Chapter One

Introduction

1.1. General Overview

A Wireless Sensor Network (WSN) is a group of sensors communicate with each other by wireless connection. It is used to screen a physical event by gathering and deliver information to the involved celebration, gives an administrator the ability to implement, view, and react to events and phenomena in a specific environment. WSNs may consist of many different types of sensors including seismic, charismatic, thermal, chart, infrared, sound, and radar, which are able to monitor a wide variety of ambient situation[1, 2]. As a result, Military surveillance, home health care or assisted living, and environmental science are three major application that used WSN . also the agricultural applications ,security, urban warfare ,ground based monitoring of land and water, health-care, surveillance ,assisted and enhanced-living scenarios, environmental monitoring, weather prediction, battlefield monitoring and exploration of the Solar System, and many other fields [3]. In WSN sensor nodes monitor the environment and send the interpretation to a BS as shown in Figure 1.1.

Sensor node can communicate with other nodes and with the base station of the network by wireless connection. The size of the network can be small (a few meters) or large (hundreds of kilometers) depending on the application in use.

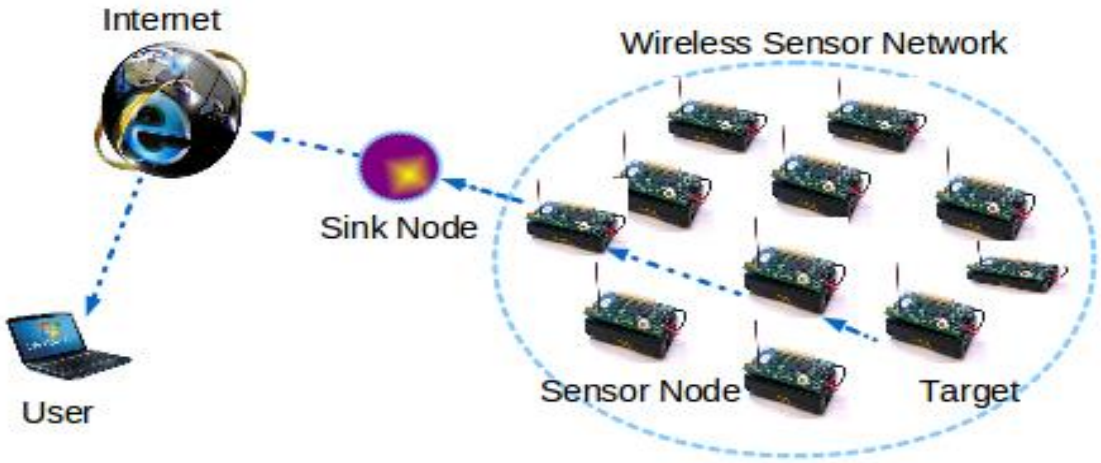

Figure 1.1: Wireless sensor network

Energy efficiency is very important factor in wireless sensor networks. In wireless communication, data broