2. Clustering Overview

Depending on the cluster-based architecture, the sensor nodes are divided into different cluster groups where each one has a leader known as cluster head (CH). First, all the sensor nodes sense local data then send it to their corresponding CH, and then they aggregate the local data and finally send it to the base station (BS). The functionality of a cluster-based WSN can be seen in two methods Figure 1.1 and figure 1.2.

- Figure 1.1: Cluster-based wireless sensor network architecture with single-hop communication between CHs and base stations.
- Figure 2.2: Cluster-based wireless sensor network architecture with multi-hop communication between CHs and base stations.

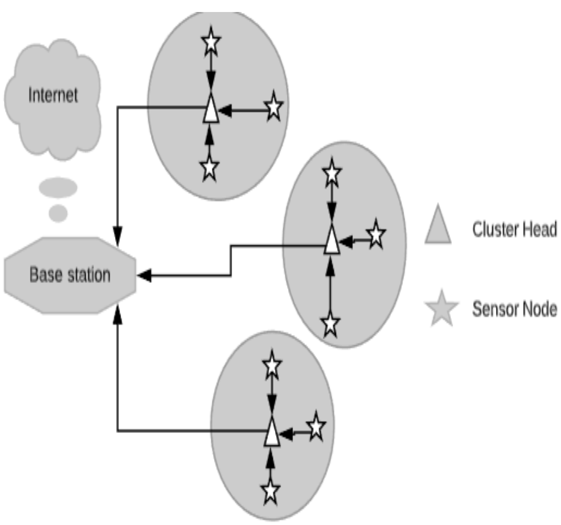

Figure 1. 1: With Single-Hop Communication

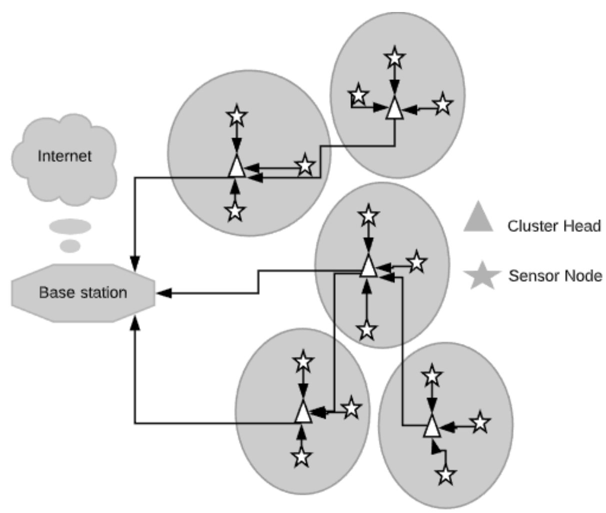

Figure 1. 2: With Multi-Hop Communication

2.2.1. Components of a clustered WSNs

The following are the components of a clustered WSN:

- *Sensor node:* The sensor node is the main component of a WSN, they perform functions such as data storage routing, sensing, and data processing WSN.

- *Clusters:* Clusters are the hierarchical units for WSNs, large sensor networks need to be divided into clusters to clarify tasks such as communication between the BS and the CHs.
- *CHs:* CHs are the leader of the cluster; they are often required to organize activities in the cluster such as data aggregation, organizing and relaying the communication schedule of a cluster.
- *BS:* BS is the sink in a WSN; it provides the communication link between the sensor network and the end-user.
- *End user:* Data is generated in WSNs in response to queries received from the end-user.

2.2.2. Challenges for clustering algorithms

Clustering schemes must consider some several key limitations in WSNs to effectively improve the network.

- *Limited energy:* Wireless sensor nodes have a reduced battery size, with limited storage capacity. Batteries cannot be recharged or replaced after exhaustion.
- *Network lifetime*: Clustering schemes help to prolong the network lifetime of WSNs by reducing energy usage.
- *Limited abilities*: The small amount of energy in a sensor node limits many of the abilities of nodes in terms of processing, memory, storage, and communication.
- *Cluster formation and CH selection*: By clustering, the WSN can avoid the energy wastage in sensors in WSN. And it can further enhance the scalability of WSN in real-world applications. The main issues in the design of clustering algorithms are selection of optimum cluster size, election, re-election of CHs, and cluster maintenance.
- *Synchronization:* Synchronization and scheduling will have a major impact on the overall network design. Cracked transmission patterns such as TDMA allow the nodes to regularly schedule nodes to be idle to reduce the energy consumption.
- *Data aggregation*: Data aggregation allows differentiation between sensed data, and user data in a large network, there are often multiple nodes sensing similar information.
- *Repair mechanisms*: It is important to look for mechanisms that ensure link recovery and reliable data communication. But due to the nature of WSNs, they are some situations can result in link failure (node death, delay, and interference)

- *Quality of service (QoS):* According to the clustering algorithms for WSN, the QoS metric must be taken into account in the design process, such as acceptable delay and packet loss tolerance.

2.3. Conclusion

The purpose of this chapter is to present some basic notions necessary for understanding the context of this thesis. At first, we recollected the background of WSNs, their advantages, and then the challenging aspects are highlighted. In a second step, we presented an overview of the clustering protocol.

In the next chapter, we will introduce the concept of the distributed energy-efficient clustering protocol, and then we will present their protocol process. Later, we will present the concept of Energy Harvesting in WSNs.

Chapter 3-State of Art

- Distributed Energy Efficient Clustering Protocol
  - Introduction
  - Heterogeneous Network Model
  - Homogeneous Network Model
  - Energy Consumption Model
- Energy Harvesting WSNs(EH-WSNs)
  - Problems of EH-WSNs
  - Categories of EH-WSNs
  - Classification of EH-WSNs
  - Solar-Based Sources
- Conclusion

Chapter 3

State of Art

3.1. Distributed Energy Efficient Clustering Protocol

Li et al. in [52] proposed a distributed energy-efficient clustering algorithm. The main idea of this Clustering Algorithm is to reduce energy consumption then increase the scalability and lifetime of the network.

3.1.1. Introduction of the Algorithm

Following the thoughts of LEACH [7], this protocol is an energy-efficient protocol for heterogeneous WSNs and it becomes homogenous after many rounds. The field is divided into different clusters. Each cluster has a CH and some sensor nodes. CH receives from a cluster the information from the sensor nodes, and then sends it to the BS. To consider the node as Cluster Head CH, a probability function is defined to compute the ratio between residual energy and the average energy of networks. The nodes with high computation have a better chance to be selected as a CH. Heterogeneous WSNs depends on two or multiple types of sensor nodes according to the levels of energy [54]. In our study, the model was devised as three levels.

Formula (1), calculates the total number of nodes that is the addition of the normal nodes and advanced nodes :

$$\boldsymbol{N} = \boldsymbol{N_{nml}} + \boldsymbol{N_{advcd}} \quad (1)$$

We consider that E_0 and $E_0 a$ represent the initial energy of a normal and advanced sensor node, respectively. a indicates how many times energies advanced node has been relative to the normal node.

Formula (2), calculates the total first energy ($\boldsymbol{E_{nml}}$) of the normal nodes:

$$\boldsymbol{E_{nml}} = \boldsymbol{N_{nml} E_0} \quad (2)$$

Formula (3), calculates the total first energy of the advanced nodes ($\boldsymbol{E_{advcd}}$):

$$\boldsymbol{E_{advcd}} = \boldsymbol{N_{advcd} E_0 a} \quad (3)$$

So, the total first energy ($\boldsymbol{E_{Total}}$) of the two-level heterogeneous WSNs is given by ***Formula (4)***:

$$\boldsymbol{E_{Total}} = \boldsymbol{E_{nml}} + \boldsymbol{E_{advcd}} \tag{4}$$

When the CH selection is performed, the distributed model includes an idea of the chances of the nodes depending on the remaining energy and the average network energy.

Formula (5), calculates the average energy ($\boldsymbol{E_{avg}}$) of the network:

$$\boldsymbol{E_{avg}} = \frac{\boldsymbol{1}}{\boldsymbol{N}} \boldsymbol{E_{Total}}\left(\boldsymbol{1} - \frac{\boldsymbol{r}}{\boldsymbol{R}}\right) \tag{5}$$

Formula (6), calculates the number of rounds predicted to the available energy and energy consumed at the current round:

$$\boldsymbol{R} = \frac{\boldsymbol{E_{Total}}}{\boldsymbol{E_{round}}} \tag{6}$$

Formula (7), calculates the threshold value at the beginning of each round:

$$T(\boldsymbol{K_i}) = \begin{cases} \dfrac{\boldsymbol{P_i}}{1 - \boldsymbol{P_i}\left(\boldsymbol{mod}\left(\boldsymbol{r}, \dfrac{\boldsymbol{1}}{\boldsymbol{P_i}}\right)\right)}, & if\ \boldsymbol{S_i} \in G \\ 0, & Otherwise \end{cases} \tag{7}$$

Formula (8), calculates the probability $\boldsymbol{P_i}$ where it should operate between 0 and 1. Here it should be noted that $\boldsymbol{E_{avg}}$ is recalculated for each round, if $\boldsymbol{E_i}(\boldsymbol{r})\boldsymbol{P_{opt}} = \boldsymbol{E_{avg}}$, the sum of all possible states is equal 1.

$$\boldsymbol{P_i} = \begin{cases} \dfrac{\boldsymbol{E_i}(\boldsymbol{r})\boldsymbol{P_{opt}}}{(1+a)\boldsymbol{E_{avg}}}\ if\ normal\ node \\ \dfrac{\boldsymbol{E_i}(\boldsymbol{r})\boldsymbol{P_{opt}}\boldsymbol{a}}{(1+a)\boldsymbol{E_{avg}}}\ if\ advance\ node \end{cases} \tag{8}$$

Table 1.1 Define each parameter of Protocol Formulas.

Parameters	Descriptions
N_{nml}	Define the numbers of normal in network
N_{advcd}	Define the numbers of advanced nodes in network
E_0	Represent the initial energy of a normal
$E_0 a$	Represent the initial energy of an advanced sensor node
a	Number of times where energies advanced node are relative to the normal node.
r	Number of rounds.
E_{round}	Define the energy consumed for each round.
G	Define the appropriate set of nodes
S_i	Define the 'i' node within the cluster
$E_i(r)$	Define the energy of the node
P_{opt}	constant probability for CH

Table 1. 1: Parameters used in Distributed Formulas.

Type of parameters	Symbol	Value
Energy depletion of the booster to deliver at a shorter distance	e_{fs}	10nJ/bit/m^2
Energy depletion of the booster to deliver at a longer distance	e_{amp}	0.0015pJ/bit/m^4
Energy depletion of the node's electronics circuit to transmit or receive the signal	E_{elec}	60nJ/bit
Energy for data aggregation	E_{DA}	5nJ/bit/signal
Threshold distance	d_0	70m
Desired probability of CH	P_{opt}	0.1
Total rounds number	R	5000
Data size	L	5000bits
Network size	-	100 * 100
Sink node position	-	(50,50)
Number of sensor nodes	N	100
Normal node numbers	N_{nml}	25
Advanced node numbers	N_{advcd}	35
Super node numbers	N_{super}	40
Network deployment	-	Randomly

Table 1. 2: Simulation Parameters

There are two phases in Distributed protocol as shown in Figure 2:

- Setup phase: the clusters are created and the cluster heads (CHs) are selected
- Steady phase: the data from non-cluster heads are transmitted to the sink.

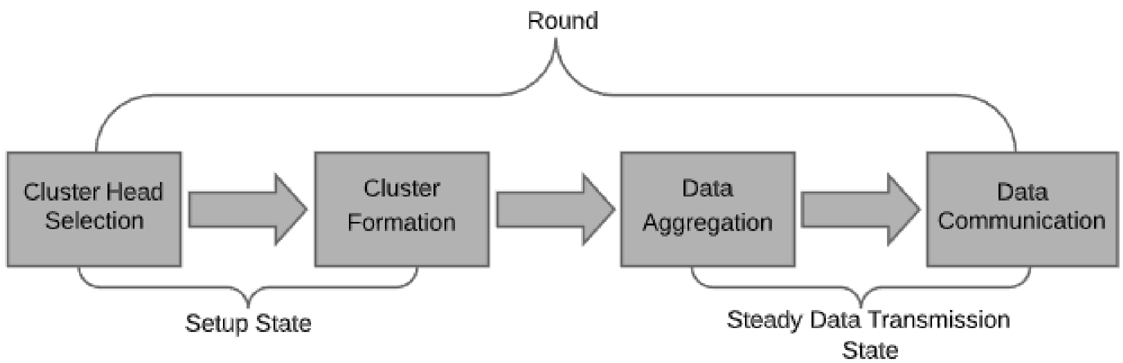

Figure 2: Flow chart of Distributed protocol

The proposed algorithm is divided into four steps:

- Initialization Phase: the possibility of being CH changes according to the capabilities of the nodes. The desired number of CH is selected according to their location. When the range of CH is defined, CH sends a membership request message to all the nodes in its range, then request to reply with their current energy status. All nodes with high residual energy and processing power will be identified and they are made to sleep, they become the backup nodes. In case the nodes are not in the range of CH, they join the cluster by sending a message to the nearest cluster member.
- Steady State Phase: the cluster members send the sensed data to the CH in the allotted time using TDMA schedule, and the non-cluster members to the cluster head through the intermediate cluster member.
- Final Phase: CH will aggregate the data from all the nodes in its cluster, and then it will transmit this data to the base station.
- Cluster Reconfiguration Phase: CH will activate the backup node if the CH residual energy reaches to the threshold value, then it will make the backup node as new CH, and transmit the new CH information to all other nodes and CH, and the old CH will become the general node in the last phase.

3.1.2. Heterogeneous Network Model

The BS is located in the center of the square region. Each node contains data that is sent to BS. Among these nodes, some are selected as CHs, and then CHs transmit the aggregated data to BS. The sensor nodes are categorized into advanced nodes and normal nodes. The total initial energy of the two-level heterogeneous networks is computed using by ***Formula (9),*** and all parameters are defined in Table 2.

$$\mathbf{E}_{total} = \boldsymbol{N}(\mathbf{1}-\boldsymbol{m})\mathbf{E}_{\mathbf{0}} + \boldsymbol{Nm}\mathbf{E}_{\mathbf{0}}(\mathbf{1}+\boldsymbol{a}) = \boldsymbol{N}\mathbf{E}_{\mathbf{0}}(\mathbf{1}+\boldsymbol{am}) \qquad (9)$$

Parameters	Descriptions
$\mathbf{E_0}$	Indicate the initial energy of normal nodes
M	Indicate the fraction of advanced nodes
Nm	Indicate total advanced nodes
$\mathbf{E_0}(\mathbf{1}+\boldsymbol{a})$	Indicate the energy associated with these nodes
$(\mathbf{1}-\boldsymbol{m})$N	Indicate total number of normal nodes

Table 2: Parameters of Formula (9)

Figure 3 illustrates a clustering structure heterogeneous model in WSNs.

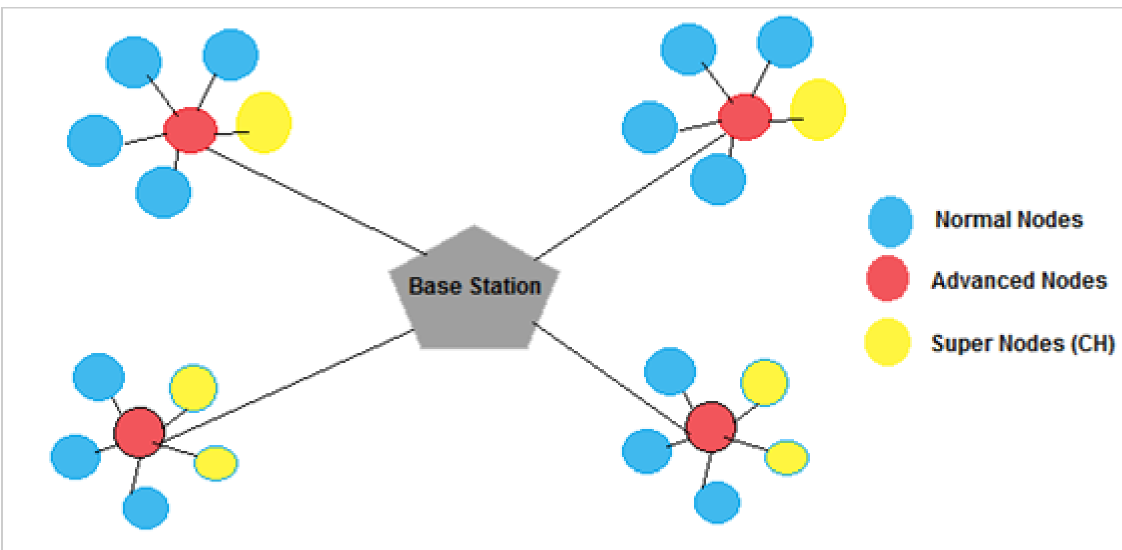

Figure 3 1: Illustrate a model of clustering heterogeneous WSN

3.1.3. Homogeneous sensor networks

A homogeneous sensor network contains BS and sensor nodes provided with the same capabilities. The data collection depends on the data dissemination structure. This structure is made in flat and hierarchical topologies.

In a hierarchical network, at first, all sensor nodes are organized in clusters. The maximum throughput determines the smallest clusters required numbers, while the higher throughput imposes the cost of nodes that acts as a cluster head. Secondly, data aggregation includes combining data in the cluster heads that reduce the number of messages sent to the base stations. So, energy consumption seeks to raise network efficiency.

2.1.4. Energy consumption model

The energy consumption of a node is divided into three objectives like sensing, processing, and wireless communication as shown in Figure 4.1.
Nodes can share input about the lifetime and the total energy of a network. The base stations give the lifetime value. The average probability is calculated by the average energy and total energy of the network. All parameters used are defined in Table 3.

The ***Formula (10)***, calculates the average energy of the network.

$$\boldsymbol{E}'(\mathbf{r}) = \frac{1}{N} E_{Total}\left(1 - \frac{r}{R}\right) \tag{10}$$

Formula (11), calculates the Network Lifetime R.

$$R = \frac{E_{Total}}{E_{round}} \tag{11}$$

Formula (12), calculates the energy expansion, and it is defined by the radio transmission of "$\boldsymbol{l}$" message and distance "$\boldsymbol{d}$".

$$E_{Tx}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2, & d < d_0 \\ lE_{elec} + l\epsilon_{mp}d^4, & d \geq d_0 \end{cases} \tag{12}$$

Hence, ***Formula (13)*** calculates the energy dissipated in a network.

$$E_{round} = L(2NE_{elec} + NE_{DA} + K\epsilon_{mpd^4(CHtoBS)} + N\epsilon_{fs}d^2(Nto\ CH) \tag{13}$$

The average distance between CH and BS is calculated using ***Formula (14)***:

$$d(CHtoBS) = \frac{M}{\sqrt{2\pi K}} \tag{14}$$

The average distance between nodes and CH is calculated using ***Formula (15)***:

$$d(Nto\ CH) = 0.765\frac{M}{2} \tag{15}$$

Parameters	Descriptions
$E'(\mathrm{r})$	Define the average energy
E_{Total}	Define the total energy of a network
R	Define the lifetime value of a network.
E_{elec}	Define the energy dissipated to run the transmitter on the receiver circuit
$\epsilon_{fs}d^2$	Define the energy dissipated in free environment
$\epsilon_{fs}d^4$	Define the energy dissipated in multipath path loss
l	Define the bits of data send it by node to CH
K	Define the clusters numbers
E_{DA}	Define the data aggregation cost
$d(CHtoBS)$	Define the average distance of CH and BS
$d(Nto\ CH)$	Define the average distance of nodes and CH

Table 3 1: Parameters used in Energy Consumptions Formulas

3.2. Energy Harvesting in WSNs (EH-WSNs)

3.2.1. Problem Statements

One of the main problems that WSNs face is energy [32, 26,27, 33,34]. The main energy source used by the sensor nodes is battery power, but many problems are related to batteries:

- The leakages that consume the battery even if not in use.
- Extreme weather conditions may break down the batteries, resulting in chemical leakages that can cause various environmental problems [35].
- The battery's energy density is limited and that may hinder the sensor node operation over a long period of time [36].

The sensor nodes need continuous power feeding to work without interruption, such in active mode to transmit the data or inactive state when the sensor nodes switched off, the operation of a sensing node is made through four units: transceiver, processing, sensing and power units as shown in Figure 4.1.

- The transceiver unit permits the sensor node to interconnect with other sensor nodes.

- The processing unit allows control and data processing in the sensor node.
- The function of the sensing unit is to monitor the specific phenomena.
- The power unit is used to provide the system supply voltage, and it can be supported by an energy harvesting unit such as small-scale windmills or solar cells.

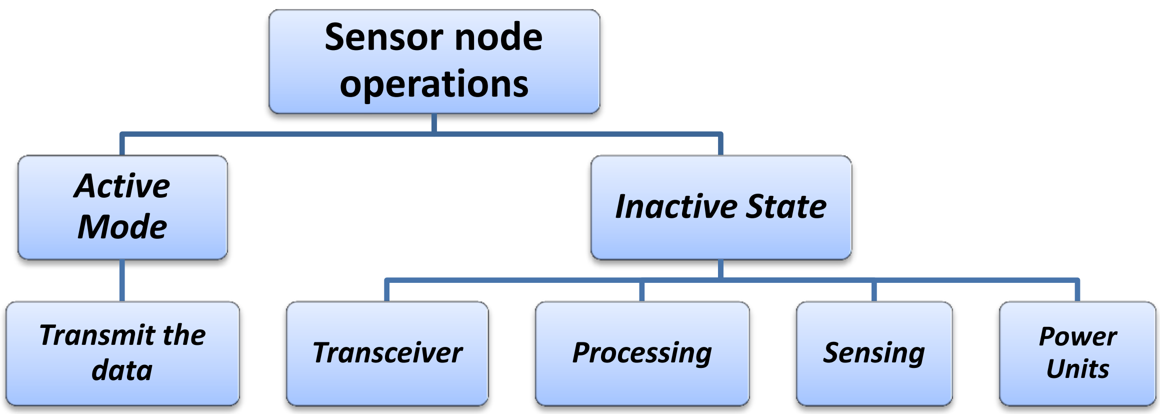

Figure 4. 1: Architecture of the sensor node.

When the sensor node is out of energy, it no longer fulfills its network role. So, we must replace the energy source (battery power) or introduce some harvesting mechanisms to bridge the energy gap.
The most basic comparison between Battery and Energy harvesting WSN are shown in Table 4.1.

Features	Battery	Energy Harvesting
Quality Of Service	As Low as possible or acceptable	As High as possible
Energy Source	Charge battery	Environment
Maintenance cost	High Cost ,Require frequent recharge or Replacement of battery	Low Cost, Self-sustaining
Requirement	Energy Efficient, Long Life battery	Energy neutral

Table 4. 1: Battery Vs Energy Harvesting

3.2.2. Categories of EH-WSNs

We can categorize two types of energy harvesting systems:

- Where ambient energy is directly converted to electrical energy to power the sensor nodes (no battery storage is required as depicted in Figure 4.2)
- Where the converted electrical energy is first stored before being supplied to the sensor node as shown in Figure 4.3.

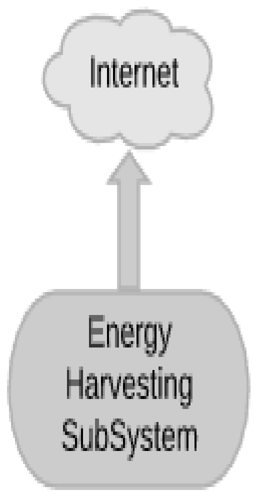

Figure 4. 2: EH Without Storage

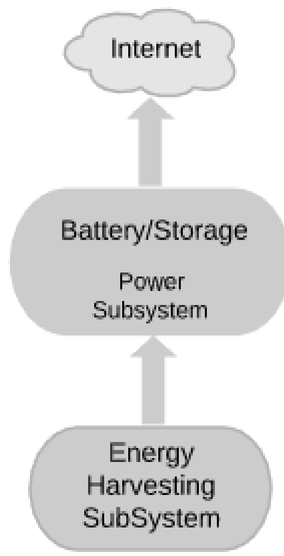

Figure 4. 3:EH With Storage

3.2.3. Classification of EH-WSNs

The energy harvesting sources are classified into two classifications:

- Ambient sources : are readily available in the environments at almost no cost
- External sources: are deployed explicitly in the environments for energy harvesting purposes

These classifications are subdivided as shown in Figure 4.4.

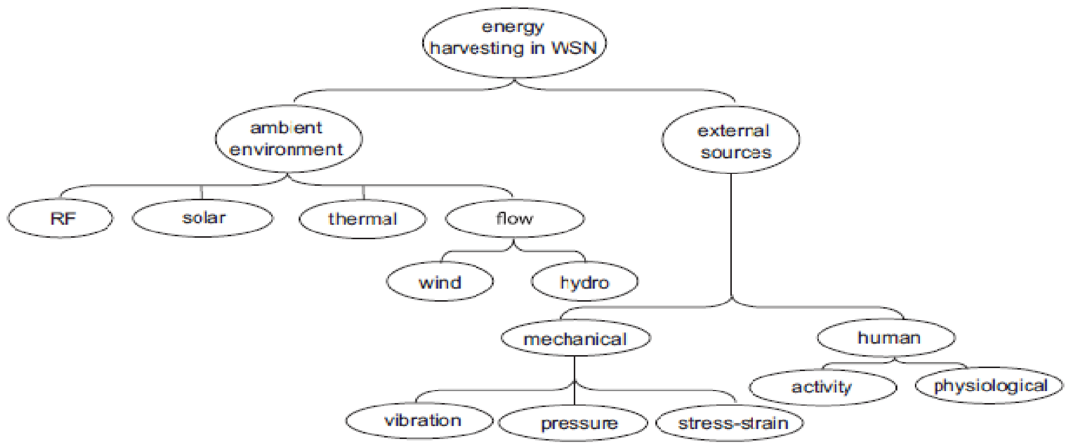

Figure 4. 4: Classification of EH sources.

The ambient sources are classified by four categories based on Energy harvesting:

- Radio Frequency : received radio waves.
- Solar: solar energy is an affordable and clean energy source that could eliminate the impending energy problem in WSNs.
- Thermal: converting heat energy into electrical energy.
- Flow:
  - Wind: It is also available free of charge and provides a good source of alternative energy (1200 mWh/day) [37].
  - Hydro: Harnessing water energy is the energy of motion and falling [38].

The External sources are classified by two categories based on Energy harvesting:

- Mechanical: Seeking to harvest energy from different sources such as vibrations, pressure, or stress-strain [39,40].
- Human: harvesting energy from the human used by the health care sector due to ensure that medical professionals can take appropriate and timely actions [41].
- Table 4.1 provides an overview of different energy sources and their corresponding characteristics.

Energy source		Predictable	Unpredictable	Controllable	Non-Controllable
Radio Frequency		●		●	
Solar		●			●
Thermal			●	●	
Flow	Wind	●			
	Hydro	●			●
Mechanical	Vibration		●	●	
	Pressure		●	●	
	Stress-strain		●	●	
Human	Activity		●	●	
	Physiological		●		●

Table 4.2: Characteristics of various energy sources.

3.2.4. Solar-Based Sources

The most source of energy harvesting used is the sun, while it is affordable [42], and it is uncontrollable [43], and we can convert it into electrical energy using photovoltaics [44]. While the amount of power generated by a cell change of the intensity of light, size, or performance. Photovoltaic cells are classified into three types of material [45,46] :

1. Mono crystalline (15–24% of effectiveness).
2. Polycrystalline (14–20.4% of effectiveness).
3. Thin film (8–13.2% of effectiveness).

Depending on the area, a photovoltaic system can produce power ranging from µW to MW. 15 mW/cm2 is the typical value of power density [47].

Figure 4.5 shows, the position of the sun in the sky comparable to the location on the Earth's specific surface:

1. by the solar altitude α (the angle between sun's position and the horizontal plane of the Earth's surface)
2. the solar azimuth β (the angle between a vertical plane incorporating the solar disk and a line running due north) [48].

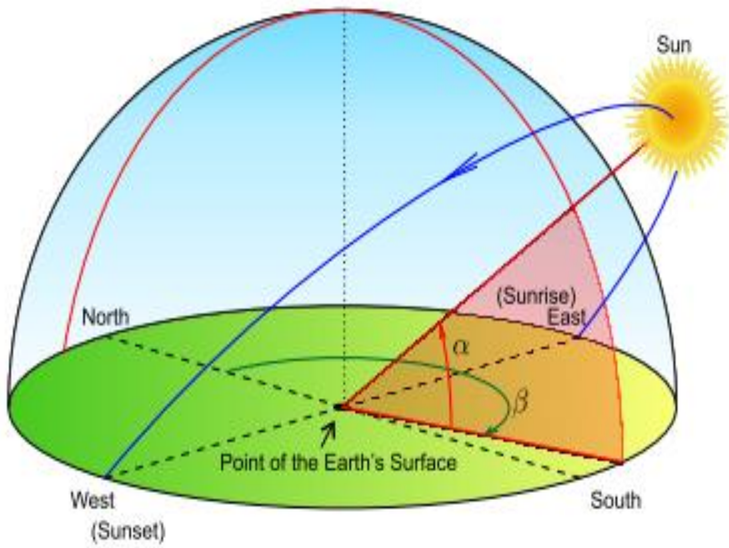

Figure 4. 5: Zone of the solar disk for maximum efficiency adopted from [48].

A system composed of a solar panel and lithium battery and the MPPT control circuit takes full advantage of solar energy to extend the life of the rechargeable battery [49, 50], some solar energy advantages are presented as follow:

- Solar energy is the external power source.
- Solar energy is unpolluted and does not harm the atmosphere.
- Highest achieved power density 10–15 mW/cm2, adapted from [53].
- Solar energy is flexible.
- Solar energy is free of cost and does not cause pollution.
- Solar energy systems need very small preservation and last for several years.

Usually, the energies received from solar panels are stored in batteries or supercapacitors.

3.2.4.1. Batteries

A battery is an electrochemical device, it has two types of primary (batteries non-rechargeable) or secondary (batteries are rechargeable), and however, a large-scale adoption would result in important environmental issues. Batteries are the most widely used energy storage technology for all electronic devices. To extend its lifetime, we would then have to increase the capacity of the conventional battery.

A battery is characterized by:

- Its voltage, expressed in volts (V), which represents the potential of oxidation-reduction between the two electrodes of the battery.

- Its electric charge in ampere-hour (Ah), which corresponds to the number of electrons that the battery can hold.
- Its electrical charging capacity represents the maximum charge provided by the battery.
- Its cyclability, expressed in the number of cycles, which characterizes the life of the battery.
- Its volume of energy density, expressed in watts per kilogram, which Define the battery life, and represents the amount of energy stored.

3.2.4.2. Supercapacitors

Supercapacitors are formed such as electrochemical double-layer capacitors (EDLCs), or pseudo capacitors [51]. They are differentiated by the high power density than the batteries.

Advantages of supercapacitors:

- Unlimited cycle life.
- Low impedance.
- rapid charging
- simple charge methods
- cost-effective energy storage

Table 4.2 illustrates the difference between rechargeable batteries and supercapacitors:

Batteries	**Supercapacitors**
Very high power during the charging	discharging cycle (up to 98%) and fast charging process
The recharging speed of the battery may be damaged due to a too fast charge.	less damaged
Thousands of charging	discharging cycles
Batteries do not work properly with a temperature below -10 _C.	tolerance at low temperatures of up to -40 _C

Table 4.3: Batteries VS Supercapacitors

3.3. Conclusion

In this chapter, we first started by introduce the main principles protocol based on Distributed Energy Efficient Clustering, then we describe the process of this protocol, later we introduced the concept of Energy Harvesting and their problems then the advantage of the using in WSNs. In the next chapter, we will depend on Energy Harvesting to produce a new algorithm for Distributed Energy Harvesting WSNs.

Chapter 4-Conduct of the research

- *Related Work*
- *Network model*
- *Energy Harvesting Model (EH-model)*
- *The proposed Algorithm*
  - *Setup Phase*
  - *Steady data transmission State phase*
- *Result of our Algorithm*
  - *Experiment 1: Number of alive nodes per time*
  - *Experiment 2: Number of packets made per time*
- *Conclusion*

Chapter 4

Conduct of the Research

4.1. Related Work

In the past decade, several researchers proposed a considerable number of algorithms that use clustering-based protocols to improve energy consumption. In this part, we introduce and describe the most relevant clustering protocols.

LEACH [8], the Low-Energy Adaptive Clustering Hierarchy is dynamic, where the clusters are randomly distributed. That means that every node can perform a Cluster Head (CH) function with a different probability. The position of cluster heads rotates between the various nodes to limit breakdown because of the energy shortage in any of the nodes. Whilst nodes are connected to the CHs with the smallest energy demanded to reach it. And in turn, the current cluster heads send the data in order through time slots allocation.

Another widely used algorithm, Energy-efficient distributed clustering algorithm HEED was proposed in [9], the cluster heads are chosen according to the remaining energy combination and their connection costs.

EECS [10], EEUC [11], EDUC [12] and EADUC [13] are coverage-aware and energy-efficient algorithms. Consequently, they focus on effective size of clusters factor, which is the distance from cluster heads to the base station. Such algorithms improve the distribution of energy throughout the network and extend the network lifetime.

A Centralized Balance Clustering (CBC) [15] protocol is proposed that is implemented in three steps. Initially, it computes the number of clusters according to the network conditions. Then, CH is selected for each block. In the final step, it is scheduled to send data while still avoiding any collision.

Later, a Hybrid Unequal Clustering with Layering Protocol (HUCL) was proposed in [14] that extends the network lifetime. This protocol is used to solve the problem of clustering overhead in case of dynamic clustering. HUCL first presents a simple compression algorithm to reduce the excess in data transmission and then proposes a mixture of static and dynamic clustering that greatly reduces the clustering overhead when compared to other dynamic aggregation techniques.

Amgoth et al. in [16] proposes an Energy-Aware Routing Algorithm (ERA) that forms clusters according to the level of energy in the CHs. Clusters are organized at different levels to build a virtual backbone of the routing data.

The study in [17] suggests an Energy and coverage-aware distributed clustering ECDC method that introduces coverage importance measures for the region and the whole coverage of points. ECDC increases the lifetime of the network by effecting these measures in calculating the waiting time and finding forwarding data the way to the sink.

Energy-Harvesting Stable Election Protocol (EH-SEP) [18] is based on the SEP algorithm introduced in [19]. EH-SEP is an energy harvested clustering protocol. The probability of newer nodes is greater than that of older nodes, and hence the remaining energy is dedicated to the cluster heads.

Clustering Routing Algorithm Based on Solar Energy Harvesting (CRBS) [20] is another method that uses both of the soft and hard thresholds to connect nodes to the network in the next round in case some nodes die. The main advantage of this protocol is that it increases the number of alive nodes and enhances the stability.

Later in [21], authors propose a Novel Energy Efficient Clustering (NEEC) protocol, which is implemented in a centralized and distributed manner. NEEC uses hybridization such as static and dynamic clustering.

In NEEC, the packets repaired in the Base Station are enhanced by supporting the consumed energy in the WSN.

S-LEACH presented in [24], is one of the most relevant studies conducted in WSN, which is based on energy harvesting sensor nodes from solar energy and they relies on a battery as a backup power source. In S-LEACH, the BS chooses the CHs and then these CHs select a new one.

In this thesis we are investigating the aggregation protocol for EH-WSN algorithms, EH-WSN algorithms must run as dynamically as possible to prevent energy harvesting sensors in the block mode from joining the network since they increase the workload and the lower productivity. This protocol aims are to resolve the energy constraints to extend the lifetime of the WSN, to raise the transmission rate and to decrease the network workload.

4.2.Network Model

Energy Harvesting in WSN is provided with Energy Harvesting nodes and a base station with unlimited network supplies. The data is taken by the sensor nodes and is sent to the base station. Some assumptions are made about the sensor nodes and the network model:

- “N” energy harvesting sensor nodes are distributed randomly in a (M* M) field.
- Nodes are homogeneous.
- Nodes are aware of their location
- The transmission power is detected according to the distance.
- CH can reach the base station in one hop or multiple hops.

4.3.Energy Harvesting Model

Energy harvesting (or energy scavenging) sensor nodes harvest the energy from the environment. The harvested energy uses a storage capacity defined by $E_{i,r-1}^{eh}$. However, the batteries have a limited amount of energy and they require a periodic charging replacement. In addition, renewable energy is not constant and it changes over time.

Based on the exponentially weighted moving average (EWMA) we used a forecast model for modeling energy harvested from sunlight [22, 23,24].

Formula (16), **calculates** the amount of energy model for an energy harvesting node *i*:

$$\mathbf{E_{rem}\ (i,\ r) = (min\ (E_{Max}\ (i),\ \ E_{rem}\ (i,\ r-1) +\ \ E_{EH}\ (i,\ r-1))} \quad (16)$$

Table 5.1 shows a description of the parameters used in formula (16).

Parameters	**Descriptions**
$\mathbf{E}_{rem\ (i,r\ )}$	Remaining energy in node i during round r.
$\mathbf{E}_{EH}\ (\ i, r\ \ -1)$	Energy harvested by node i during previous round $r-1$.
$\mathbf{E}_{\mathbf{Max}}(\ i\ )$	Maximum battery storage capacity in node i.
R	Initial energy in a node at the beginning of a round.

Table 5. 1: Parameters of Formula (16)

Formula (17), calculates the amount of energy harvesting:

$$\mathbf{E_{EH}(i, r - 1) = \mu_i \Delta t} \quad (17)$$

Formula (18), calculates the energy harvesting rate:

$$\mu_i = \textit{rand}\ (P_{h,Min}\ (r - 1),\ P_{h,Max}\ (r - 1)) \quad (18)$$

Table 5.2 shows a description of the parameters used in Formulas (17) and (18).

Parameters	Descriptions
μ_i	Is the Energy harvesting rate of node i during the previous round " $r-1$".
$P_{h,Min}(r - 1)$	Is the Probable maximum energy harvesting rate for all nodes during the previous round " $r-1$".
$P_{h,Max}(r - 1)$	Is the Probable minimum energy harvesting rate for all nodes during the previous round "$r-1$".
Δt	Is the round duration time.

Table 5. 2: Parameters of Formulas (17 & 18).

The Energy harvested in any node has a low and high threshold. Each node will be automatically blocked if the node's energy is below the low threshold and will not participate in the current round, or send data until the power state has reached the required level. That means that the node will continue to recharge the battery. When the energy capacity in the battery of the blocked node becomes above the low threshold, the node will switch to active mode. In other words, the node will be able join the network at the next setup stage and will begin sending and receiving data in the next round.

4.4. The Proposed Algorithm

In this section, we introduce a new protocol, named DEH-WSN (Energy Harvesting for Distributed Clustering wireless sensor networks). We consider that the sensor nodes know the information about their location depending on the frequency power.

In DEH-WSN, the selection of the cluster head is based on the following factors: (i) constant probability value, (ii) initial energy level, (iii) processing power and (iv) the amount of harvested energy. Moreover, some nodes are selected as cluster head based on their location.

The **first operation** of the base station after node deployment is employed as follows:

- BS transmits a mission to the network.
- Each node computes the distance to BS.
- All nodes send the location information to BS.
- In case a node is located at a distance less than d_0 to the base station, it will be placed in the first layer.
- All the base stations compute the farthest node from itself (d_N).

Empirically, there are four layers that data is transferred from the cluster heads to the base station.
Formula (19) calculates the length of each layer, whilst Table 6.1 Define their parameters:

$$d_N = d_{max} - d_{min} \qquad \& \qquad L_{RL} = \frac{d_N}{4} \tag{19}$$

Parameters	Descriptions
d_N	The difference between the farthest and closest nodes to base station
d_{max}	The distance between the furthest node and base station
d_{min}	The distance between the closest node and base station

Table 6. 1: Parameters of Formula (19).

The **second operation** of the base station is employed as follows: We calculate the cluster radius for every node using Formula (20), and defined in Table 6.2.

$$R_c(i) = \left[1 - c\frac{d_{max} - d_{i,BS}}{d_{max} - d_{min}}\right] RL_{max} \tag{20}$$

Parameters	Descriptions
$R_c(i)$	The radius of node **i**
RL_{max}	The maximum radius that enables a node to become a CH
$d_{i,BS}$	The distance between node **i** and BS.
C	Weighting factor $(0 \leq C \leq 1)$

Table 6. 2: Parameters of Formula (20).

The **third operation** of BS is employed as follows:

- Determine the location degree of each node where it's equal to the number of neighbor nodes "i".
- Based on LEACH, Here the both energy consumption per bit are used, the free space E_{fs} (d^n power loss) and the multipath fading E_{amp} (d^n power loss), they rely on the distance between the sender and receiver. Where

we use the free space model with **n=2**. If the distance is less than a threshold d_0. Otherwise, the multipath model is used where **n=4**.

- ***Formula (21)*** calculate the distance d_0, where E_{fs} and E_{mp} are energy consumption to run the transmitter amplifier according to LEACH protocol.

$$d_0 = \sqrt{\frac{E_{fs}}{Emp}} \tag{21}$$

This protocol is performed in a distributed manner and distributed into two rounds, where each one implemented in two steps: Setup State and Steady data transmission State as shown in Figure 5 and illustrated as follows:

- First step: Setup phase, the cluster heads are selected and the normal clusters are formed according to the algorithm discussed later, then a matching between energy harvested and cluster heads are performed.
- Second step: Steady data transmission state phase, the network schedule is divided into multiple rounds. In each one, cluster heads receive sensed data from the cluster member's nodes and collect data before transferring them to the base station.

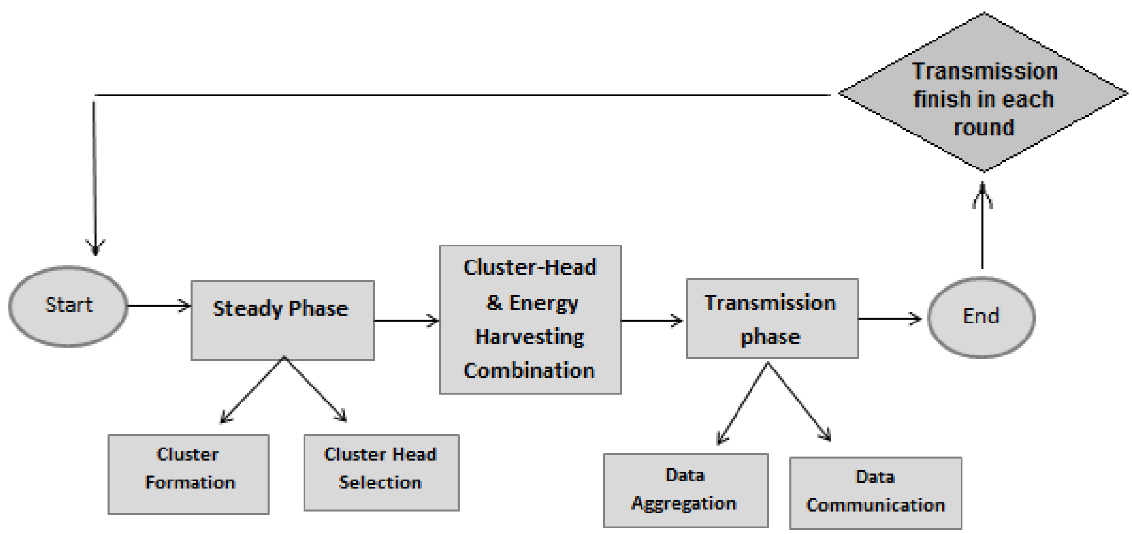

Figure 5: Flow chart describes our Protocol

4.4.1. Setup Phase

The implementation of this phase requires four sub-phases presented as follows:

Step 1: Calculation of delay time:

- According to ***Formula (22),*** the nodes calculate delay time, which help to choose the appropriate cluster head. The parameter functions are described in Table 7.1.

$$T_W\ (i, r) = \frac{E_{Max}\ (i)}{E_{rem}\ (i,r)} \times \frac{1}{\max\ (E_{EH}(i,r)\ ,\varepsilon)} \times \frac{1}{\max\ (|\ NL\ (i)\ |\ ,\varepsilon)\ )} \times d_{i,BS} \times \max\ (\alpha(i)\ ,\ \varepsilon) \times V_r \times T_2 \qquad (22)$$

Parameters	Descriptions
T_W	The delay of node i
Ε	A small number, if the parameter is zero, ε do not affect the equation.
| NL (i)|	The number of neighbors of node "i".
α(i):	The number of times that node " i" is selected as CH.
T_2	The time needed for the second phase.
V_r	A number between 0.1 and 0.2.

Table 7. 1: Parameters of Formula (22)

Step 2: Selection of a cluster head:

- The selection of cluster heads depends on the energy level and capacity of the harvested energy.
- All nodes must wait to finish the delay time.
- The node that has less delay time has more chance to be selected as a CH.
- If the node doesn't receive a message from the nearest neighbors, it declares itself as a CH.
- If the node receives CH massage, it couldn't be selected as a cluster head at all.
- In case the two nodes have similar delay times, the selected node should have a smaller ID

Step 3: Cluster formation:

- Each node that is selected as CH, forwards a message including the energy level to the non-cluster-head.
- The non-cluster-head follows the cluster head among the lowest energy needed to transmit data to them.
- A cluster head schedule nodes according to Time Division Multiple Access (TDMA). The further nodes must forward data as soon as possible and the further CH calculates the average energy.

- CH calculates the distance thresholds, where d_c is the closest, and d_f is the furthest nodes. The nodes that have a distance lower than d_c relay in the first layer and the nodes that have a distance greater than d_f are laid in the second layer.

Step 4: Route construction:

- Member nodes have the chance to turn to the sleep mode.
- Nodes of the first layer including nodes with distances smaller than d_0, from base station transmit the Route Request message into the network.
- Nodes of the second layer which received the message updated their routing tables, then started to transmit Route Request message to upper layers.
- The cluster heads who are in the first layer send data directly to the base stations.
- ***Formula (23)*** computes the cost to transfer data to BS.

$$relay_o(i) = E_{TX}^{CHi,BS}(i, l, d) \quad (23)$$

- ***Formula (24)*** calculates the evaluation parameter to transfer data to a base station, or to the central cluster head, Table 7.2 shows a description of the parameters used in formulas (23) and (24).

$$\begin{cases} E_{TX}^{CHi,CHj}(i,l,d) + E_{TX}^{CHi,next\ hopj}(j,l,d) + E_{RX}(j,1), & if\ E_{rem}(i,r) \geq E_{TX}^{CHi,CHj}(i,l,d)\ and\ E_{rem}(j,r) \\ & Inf \geq E_{TX}^{CHi,next\ hopj}(j,l,d) + (E_{RX}(j,1) * (M(j) + R(j) + 1)) \end{cases} \quad (24)$$

Parameters	Descriptions
$E_{TX}^{CHi,BS}$	The energy needed to send data from CH i to BS.
$E_{TX}^{CHi,CHj}(j,l,d)$	The energy needed to send data from CH i to CH j.
$E_{TX}^{CHi,next\ hopj}(j,l,d)$	The energy needed to send data from CH j to the next phase.
$E_{RX}(j,1)$	The energy needed to receive data in node j.
$M(j)$	The number of member nodes in CH j.
$R(j)$	The number of CHs, where node j acts as relay node and receive the CHs data.

Table 7. 2: Parameters of Formulas (23 & 24)

- Hence, CH is chosen as the cheapest relay cost; thereafter CHi sends a Route-Reply-message to the selected CH.
- Routing to the base station is made according to the route exposure phase without disruption during routing.

Algorithm 1 shows the setup phase pseudo-code.

```
SETUP PHASE OF DEH-WSN ALGORITHM
BEGIN
 IF (S[i].state = "CHP")
        exit
 ELSE
        S[i].state = node
        BEGIN
 (CT < TimePh1)
             V r = rand (0.1, 0.2)
             Compute the Delay Time
        ENDWHILE
        T = TimePh1 + Delay Time
 WHILE (CT < TimePh2)
        IF (CT > T)
             S [i].state = 'CH'
             Send Msg_Head
             Receive Msg_Head from CH
             Store in List_Head CHL[] along with distance
       ELSEIF (List_Head is received from any neighbor)
             S [i].state = 'CM'
             Store '[j]' in List_Head CHL[] along with distance
       ENDIF
 ENDWHILE
 WHILE (CT < TimePh3)
        IF (S [i].state = 'CM')
            Choose  the nearest CHS [j] from  CHL[] list
            S [i]. head = S [j]
            End Join Cluster Msg to S [j]
       ELSEIF ([i].state = CH)
            Receive Join Cluster Msg from CM
            Store in CM[] List
            Calculate the Ave Energy of Cluster & weak members
            Prepare and transmit TDMA to members of CM[]
            S[i].state = 'CHP'
       ENDIF
 ENDWHILE
 WHILE (CT < TimePh4)
            IF (Sensor[i].state = "CHP")
                WHILE (CT < Time Route Msg)
                     Wait and receive Route_Msg from CHs on based  Layering
                     Broadcast Route_Msg on based turn Layering
                     Store CHs information in Relay_CH List[]
            ENDIF
 Select the next hop CH from Relay_CH
 WHILE (CT < Time Route Replay)
               Wait for receive Route_Replay from CHs in higher Layer
               IF (receive Route_Replay)
                     Store CHs information in higher Layer
               ENDIF
ENDWHILE
Transmit  Route_Replay  to next hop in lower Layer
 ENDWHILE
 END
```

4.4.2. Steady data transmission State Phase

- The Carrier Sense Multiple Access (CSMA) method [25] is utilized for data transmission. Based on Time Division Multiple Access (TDMA) schedules, each node is in one cluster to send packets.
- First, nodes send all data to the cluster head.
- Then, Cluster head sends the packets to BS.
- At the end, the nodes which are in the first layer send to BS directly by default.

Algorithm 2 shows the transmission state pseudo code.

STEADY DATA TRANSMISSION STATE PHASE OF DEH ALGORITHM

```
BEGIN
 WHILE (CT < TimePh1)
  IF (S [i].state = "CM")
  WHILE  (CT < time of  transmitting  from TDMA )
     In case the Sensor is relay for CM farther by TDMA
     Receive Data Packet from CM and aggregation Data Packet
  ENDIF
  ENDWHILE
 Transmit Data Packet to BS or next hop from CM closer by TDMA
    ELSEIF  (S[i].state = "CH P")
    WHILE (CT < end of TDMA )
     Receive Data Packet from CM and aggregation Data Packet
    ENDIF
  ENDWHILE
  WHILE  (CT < TimePh2)
            IF ( S[i].state = "CH P")
 WHILE  (CT < time of transmit base Layering)
            In case Sensor be relay for CH further in higher Layer
            Receive Data Packet from CH and aggregation Data Packet
 ENDWHILE
 ENDIF
 Transmit Data Packet to BS or next hop
END
```

Table 7.3 shows a description of all the parameters used in the two algorithms.

Parameters	Descriptions
S	Define the Sensor Node.
CH	Define the Cluster Head.
CHP	Define the period that networks operational.
BS	Define the Base Station.
CHL [ ]	Define the List of Cluster Head.
CHS [ ]	Define the list of Sensor Node.
T	Define the delay of a node.
CT	Define the calculation the time of a cluster.

Vr	Define a random number between 0.1 & 0.2.
CM	Define the Cluster Member.
TimePh1	Define Time of the first phase.
TimePh2	Define Time of the second phase.
TimePh3	Define Time of the third phase.
TimePh4	Define Time of the fourth phase.

Table7.3 : Parameters used in Algorithms.

4.5. Simulation results

We assess the performance of the DEH-WSN protocol via simulations using MATLAB. We compared our proposed protocol (DEH-WSN) with respect to other NEEC [21] and HUCL [14].

Our Study evaluates:

- The network stability.
- The First Node Death (FND).
- The Half Node Death (HND).
- The number of alive nodes.
- The average energy during simulations.
- The throughput

We have simulated our WSN in a sensing field of (1km × 1km). In these simulations' experiments, two scenarios have been evaluated, as shown in Table 8.1. Other parameters are shown in Table 8.2.

The clusters closer to BS should be smaller, the cluster should have more strength to route higher-level packets to the base stations.

In our simulation, we assume that P_opt = 0.1 and C= 0.4 in Formula (20), and we set $\boldsymbol{RL_{max}}$ = 550 m where $\boldsymbol{RL_{max}}$ =1.25*550 (second and third layers) and $\boldsymbol{RL_{max}}$ =1.75*500 (fourth layer).

In the first scenario, 100 sensors nodes are distributed randomly in the area, A sink node is at the location of coordinates (x = 1500, y =500). In the second scenario, 50 sensor nodes are randomly distributed in the same simulation area with a sink node is at the location of coordinates (x = 500, y =500).

According to this data information, we presented a comparison between these two scenarios for the two experiments.

Scenarios	Base Station	Nodes Number	Network Space
Scenario 1	(1500,500)	100	1km ×1 Km
Scenario 2	(500,500)	50	1km ×1 Km

Table 8. 1: Scenarios used for the simulations.

Parameters	Symbols	Value
Energy depletion of the node's electronics circuit to transmit or receive the signal	Eelec	60nJ/bit
Energy depletion of the booster to deliver at a shorter distance	Efs	10nJ/bit/m2
Energy depletion of the booster to deliver at a longer distance	Eamp	0.0015pJ/bit/m4
Energy for data aggregation	EDA	5nJ/bit/signal
First energy of normal nodes	Eo (first energy of normal nodes)	0.5 J
Data packet size	-	5000 byte
Packet header size	-	25 byte
Control message size	-	50 byte
Lower threshold for energy harvesting	ETh_down	0.1 J
Upper threshold for energy harvesting	ETh_up	1 J
Probability rate of CH	P_opt	0.1

Table 8. 2: Simulation Parameters

After running the algorithm 50 times the simulation results are presented in two experiments.

4.5.1. Experiment 1: Number of alive nodes per time

The first experiment is about the number of alive nodes per time for the three different protocols. Figures 6.1 and 6.2 illustrate the number of alive nodes during simulations, seeking to calculate the average number of alive nodes shown in Table 9, these results demonstrate that DEH-WSN has a great performance since it raises the number of alive nodes 25% against HUCL protocol.

Protocol	Scenario 1	Scenario 2
DEH-WSN	80.6351	41.7261
NEEC [21]	80.6	40.9
HUCL [14]	60.5	33.5

Table 9: Rate of the average number of alive nodes.

.

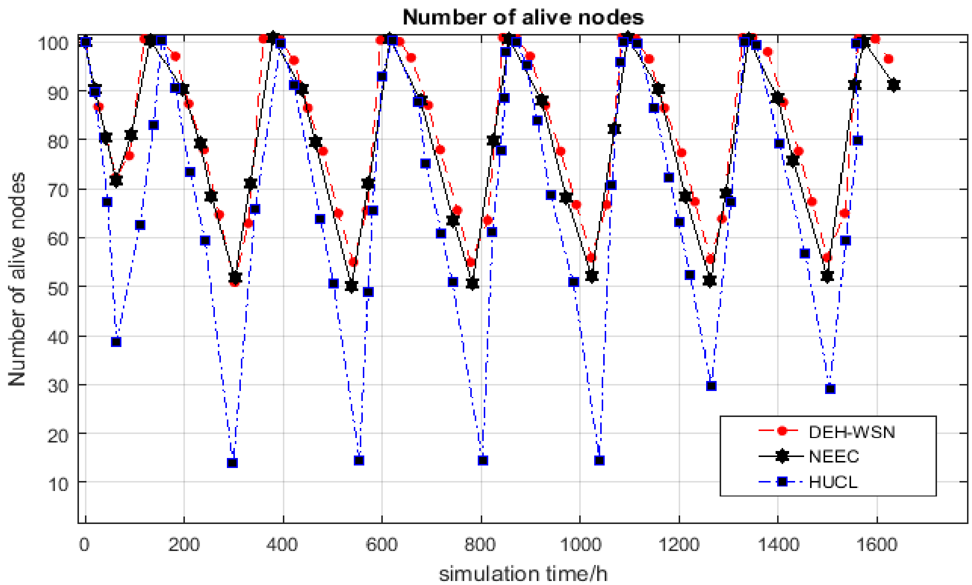

Figure 6. 1: Alive nodes numbers per time for Scenario 1

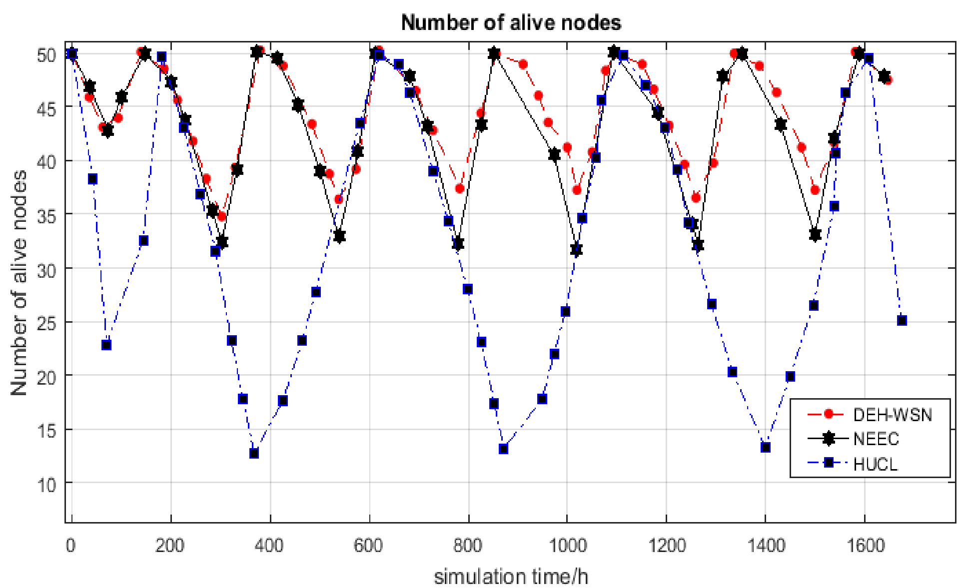

Figure 6. 2: Alive nodes numbers per time for Scenario 2

4.5.2. Experiment 2: Number of packets made per time

The second experiment, evaluate the number of packets made per time or throughput for the three differences protocols, Figure 7.1 and 7.2 shows numbers of each packet established during the two simulations, the results improve that DEH-WSN has throughput more improved and capable to transfer larger packets versus other protocols.

Table 10 shows the performance evaluation of protocols and the rate FND parameter and HND in the two scenarios. The performance result of the protocol shows that DEH-WSN effective for improving FND and HND and transfer more packets against other protocols. Stability is increased by changing the possibility of nodes to be CH according to the energy status of nodes and the amount of harvested energy. CH consumes more energy consumption balanced between nodes during the simulation. The results prove the efficiency DEH-WSN in increasing of HND against other protocol during the simulation.

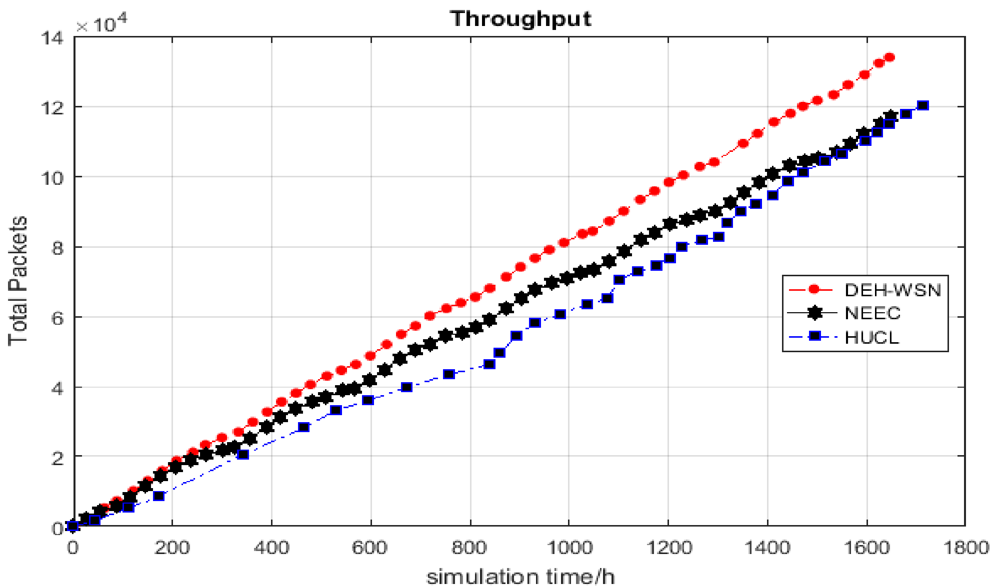

Figure 7. 1: Packets numbers made per time for Scenario 1.

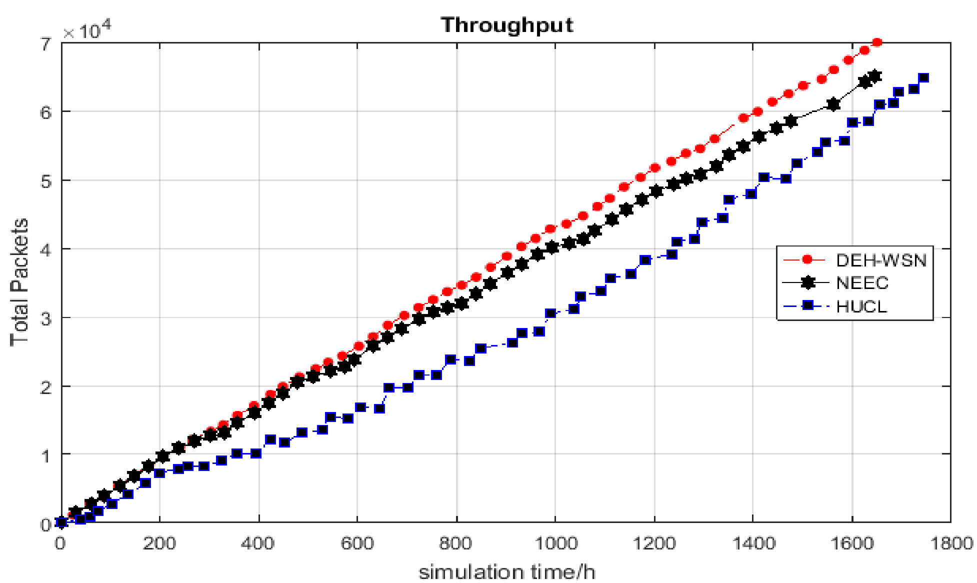

Figure 7. 2: Packets numbers made per time for scenario 2.

Scenario 1 Protocol	FND(100 Nodes)		HND(50 Nodes)	
	Time	Packets	Time	Packets
DEH-WSN	23 min	290.6	31 h: 23 min	2.6049e+04
NEEC [21]	30 min	254.9	21 h: 24 min	27163.3
HUCL [14]	24 min	124.2	8 h:30 min	3229.4

Scenario 2 Protocol	FND(100 Nodes)		HND(50 Nodes)	
	Time	Packets	Time	Packets
DEH-WSN	1 h:51 min	N	N	N
NEEC [21]	1 h:48 min	876.8	N	N
HUCL [14]	1 h:12 min	599.8	8 h:30 min	3431.6

Table 10: Average of FND and HND

4.6. Conclusion

In this chapter, we presented and evaluated a new routing protocol that relies on matching between clustering and energy harvesting in a distributed manner, which aims to maximize network lifetime and to become unlimited, by using energy harvesting instead of energy efficient Clustering protocols. Starting with these aims, and motivated by the various studies of WSN algorithm, we introduced the model network and the energy harvesting model to match between distributed Clustering and energy harvesting. Based on this approach, we proposed our new Distributed energy Harvesting Clustering algorithm “DEH-WSN”.

Moreover, in DEH-WSN, we use an estimation scheme to solve the average residual energy for the network that is recharged thanks to a uniform energy harvesting system. Our approach can be applied to the design of several types of wireless sensor network protocols that require network stability, since DEH-WSN can significantly improve the First Node Death (FND), the Half Node Death (HND), the number of alive nodes, the average energy, and the throughput. Furthermore, we presented two scenarios to evaluate this algorithm against other protocols. According to the simulation results, we demonstrate that the proposed algorithm balances the energy consumption, increases the number of available nodes and increases the number of repair packets in the BS.

Chapter 5-Conclusion

This research aims to develop and improve a new distributed clustering protocol for wireless sensor networks by matching the clustering with energy harvesting. The Clustering technique benefits from the energy storage and channel accessing to prolong the lifetime of a sensor network and reduce energy consumption. We suppose that all the nodes in a sensor network are provided by both distributed and multi-hop routing by taking into consideration the energy level, the amount of energy harvesting and neighbor numbers. Based on the two scenarios of the simulation, we prove by evaluation the superiority of our protocol over other methods in terms of the number of alive nodes, the throughput and network stability.

In future work, we can take a different energy consumption for cluster head selection method to a heterogeneous wireless sensor network.

Bibliography

[1] Wu Y , Liu W . Routing protocol based on genetic algorithm for energy harvesting-wireless sensor networks. IET Wireless Sens Syst 2013;3(2):112–18 .

[2] Xiao G , Tan H . Clustering algorithms for maximizing the lifetime of wireless sensor networks with energy-harvesting sensors. Comput Netw 2013;57(14):2689–704 .

[3] Ian F. Akyildiz and Mehmet Can Vuran. Wireless sensor networks, volume 4.John Wiley & Sons, 2010.

[4] Ian F. Akyildiz, Weilian Su, Yogesh Sankara subramaniam, and Erdal Cayirci.Wireless sensor networks: A survey. Computer networks, 38(4):393–422,2002.

[5] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. Computer networks, 52(12):2292–2330, 2008.

[6] Maher Rebai, Matthieu Le Berre, Hichem Snoussi, Faicel Hnaien, and Lyes Khoukhi. Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. Computers & Operations Research, 59:11–21, 2015.

[7] Wendi Beth Heinzelman. Application-specific protocol architectures for wireless networks. PhD thesis, Massachusetts Institute of Technology, 2000.

[8] Heinzelman WR , Chandrakasan A , Balakrishnan H . Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd annual Hawaii international conference on system sciences (HICSS), 3005-3014, Big Island, Hawaii, USA; 2000 .

[9] Younis Ossama, Fahmy Sonia. Heed: a hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks. IEEE Trans Mobile Comput 2004;3(4):660–9.

[10]Ye Mao, Li Chengfa, Chen Guihai, Wu Jie. EECS: an energy efficient clustering scheme in wireless sensor networks. Ad Hoc Sensor Wirel Netw 2006:99–119.

[11] Li Chengfa, Chen Guihai, Ye Mao, Wu Jie. An uneven cluster-based routing protocol for wireless sensor networks. Chin J Comput 2007;30(1):27–36.

[12] Yu Jiguo, Qi Yingying, Wang Guanghui. EDUC: an energy-driven unequal clustering protocol for heterogeneous wireless sensor networks. J ControlTheory Appl 2011;9(1):133–9.

[13] Yu Jiguo, Qi Yingying, Wang Guanghui, Guo Qiang, Gu Xin. EADUC: an energy-aware distributed unequal clustering protocol for wireless sensornetworks. Int J Distrib Sensor Netw 2011;2011 (Article ID 202145).

[14] Malathi L , Gnanamurthy RK , Chandrasekaran K . Energy efficient data collection through hybrid unequal clustering for wireless sensor networks. Comput Electr Eng 2015;48:358–70 .

[15] Chen J , Li Z , Kuo YH . A centralized balance clustering routing protocol for wireless sensor network. Wireless Pers Commun 2013;72(1):623–34 .

[16] Amgoth T , Jana PK . Energy-aware routing algorithm for wireless sensor networks. Comput Electr Eng 2015;41:357–67 .

[17] Gu X , Yu J , Yu D , Wang G , Lv Y . ECDC: an energy and coverage-aware distributed clustering protocol for wireless sensor networks. Comput Electr Eng 2014;40(2):384–98

[18] Xu X , Xiao M , Yan W . Clustering routing algorithm for heterogeneous WSN with energy harvesting. Appl Mech Mater 2015;733:734–9 .

[19] Xiao M , Zhang X , Dong Y . An effective routing protocol for energy harvesting wireless sensor networks. In: IEEE wireless communications and network conference (WCNC); 2013. p. 2080–4 .

[20] Yukun Y , Zhilong Y , Guan W . Clustering routing algorithm of self-energized wireless sensor networks based on solar energy harvesting. J China Univ Posts Telecommun 2015;22(4):66–73 .

[21] S.M. Bozorgi et al., A new clustering protocol for energy harvesting-wireless sensor networks, Computers and Electrical Engineering (2017), http://dx.doi.org/10.1016/j.compeleceng.2017.08.022.

[22] Kansal A , Hsu J , Zahedi S , Srivastava MB . Power management in energy harvesting sensor networks. ACM Trans Embedded Comput Syst 2007;6(4):1–32 .

[23] Yegnanarayanan V , Balas VE , Chitr G . On certain graph domination numbers and applications. Int J Adv Intell Paradigms 2014;6(2):122–35

[24] Li J , Liu D . DPSO-based clustering routing algorithm for energy harvesting wireless sensor networks. In: International conference on wireless communications & signal processing (WCSP); 2015. p. 1–5 .

[25] Yegnanarayanan V , Balas VE , Chitr G . On certain graph domination numbers and applications. Int J Adv Intell Paradigms 2014;6(2):122–35 .

[26] kyildizF, SuW, SankarasubramaniamY, CayirciE. Wirelesssensornetworks:asurvey. ComputNetw2002;38(4):393–422.

[27] ShaikhFK, ZeadallyS, SiddiquiF.Energyefficient routingin wireless sensor networks. In:Chilamkurti N ,ZeadallyS, ChaouchiH, editors.Next- generation

wireless technologies:4 Gand beyond, computer communications andnetworks.London:Springer;2013.p.131–57.

[28] Ali A, KhelilA, ShaikhF, SuriN. Efficient predictive monitoring of wireless sensor networks. IntJAutonAdapt Commun Syst 2012;5(3):233–54.

[29] Guang-HuiL,JunZ,ZhiW.Research on forest fire detection basedon wireless sensor network.In:Proceeding sof the world congress on intelligent control andautomation;2006.p.275–9.

[30] HartungC, HanR, SeielstadC, HolbrookS. FireWxNet:amultitieredp or- table wireless system for monitoring weather conditions in wildland fire environments. In :Proceeding sof the 4th international conference on mobile systems,applicationsandservices;2006.p.28–41.

[31] PenellaMT,AlbesaJ,GasullaM.Powering wireless sensor nodes: primary batteries versus energy harvesting. In:2009 IEEE instrumentation and measurement technologyconference,I2MTC'09.Singapore:IEEE;2009.p. 1625–30.

[32] Pantazis NA,Vergados DD. A survey on power control issues in wireless sensor networks.IEEECommunSurvTutor2007;9(4):86–107.

[33] ShaikhFK, ZeadallyS, ExpositoE. Enabling technologies for green internet ofthings.IEEESystJ2015;PP(99):1–12.

[34] KehH-C,WangY-H,LinK-Y,LinC-C.Powers a ving mechanism with optimal sleep control in wireless sensor networks.TamkangJSciEng2011;14 (3):235–43.

[35] Tiliute DE.Battery management in wireless sensor networks. ElectronElectr Eng—Kaunas: Technol2007;76(4):9–12.

[36]PenellaMT,AlbesaJ, Gasulla M.Powering wireless sensor nodes :primary batteries versus energy harvesting .In: 2009 IEEE instrumentationand measurement technologyconference,I2MTC'09.Singapore:IEEE;2009.p. 1625–30.

[37] SudevalayamS ,KulkarniP .Energy harvesting sensor nodes:surveyand implications. IEEECommunSurvTutor2011;13(3):443–61.

[38] Morais R,MatosSG, Fernandes MA,ValenteAL, SoaresSF, FerreiraP, ReisM. Sun, windandwater flow as energy supply for small stationary data acquisitionplatforms.ComputElectronAgric2008;64(2):120–32.

[39] StephenN. Onenergy harvesting from ambient vibration. JSoundVib 2006 ; 293 (1) : 409–25.

[40] Mathúna CÓ, ODonnellT, Martinez-CatalaRV, RohanJ, OFlynnB. Energy cavenging for long-term deploy able wireless sensor networks.Talanta 2008;75(3):613–23.

[41] Shenck NS, Paradiso JA.Energy scavenging with shoe-mounted piezo-electrics. IEEEMicro2001;21:30–42.

[42] Shad, R.; Steingart, D.; Frechette, L.; Wright, P.; Rabaey, J. Power Sources for Wireless Sensor Networks. In Wireless Sensor Networks; Karl, H., Wolisz, A., Willig, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 1–17

[43] Mousavi, S.M.; Mostafavi, E.S.; Jiao, P. Next generation prediction model for daily solar radiation on horizontal surface using a hybrid neural network and simulated annealing method. Energy Convers. Manag. 2017, 153, 671–682. [CrossRef] 9.

[44] Milichko, V.A.; Shalin, A.S.; Mukhin, I.S.; Kovrov, A.E.; Krasilin, A.A.; Vinogradov, A.V.; Belov, P.A.; Simovski, C.R. Solar photovoltaics: Current state and trends. Physics-Uspekhi 2016, 59, 727. [CrossRef].

[45] Akhtar, F.; Rehmani, M.H. Energy replenishment using renewable and traditional energy resources for sustainable wireless sensor networks: A review. Renew. Sustain. Energy Rev. 2015, 45, 769–784. [CrossRef] 12.

[46] Panatik, K.; Kamardin, K.; Shariff, S.; Yuhaniz, S.; Ahmad, N.; Yusop, O.; Ismail, S. Energy harvesting in wireless sensor networks: A survey. In Proceedings of the 2016 IEEE 3rd International Symposium on Telecommunication Technologies, Kuala Lumpur, Malaysia, 28–30 November 2016; pp. 53–58.

[47] Raghunathan, V.; Kansal, A.; Hsu, J.; Friedman, J.; Srivastava, M. Design considerations for solar energy harvesting wireless embedded systems. In Proceedings of the 2005 4th International Symposium on Information Processing in Sensor Networks, Boise, ID, USA, 15 April 2005; Volume 2005, pp. 457–462.

[48] Kelly, N.A.; Gibson, T.L. Increasing the solar photovoltaic energy capture on sunny and cloudy days. Sol. Energy 2011, 85, 111–125. [CrossRef]

[49] Li, Y.; Shi, R. An intelligent solar energy-harvesting system for wireless sensor networks. EURASIP J. Wirel. Commun. Netw. 2015, 2015, 179. [CrossRef] 23.

[50] Kim, S.; Chou, P.H. Energy harvesting by sweeping voltage-escalated charging of a reconfigurable supercapacitor array. In Proceedings of the 17th IEEE/ACM International Symposium on Low-Power Electronics and Design, Fukuoka, Japan, 1–3 August 2011; pp. 235–240

[51] Okonkwo, P.; Collins, E.; Okonkwo, E. Application of Biopolymer Composites in Super Capacitor. In Biopolymer Composites in Electronics; Elsevier: New York, NY, USA, 2017; pp. 487–503

[52] Li Qing, Qingxin Zhu, and Mingwen Wang. Design of a distributed energy efficient clustering algorithm for heterogeneous wireless sensor networks.Computer communications, 29(12):2230–2237, 2006.

[53]Sharma, AHaque, and Zainul A. Jaffery, Journal of Renewable and Sustainable Energy 10, 023704 (2018); doi: 10.1063/1.5006619

[54] N. Javaid, T. N. Qureshi, A. H. Khan, A. Iqbal, E. Akhtar, and M. Ishfaq, "EDDEEC: Enhanced Developed Distributed Energy-Efficient Clustering for Heterogeneous Wireless Sensor Networks," Procedia Computer Science, vol. 19, pp. 914–919, 2013

Appendix

We have used MATLAB to simulate DEH-WSN protocol.

I. EH-Simulation

```
mit=0;
hour=0;
day=0;
Eharwest = zeros(N_Nodes,RoundMax);
for hour_EH = 1 : RoundMax
mit = mit + 6;
if mit == 60
mit = 0;
hour = hour + 1;
if hour == 24
hour = 0;
day=day+1;
end
end
iArray = unique(randi([0,10001],N_Nodes*3/2,1))./10000;
if hour == 0 || hour == 1 || hour == 2 || hour == 3 || hour == 4 || hour == 5 ||  hour == 6 ||
hour == 7 || hour == 17 || hour == 18 || hour == 19 || hour == 20 || hour == 21 || hour ==
22 || hour == 23
for i = 1 : N_Nodes
Eharwest(i,hour_EH) = 0;
end
end

if  hour == 9
if mit >= 0 || mit < 6
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (25 - 0).* iArray(i);
end
elseif mit >= 6 || mit < 12
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (50 - 0).* iArray(i);
end
elseif mit >= 12 || mit < 18
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (75 - 0).* iArray(i);
end
elseif mit >= 18 || mit < 24
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (100 - 0).* iArray(i);
end
elseif mit >= 24 || mit < 30
for i = 1 : N_Nodes
```

```
Eharwest(i,hour_EH) =  0 + (125 - 0).* iArray(i);
end
elseif mit >= 30 || mit < 36
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (150 - 0).* iArray(i);
end
elseif mit >= 36 || mit < 42
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (175 - 0).* iArray(i);
end
elseif mit >= 42 || mit < 48
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (200 - 0).* iArray(i);
end
elseif mit >= 48 || mit < 54
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (225 - 0).* iArray(i);
end
elseif mit >= 54
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (250 - 0).* iArray(i);
end
end
end
if  hour == 10
if mit >= 0 || mit < 6
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (275 - 0).* iArray(i);
end
elseif mit >= 6 || mit < 12
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (300 - 0).* iArray(i);
end
elseif mit >= 12 || mit < 18
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (325 - 0).* iArray(i);
end
elseif mit >= 18 || mit < 24
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (350 - 0).* iArray(i);
end
elseif mit >= 24 || mit < 30
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (375 - 0).* iArray(i);
end
elseif mit >= 30 || mit < 36
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (400 - 0).* iArray(i);
end
elseif mit >= 36 || mit < 42
```

```
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (425 - 0).* iArray(i);
end
elseif mit >= 42 || mit < 48
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (450 - 0).* iArray(i);
end
elseif mit >= 48 || mit < 54
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (475 - 0).* iArray(i);
end
elseif mit >= 54
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (504 - 0).* iArray(i);
end
end
end
if hour == 11
if mit >= 0 || mit < 20
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (499 - 0).* iArray(i);
end
elseif mit >= 20 || mit < 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (494 - 0).* iArray(i);
end
elseif mit >= 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (489 - 0).* iArray(i);
end
end
end
if hour == 12
if mit >= 0 || mit < 20
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (484 - 0).* iArray(i);
end
elseif mit >= 20 || mit < 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (479 - 0).* iArray(i);
end
elseif mit >= 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (474 - 0).* iArray(i);
end
end
end
if hour == 13
if mit >= 0 || mit < 20
for i = 1 : N_Nodes
```

```
Eharwest(i,hour_EH) =  0 + (469 - 0).* iArray(i);
end
elseif mit >= 20 || mit < 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (464 - 0).* iArray(i);
end
elseif mit >= 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (459 - 0).* iArray(i);
end
end
end
if hour == 14
if mit >= 0 || mit < 20
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (454 - 0).* iArray(i);
end
elseif mit >= 20 || mit < 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (449 - 0).* iArray(i);
end
elseif mit >= 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (444 - 0).* iArray(i);
end
end
end
if hour == 15
if mit >= 0 || mit < 20
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (439 - 0).* iArray(i);
end
elseif mit >= 20 || mit < 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (434 - 0).* iArray(i);
end
elseif mit >= 40
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (430 - 0).* iArray(i);
end
end
end
if hour == 16
if mit >= 0 || mit < 6
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (387 - 0).* iArray(i);
end
elseif mit >= 6 || mit < 12
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (344 - 0).* iArray(i);
```

```
end
elseif mit >= 12 || mit < 18
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (301 - 0).* iArray(i);
end
elseif mit >= 18 || mit < 24
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (258 - 0).* iArray(i);
end
elseif mit >= 24 || mit < 30
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (215 - 0).* iArray(i);
end
elseif mit >= 30 || mit < 36
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (172 - 0).* iArray(i);
end
elseif mit >= 36 || mit < 42
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (129 - 0).* iArray(i);
end
elseif mit >= 42 || mit < 48
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (86 - 0).* iArray(i);
end
elseif mit >= 48 || mit < 54
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0 + (43 - 0).* iArray(i);
end
elseif mit >= 54
for i = 1 : N_Nodes
Eharwest(i,hour_EH) =  0;
end
end
end
end
Eharwest =  Eharwest ./10;
```

II. DEH-WSN-Simulation

```
Flag_First_DEad_Node=0;
Flag_Half_Node_DeadNode=0;
First_DEad_Node=0;
Half_Node_DeadNode=0;
Cl_Number=0;
Round_Temp=1;
Count_R=1;
Alive_Network = true;
Total_PacketsMade=0;
Total_PacketsLoss = 0;
Round_PacketsMade=0;
Round_PacketsLoss=0;
epsilon=0.00000001;
N-Position=Inf;
F-Position=0;
C_Alpha = 0.5;
P_opt =0.1;
mit=0;
hour=0;
day=0;
Avg_FNDead = 0;
AvgPackets_First_DEad_Node = 0;
Avg_HNDead = 0;
AvgPackets_HNDead = 0;
AVG_SimAliveNode = 0;
AVG_Alive(1:RoundMax) = 0;
AVG_Energy(1:RoundMax) = 0;
AVG_PacketsMadee(1:RoundMax) = 0;
AVG_PacketsLosss(1:RoundMax) = 0;
AVG_PacketsRoundMade(1:RoundMax)=0;
AVG_PacketsRoundLosss(1:RoundMax)=0;
N_Nodes = 100;
xm=1000;
ym=1000;
Sink.X=500;
Sink.Y=1500;
RoundMax=1680;
Eo=5;
Emax=5;
Size_DataPacket=4000;
Size_PacketHeader=200;
Control_msgSize=400;

ETX=50*0.000000001;
ERX=50*0.000000001;
Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;
```

```
EDA=5*0.000000001;

orignal_Distance=sqrt(Efs/Emp);
EHthreshold_DO = 0.1;
EHthreshold_up = 1;

for i=1:1:N_Nodes
S(i).X=rand(1,1)*xm;
S(i).Y=rand(1,1)*ym;
S(i).Energy=Eo;
end
MaxRound = 550;
for i=1:1:N_Nodes
S(i).ID=i;
S(i).Layer_ID=0;
S(i).Nb_IDLayer=0;
S(i).CH_ID=0;
S(i).NextHop_ID=0;
S(i).DelayTime=0;
S(i).DegreeNode=0;
S(i).Radius_Num=orignal_Distance;
S(i).C_AlphaH=0;
S(i).State='L';
S(i).Switch_Active='T';
S(i).CL_Number=0;
S(i).Relay_Num=0;
S(i).EnergyAvg_ClNumber=0;
S(i).EnergyFlag_ClNumber='F';
S(i).N_Distance=0;
S(i).Distance_To_Relay=0;
S(i).a = 0;
S(i).Distance_To_BS=sqrt( (S(i).X-(Sink.X) )^2 + (S(i).Y-(Sink.Y) )^2 );
if F-Position < S(i).Distance_To_BS
F-Position = S(i).Distance_To_BS;
end
if N-Position > S(i).Distance_To_BS
N-Position = S(i).Distance_To_BS;
end
end
Layer_Distance = (F-Position - N-Position)/4;
Layer1 = N-Position + Layer_Distance;
Layer2 = Layer1 + Layer_Distance;
Layer3 = Layer2 + Layer_Distance;
Layer4 = Layer3 + Layer_Distance;
for i=1:1:N_Nodes
S(i).Energy = S(i).Energy- ( (ERX )*Control_msgSize);
if S(i).Distance_To_BS > orignal_Distance
S(i).Energy = S(i).Energy- ( (ETX)*(Control_msgSize) + Emp*Control_msgSize*(
S(i).Distance_To_BS^4));
end
```

```
if S(i).Distance_To_BS <= orignal_Distance
S(i).Energy = S(i).Energy- ( (ETX)*(Control_msgSize) + Efs*Control_msgSize*(
S(i).Distance_To_BS^2));
end
if S(i).Distance_To_BS <= Layer1
S(i).Radius_Num = (1-(P_opt *( (F-Position-S(i).Distance_To_BS)/(F-Position-N-
Position ) ) ) )*MaxRound;
elseif S(i).Distance_To_BS <= Layer2 && S(i).Distance_To_BS > Layer1
S(i).Radius_Num = (1-(P_opt *( (F-Position-S(i).Distance_To_BS)/(F-Position-N-
Position ) ) ) )* ( MaxRound * 1.25 );
elseif S(i).Distance_To_BS <= Layer3 && S(i).Distance_To_BS > Layer2
S(i).Radius_Num = (1-(P_opt *( (F-Position-S(i).Distance_To_BS)/(F-Position-N-
Position ) ) ) )* ( MaxRound * 1.25 );
elseif S(i).Distance_To_BS <= Layer4 && S(i).Distance_To_BS > Layer3
S(i).Radius_Num = (1-(P_opt *( (F-Position-S(i).Distance_To_BS)/(F-Position-N-
Position ) ) ) )* ( MaxRound * 1.75 );
end
if S(i).Distance_To_BS <= Layer1 ||  S(i).Distance_To_BS <= orignal_Distance
S(i).Layer_ID = 1;
elseif S(i).Distance_To_BS <= Layer2 && S(i).Distance_To_BS > Layer1   &&
S(i).Distance_To_BS > orignal_Distance
S(i).Layer_ID = 2;
elseif S(i).Distance_To_BS <= Layer3 && S(i).Distance_To_BS > Layer2   &&
S(i).Distance_To_BS > orignal_Distance
S(i).Layer_ID = 3;
elseif S(i).Distance_To_BS <= Layer4 && S(i).Distance_To_BS > Layer3   &&
S(i).Distance_To_BS > orignal_Distance
S(i).Layer_ID = 4;
end
for j=1:1:N_Nodes
Distance_i_To_j=sqrt( (S(i).X-(S(j).X) )^2 + (S(i).Y-(S(j).Y) )^2 );
ifDistance_i_To_j <= S(i).Radius_Num && i~=j
S(i).DegreeNode = S(i).DegreeNode + 1;
end
end
end
while  Count_R <= RoundMax
Round_PacketsMade=0;
Round_PacketsLoss=0;
DeadNode=0;
for i=1:1:N_Nodes
if (S(i).Switch_Active=='F')
if S(i).Energy >= EHthreshold_up
S(i).Switch_Active='T';
S(i).State='L';
S(i).CH_ID=0;
S(i).NextHop_ID=0;
S(i).Distance_To_Relay=0;
S(i).DelayTime=0;
S(i).N_Distance=0;
```

```
S(i).CL_Number=0;
S(i).Relay_Num=0;
S(i).EnergyAvg_ClNumber=0;
S(i).EnergyFlag_ClNumber='F';
S(i).Nb_IDLayer=0;
elseif S(i).Energy < EHthreshold_up
DeadNode = DeadNode + 1;
S(i).Switch_Active='F';
end
end
if S(i).Switch_Active=='T'
if S(i).Energy <= EHthreshold_DO
DeadNode = DeadNode + 1;
S(i).Switch_Active='F';
elseif S(i).Energy > EHthreshold_DO
S(i).Switch_Active='T';
S(i).State='L';
S(i).CV = 'T';
S(i).CH_ID=0;
S(i).NextHop_ID=0;
S(i).Distance_To_Relay=0;
S(i).DelayTime=0;
S(i).N_Distance=0;
S(i).CL_Number=0;
S(i).Relay_Num=0;
S(i).EnergyAvg_ClNumber=0;
S(i).EnergyFlag_ClNumber='F';
S(i).Nb_IDLayer=0;
end
end
end
LveNode = N_Nodes - DeadNode;
if (DeadNode>=1) && (Flag_First_DEad_Node==0)
Flag_First_DEad_Node=1;
First_DEad_Node=Count_R
First_Node_Dead_Avg =First_Node_Dead_Avg + First_DEad_Node;
AvgPackets_First_DEad_Node = AvgPackets_First_DEad_Node +
Total_PacketsMade;
end
if (DeadNode>=N_Nodes./2) && (Flag_Half_Node_DeadNode==0)
Flag_Half_Node_DeadNode=1;
Half_Node_DeadNode=Count_R
Avg_HNDead = Avg_HNDead + Half_Node_DeadNode;
AVG_PacketsHalfNode_Dead = AvgPackets_HNDead + Total_PacketsMade;
end

for i=1:N_Nodes
if S(i).Switch_Active == 'T'
v_rand = 0.1 + rand*(0.2-0.1);
S(i).DelayTime = (Emax/S(i).Energy) * (1/(max(Eharwest(i,Count_R),epsilon)))  *
```

```
(1/(max(S(i).DegreeNode ,epsilon)))  * v_rand *
(max(S(i).C_AlphaH,epsilon))*((max(S(i).Distance_To_BS ,epsilon)));
elseif S(i).Switch_Active == 'F'
S(i).DelayTime=Inf;
end
end

for i=1:N_Nodes
table=[];
for j=1:N_Nodes
if(i~=j && S(i).Switch_Active == 'T' && S(j).Switch_Active == 'T' &&
S(i).Radius_Num >= sqrt( (S(i).X-(S(j).X) )^2 + (S(i).Y-(S(j).Y) )^2 ))
table=[table,S(j).DelayTime];
end
end
if(S(i).DelayTime < min(table))
S(i).State ='C';
end
end
second_Flag= true;
Waiting_Time(1:N_Nodes)=Inf;
while second_Flag
for i=1:1:N_Nodes
if (S(i).Switch_Active == 'T') && S(i).State == 'L'
flag_CH = 1;
for j=1:1:N_Nodes
if (S(j).Switch_Active == 'T') && S(j).State ~= 'N' && i~=j
Distance_i_To_j=sqrt( (S(i).X-(S(j).X) )^2 + (S(i).Y-(S(j).Y) )^2 );
ifDistance_i_To_j <= S(i).Radius_Num
        if S(j).DelayTime < S(i).DelayTime
                flag_CH = 0;
                if S(j).State == 'C'
                        S(i).State = 'N';
                        if Waiting_Time(i) > S(j).DelayTime
                                Waiting_Time(i)=S(j).DelayTime;
                                S(i).CH_ID=j;
                        end
                end
        end
end
end
end
if flag_CH == 1
S(i).State = 'C';
end
end
end
LNode = 0;
for i=1:1:N_Nodes
if S(i).Switch_Active=='T'&& S(i).State == 'L'
```

```
LNode = LNode + 1;
end
end
if LNode == 0
second_Flag = false;
end
end
Cl_Number=0;
for i=1:1:N_Nodes
if(S(i).Switch_Active=='T') &&  (S(i).State=='C')
Cl_Number=Cl_Number+1;
S(i).NextHop_ID=0;
S(i).CL_Number=0;
S(i).State='C';
S(i).EnergyAvg_ClNumber=0;
S(i).C_AlphaH = S(i).C_AlphaH + 1;
if S(i).Radius_Num <= orignal_Distance
S(i).Energy = S(i).Energy - ( (ETX)*(Control_msgSize) + Efs*Control_msgSize*(
S(i).Radius_Num^2));
elseif S(i).Radius_Num > orignal_Distance
S(i).Energy = S(i).Energy - ( (ETX)*(Control_msgSize) + Emp*Control_msgSize*(
S(i).Radius_Num^4));
end
for j=1:N_Nodes
if S(j).Switch_Active=='T' && S(j).State=='N'
Distance_i_To_j=sqrt( (S(i).X-(S(j).X) )^2 + (S(i).Y-(S(j).Y) )^2 );
ifDistance_i_To_j <= S(i).Radius_Num && i~=j
S(j).Energy = S(j).Energy- ( (ERX )*Control_msgSize);
end
end
end

end
end
for i=1:1:N_Nodes
if ( S(i).State=='N' && S(i).Switch_Active=='T' )
if  S(i).CH_ID ~= 0
Minumn_Distance=sqrt( (S(i).X-S(S(i).CH_ID).X)^2 + (S(i).Y-S(S(i).CH_ID).Y)^2 );
Minumn_Distance_Cl_Number=S(i).CH_ID;
S(i).CH_ID;
for c=1:1:N_Nodes
Distance_ij=sqrt( (S(i).X-S(c).X)^2 + (S(i).Y-S(c).Y)^2 );
if S(c).Switch_Active == 'T' && Distance_ij <= S(c).Radius_Num && S(c).State =='C'
temp=min(Minumn_Distance, Distance_ij );
if ( temp<Minumn_Distance )
        Minumn_Distance=temp;
        Minumn_Distance_Cl_Number=c;
end
end
end
```

```
if Minumn_Distance_Cl_Number ~= 0
S(Minumn_Distance_Cl_Number).CL_Number =
S(Minumn_Distance_Cl_Number).CL_Number+1;
S(i).CH_ID=Minumn_Distance_Cl_Number;
S(i).N_Distance=Minumn_Distance;
if Minumn_Distance>orignal_Distance
S(i).Energy = S(i).Energy- ( (ETX)*(Control_msgSize) + Emp*Control_msgSize*(
Minumn_Distance^4));
end
if Minumn_Distance<=orignal_Distance
S(i).Energy = S(i).Energy- ( (ETX)*(Control_msgSize) + Efs*Control_msgSize*(
Minumn_Distance^2));
end
if  Minumn_Distance_Cl_Number~=0
S(Minumn_Distance_Cl_Number).Energy =
S(Minumn_Distance_Cl_Number).Energy- ( (ERX )*Control_msgSize);
end
if (S(i).Energy<=0)
S(i).Switch_Active='F';
end
if (S(Minumn_Distance_Cl_Number).Energy<=0)
S(Minumn_Distance_Cl_Number).Switch_Active='F';
end
end
end
end
end
if(Cl_Number > 0)
for i=1:1:N_Nodes
if ( S(i).State=='C' && S(i).Switch_Active == 'T')
S(i).EnergyAvg_ClNumber=0;
S(i).a = 0;
Energy_a =0;
N-Position =0;
F-Position= 0;
for j=1:1:N_Nodes
if S(j).CH_ID == i && S(j).State=='N' && S(j).Switch_Active == 'T'
S(i).EnergyAvg_ClNumber = S(i).EnergyAvg_ClNumber + S(j).Energy;
if S(j).Energy > Energy_a
        Energy_a = S(j).Energy;
        S(i).a = j;
end
if F-Position < S(j).N_Distance
        F-Position = S(j).N_Distance;
end
if N-Position > S(j).N_Distance
        N-Position = S(j).N_Distance;
end
end
end
```

```
di_distance = (F-Position - N-Position)./2;
for j=1:1:N_Nodes
if S(j).CH_ID == i && S(j).State=='N' && S(j).Switch_Active == 'T'
if di_distance >= S(j).N_Distance
        S(j).Nb_IDLayer = 1;
else
        S(j).Nb_IDLayer = 2;
end
end
end
S(i).EnergyAvg_ClNumber = S(i).EnergyAvg_ClNumber ./ S(i).CL_Number ;
for j=1:1:N_Nodes
if S(j).CH_ID == i && S(j).State=='N' && S(j).Switch_Active == 'T' && S(j).Energy >
((1)* S(i).EnergyAvg_ClNumber)
S(j).EnergyFlag_ClNumber='T';
elseif S(j).CH_ID == i && S(j).State=='N' && S(j).Switch_Active == 'T' &&
S(j).Energy <= ((1)* S(i).EnergyAvg_ClNumber)
S(i).EnergyFlag_ClNumber='F';
end
end
end
end
end
for i=1:1:N_Nodes
if ( S(i).State=='N' && S(i).Switch_Active=='T' )
if  S(i).CH_ID ~= 0
Select_FlagHop= false;
Minumn_Distance=S(i).N_Distance;
if (S(i).Nb_IDLayer == 1)
S(i).NextHop_ID = S(i).CH_ID;
S(i).N_Distance;
S(i).CH_ID;
Select_FlagHop= true;
end
ifSelect_FlagHop== false
if S(i).N_Distance > orignal_Distance
EConsumption = ( (ETX)*(Size_DataPacket+Size_PacketHeader)  +
Emp*Size_DataPacket+Size_PacketHeader*( S(i).N_Distance ^4 ));
else
EConsumption = ( (ETX)*(Size_DataPacket+Size_PacketHeader)  +
Efs*Size_DataPacket+Size_PacketHeader*( S(i).N_Distance ^2 ));
end
S(i).NextHop_ID = S(i).CH_ID;
S(i).N_Distance;
S(i).CH_ID;
Select_FlagHop= true;
distance_k_To_i=0;
for k = 1:1:N_Nodes
if S(k).CH_ID == S(i).CH_ID && S(k).State=='N' && S(k).Switch_Active == 'T' &&
S(k).EnergyFlag_ClNumber=='T' && S(k).Nb_IDLayer == 1
```

```
        distance_i_To_Ch=sqrt( (S(i).X-S(S(i).CH_ID).X)^2 + (S(i).Y-
S(S(i).CH_ID).Y)^2 );
        distance_Kch=sqrt( (S(k).X-S(S(k).CH_ID).X)^2 + (S(k).Y-S(S(k).CH_ID).Y)^2
);
        if distance_i_To_Ch > distance_Kch
                distance_k_To_i=sqrt( (S(k).X-S(i).X)^2 + (S(k).Y-S(i).Y)^2 );
                if distance_k_To_i <= orignal_Distance
                        EConsumption = (
(ETX)*(Size_DataPacket+Size_PacketHeader)  +
Efs*Size_DataPacket+Size_PacketHeader*( distance_k_To_i ^2 ));
                elseif distance_k_To_i > orignal_Distance
                        EConsumption = (
(ETX)*(Size_DataPacket+Size_PacketHeader)  +
Emp*Size_DataPacket+Size_PacketHeader*( distance_k_To_i ^4 ));
                end
                if  EConsumption < EConsumption
                        EConsumption = EConsumption;
                        S(i).NextHop_ID = k;
                        S(i).N_Distance = distance_k_To_i;
                        S(i).CH_ID;
                  Select_FlagHop= true;
                end
        end
end
end
end
end
end
end
for i=1:1:N_Nodes
if( S(i).State=='C') && S(i).Switch_Active=='T'
if S(i).Radius_Num <= orignal_Distance
S(i).Energy = S(i).Energy - ( (ETX)*(Control_msgSize) + Efs*Control_msgSize*(
S(i).Radius_Num^2));
elseif S(i).Radius_Num > orignal_Distance
S(i).Energy = S(i).Energy - ( (ETX)*(Control_msgSize) + Emp*Control_msgSize*(
S(i).Radius_Num^4));
end
elseif ( S(i).State=='N' ) && S(i).Switch_Active=='T' && S(i).CH_ID ~= 0
S(i).Energy = S(i).Energy- ( (ERX )*Control_msgSize);
end
end
if(Cl_Number > 0)
Array_CH_BS(1:Cl_Number,1:2)=0;
Counter=0;
for i=1:1:N_Nodes
if(S(i).Switch_Active=='T') && (S(i).State=='C')
Counter = Counter+1;
Array_CH_BS(Counter,1)=i;
Array_CH_BS(Counter,2)=S(i).Distance_To_BS;
```

```
end
end
Array_CH_BS = sortrows(array_sort_CH_far_BS,2);
for SCounter=1:1:Cl_Number
if  (Cl_Number>0)
i =Array_CH_BS(SCounter,1);
if i >= 1 && i <= N_Nodes
if S(i).State=='C' && S(i).Switch_Active=='T'
if S(i).Layer_ID ~= 4

if (2*S(i).Radius_Num) <= orignal_Distance
S(i).Energy=S(i).Energy- ( (ETX)*(Control_msgSize)  + Efs*Control_msgSize*(
(2*S(i).Radius_Num) ^2 ));
end
if (2*S(i).Radius_Num) > orignal_Distance
S(i).Energy=S(i).Energy- ( (ETX)*(Control_msgSize)  + Emp*Control_msgSize*(
(2*S(i).Radius_Num) ^4 ));
end
if (S(i).Energy<=0)
S(i).Switch_Active='F';
end
end
if S(i).Layer_ID ~= 1
for j=1:N_Nodes
if S(j).Switch_Active=='T' && S(j).State=='C'
Distance_i_To_j=sqrt( (S(i).X-(S(j).X) )^2 + (S(i).Y-(S(j).Y) )^2 );
ifDistance_i_To_j >= 2*S(i).Radius_Num && i~=j && S(j).Layer_ID ~= 1
S(j).Energy = S(j).Energy- ( (ERX )*Control_msgSize);
end
if (S(j).Energy<=0)
S(j).Switch_Active='F';
                                end
                        end
                end
end
if S(i).Layer_ID == 1
        S(i).Relay=0;
        S(i).NextHop_ID=0;
        S(i).N_Distance = S(i).Distance_To_BS;
        S(i).Distance_To_Relay = S(i).Distance_To_BS;
else
flag=0;
relay=0;
if (S(i).Distance_To_BS>orignal_Distance)
   EC_Multihop= ( ETX*(Size_DataPacket+Size_PacketHeader) +
Emp*(Size_DataPacket+Size_PacketHeader)*( S(i).Distance_To_BS^4 ));
end
if (S(i).Distance_To_BS<=orignal_Distance)
   EC_Multihop= ( ETX*(Size_DataPacket+Size_PacketHeader) +
Efs*(Size_DataPacket+Size_PacketHeader)*( S(i).Distance_To_BS ^2));
```

```
end
for j=1:N_Nodes
if i~=j && S(j).State=='C' && S(j).Switch_Active=='T'
Distance_ij=sqrt( (S(i).X-(S(j).X) )^2 + (S(i).Y-(S(j).Y) )^2 );
if S(j).Layer_ID < S(i).Layer_ID
Energy_XRound=( (ERX + EDA)*(Size_DataPacket+Size_PacketHeader) );
        Energytemp_Xj =0;
        Energytemp_Xi =0;
        if (Distance_ij>orignal_Distance)
                Energytemp_Xi = ( ETX*(Size_DataPacket+Size_PacketHeader) +
Emp*(Size_DataPacket+Size_PacketHeader)*( Distance_ij^4 ));
        elseif (Distance_ij<=orignal_Distance)
                Energytemp_Xi = ( ETX*(Size_DataPacket+Size_PacketHeader) +
Efs*(Size_DataPacket+Size_PacketHeader)*( Distance_ij ^2));
        end
        if ( S(j).Distance_To_Relay >orignal_Distance)
                Energytemp_Xj =  ( ETX*(Size_DataPacket+Size_PacketHeader) +
Emp*(Size_DataPacket+Size_PacketHeader)*( S(j).Distance_To_Relay ^4 ));
        elseif ( S(j).Distance_To_Relay <=orignal_Distance)
                Energytemp_Xj =  ( ETX*(Size_DataPacket+Size_PacketHeader) +
Efs*(Size_DataPacket+Size_PacketHeader)*( S(j).Distance_To_Relay ^2 ));
        end
        if ((S(j).CL_Number + S(j).Relay_Num + 1)*ER_x) + Energytemp_Xj  <
S(j).Energy && Energytemp_Xi < S(i).Energy
                relay = Energytemp_Xi + Energytemp_Xj + ER_x;
                if relay < EC_Multihop
                        EC_Multihop=relay;
                        S(i).NextHop_ID=j;
                        S(i).N_Distance = Distance_ij;
                        S(i).Distance_To_Relay=Distance_ij;
                        flag = 1;
                end
        end
end
end
end
        if flag == 1
                if S(i).NextHop_ID ~=0
                        S(S(i).NextHop_ID).Relay_Num =
S(S(i).NextHop_ID).Relay_Num + 1;
                end
        elseif flag == 0
                S(i).Distance_To_Relay=S(i).Distance_To_BS;
                S(i).hop=0;
                flag = 1;
        end
end
end
end
end
```

```
end
end
Round_Temp=1;
while Round_Temp <= 10
Count_R;
DeadNode=0;
for i=1:1:N_Nodes
if (S(i).Switch_Active=='F')
DeadNode = DeadNode + 1;
S(i).Switch_Active='F';
end
if S(i).Switch_Active=='T'
if S(i).Energy <= EHthreshold_DO
DeadNode = DeadNode + 1;
S(i).Switch_Active='F';
elseif S(i).Energy > EHthreshold_DO
S(i).Switch_Active='T';
end
end
end
LveNode = N_Nodes-DeadNode;
if (DeadNode>=1) && (Flag_First_DEad_Node==0)
Flag_First_DEad_Node=1;
First_DEad_Node=Count_R
First_Node_Dead_Avg =First_Node_Dead_Avg + First_DEad_Node;
AvgPackets_First_DEad_Node = AvgPackets_First_DEad_Node +
Total_PacketsMade;
end
if (DeadNode>=N_Nodes./2) && (Flag_Half_Node_DeadNode==0)
Flag_Half_Node_DeadNode=1;
Half_Node_DeadNode=Count_R
Avg_HNDead = Avg_HNDead + Half_Node_DeadNode;
AVG_PacketsHalfNode_Dead = AvgPackets_HNDead + Total_PacketsMade;
end
for i=1:1:N_Nodes
if ( S(i).State=='N' && S(i).Switch_Active=='T' && S(i).CH_ID ~= 0 )
Minumn_Distance=S(i).N_Distance;
if (Minumn_Distance>orignal_Distance)
EConsumption =  ( ETX*(Size_DataPacket+Size_PacketHeader) +
Emp*(Size_DataPacket+Size_PacketHeader)*( Minumn_Distance^4 ));
end
if (Minumn_Distance<=orignal_Distance)
EConsumption =  ( ETX*(Size_DataPacket+Size_PacketHeader) +
Efs*(Size_DataPacket+Size_PacketHeader)*( Minumn_Distance ^2));
end
S(i).Energy = S(i).Energy - EConsumption;
if S(i).CH_ID == S(i).NextHop_ID
S(S(i).CH_ID).Energy = S(S(i).CH_ID).Energy- ( (ERX +
EDA)*(Size_DataPacket+Size_PacketHeader) );
elseif S(i).CH_ID ~= S(i).NextHop_ID
```

```
if (S(i).NextHop_ID ~= 0)
S(S(i).NextHop_ID).Energy = S(S(i).NextHop_ID).Energy- ( (ERX +
EDA)*(Size_DataPacket+Size_PacketHeader) );
end
end
if (S(S(i).NextHop_ID).Energy<=0) || (S(i).Energy<=0)
Total_PacketsLoss = Total_PacketsLoss + 1;
Round_PacketsLoss=Round_PacketsLoss+1;
end
Total_PacketsMade=Total_PacketsMade+1;
Round_PacketsMade=Round_PacketsMade+1;
elseif ( S(i).State=='N' && S(i).Switch_Active=='T' && S(i).CH_ID == 0)
S(i).CH_ID=0;
Minumn_Distance=S(i).Distance_To_BS;
if (Minumn_Distance>orignal_Distance)
EConsumption = ( ETX*((Size_DataPacket+Size_PacketHeader)) +
Emp*(Size_DataPacket+Size_PacketHeader)*( Minumn_Distance ^ 4 ));
end
if (Minumn_Distance<=orignal_Distance)
EConsumption = ( ETX*((Size_DataPacket+Size_PacketHeader)) +
Efs*(Size_DataPacket+Size_PacketHeader)*( Minumn_Distance ^ 2 ));
end
S(i).Energy = S(i).Energy - EConsumption;
if S(i).Energy < 0
Total_PacketsLoss = Total_PacketsLoss + 1;
Round_PacketsLoss=Round_PacketsLoss+1;
end
Total_PacketsMade=Total_PacketsMade+1;
Round_PacketsMade=Round_PacketsMade+1;
end
end
for i=1:1:N_Nodes
if S(i).State=='C' && S(i).Switch_Active=='T'  && Cl_Number > 0
if S(i).NextHop_ID ~=0
if S(i).Distance_To_Relay <= orignal_Distance
EConsumption =  ( (ETX)*((Size_DataPacket+Size_PacketHeader))  +
Efs*(Size_DataPacket+Size_PacketHeader)*( S(i).Distance_To_Relay ^2 ));
end
if S(i).Distance_To_Relay > orignal_Distance
EConsumption = ( (ETX)*((Size_DataPacket+Size_PacketHeader))  +
Emp*(Size_DataPacket+Size_PacketHeader)*( S(i).Distance_To_Relay ^4 ));
end
S(i).Energy = S(i).Energy - EConsumption;
S(S(i).NextHop_ID).Energy = S(S(i).NextHop_ID).Energy- ( (ERX +
EDA)*(Size_DataPacket+Size_PacketHeader) );
if (S(S(i).NextHop_ID).Energy<=0) || (S(i).Energy<=0)
Total_PacketsLoss = Total_PacketsLoss + 1;
Round_PacketsLoss=Round_PacketsLoss+1;
end
Total_PacketsMade=Total_PacketsMade+1;
```

```
Round_PacketsMade=Round_PacketsMade+1;

elseif S(i).NextHop_ID ==0
if S(i).Distance_To_BS <= orignal_Distance
EConsumption = ( (ETX)*((Size_DataPacket+Size_PacketHeader))  +
Efs*(Size_DataPacket+Size_PacketHeader)*( S(i).Distance_To_BS ^2 ));
end
if S(i).Distance_To_BS > orignal_Distance
EConsumption = ( (ETX)*((Size_DataPacket+Size_PacketHeader))  +
Emp*(Size_DataPacket+Size_PacketHeader)*( S(i).Distance_To_BS ^4 ));
end
S(i).Energy = S(i).Energy - EConsumption;
if S(i).Energy < 0
Total_PacketsLoss = Total_PacketsLoss + 1;
Round_PacketsLoss=Round_PacketsLoss+1;
end
Total_PacketsMade=Total_PacketsMade+1;
Round_PacketsMade=Round_PacketsMade+1;
end
end
end
mit = mit + 6;
if mit == 60
mit = 0;
hour = hour + 1;
if hour == 24
hour = 0;
day=day+1;
end
end
for i = 1:1: N_Nodes
Eeh=Eharwest(i,Count_R);
if S(i).Energy < 0
S(i).Energy =0;
end
S(i).Energy= min(Emax,S(i).Energy + Eeh);
end
sumEnergy=0;
counter_node=0;
for i=1:1:N_Nodes
if(S(i).Switch_Active=='T')
sumEnergy=sumEnergy+S(i).Energy;
counter_node=counter_node+1;
end
end
E_AVG=sumEnergy/counter_node;
AVG_SimAliveNode=AVG_SimAliveNode+LiveSim;
AVG_Alive(Count_R) = AVG_Alive(Count_R) + LiveSim;
AVG_Energy(Count_R) = AVG_Energy(Count_R) + E_AVG;
AVG_PacketsMadee(Count_R) = AVG_PacketsMadee(Count_R) +
```

```
Total_PacketsMade;
AVG_PacketsRound(Count_R)=AVG_PacketsRound(Count_R)+
Round_PacketsMade;
AVG_PacketsRoundLosss(Count_R)=AVG_PacketsRoundLosss(Count_R)+
Round_PacketsLoss;
Count_R = Count_R + 1;
Round_Temp = Round_Temp + 1;
end
end
if Flag_First_DEad_Node ==0
disp('Error cz First node dead event has not happened in this simulation.')
First_DEad_Node=Inf;
First_Node_Dead_Avg =First_Node_Dead_Avg + First_DEad_Node;
AvgPackets_First_DEad_Node = AvgPackets_First_DEad_Node + Inf;
end
if Flag_Half_Node_DeadNode ==0
disp('Error CZ Half Node Dead event has not happened in this simulation.')
Half_Node_DeadNode=Inf;
Avg_HNDead = Avg_HNDead + Half_Node_DeadNode;
AVG_PacketsHalfNode_Dead = AvgPackets_HNDead + Inf;
end
```

III. MainSimulation

```
clear
close all
clc
total_sim = 50;
EH-Simulation
for test = 1 : total_sim
    test
    SetupF
        EH-Simulation
    DEH-WSN-Simulation
end
clc
Avg_FNDead =First_Node_Dead_Avg / total_sim
S1=S_Nb_2(fix(Avg_FNDead ./10));
S2=S_Nb_2(fix(mod(Avg_FNDead,10)*6));
Stext=strcat('FirstNodeDead Avg is >> hour :: ',S1,'  min :: ',S2,' ');
disp(Stext);
AvgPackets_First_DEad_Node = AvgPackets_First_DEad_Node / total_sim
Avg_HNDead = Avg_HNDead / total_sim
S1=S_Nb_2(fix(Avg_HNDead ./10));
S2=S_Nb_2(fix(mod(Avg_HNDead,10)*6));
Stext=strcat('HalfNodeDead Avg is >> hour :: ',S1,'  min :: ',S2,' ');
disp(Stext);
AvgPackets_HNDead = AvgPackets_HNDead / total_sim
AVG_SimAliveNode = ((AVG_SimAliveNode  / total_sim)/RoundMax-1)
```

```
figure(1)
grid
title('Number of alive nodes')
ylabel('Number of alive nodes')
xlabel('simulation time/h')
hold on
for r=1:30:RoundMax-30
    plot([r r+30],[AVG_Alive(r)/total_sim AVG_Alive(r+30)/total_sim],'k','LineWidth',1.5);
end
hold on
for r=1:30:RoundMax-30
    plot(r,AVG_Alive(r)/total_sim,'ks','LineWidth',1.5);
end
axis([0 RoundMax-1 0 N_Nodes]);
figure(2)
grid
title('Throughput')
ylabel('Throughput');
xlabel('simulation time/h')
hold on
for r=1:30:RoundMax-30
    plot([r r+30],[AVG_PacketsMadee(r)/total_sim
AVG_PacketsMadee(r+30)/total_sim],'k','LineWidth',1.5);
end
hold on
for r=1:30:RoundMax-30
    plot(r,AVG_PacketsMadee(r)/total_sim,'ks','LineWidth',1.5);
end
AVG_Alive(1:1:RoundMax-60)=AVG_Alive(1:1:RoundMax-60)/total_sim;
AVG_Energy(1:1:RoundMax-60)=AVG_Energy(1:1:RoundMax-60)/total_sim;
AVG_PacketsMadee(1:1:RoundMax-60)=AVG_PacketsMadee(1:1:RoundMax-
60)/total_sim;
AVG_PacketsLosss(1:1:RoundMax-60)=AVG_PacketsLosss(1:1:RoundMax-
60)/total_sim;
AVG_PacketsRound(1:1:RoundMax-60)=AVG_PacketsRound(1:1:RoundMax-
60)/total_sim;
AVG_PacketsRoundLosss(1:1:RoundMax-
60)=AVG_PacketsRoundLosss(1:1:RoundMax-60)/total_sim;
```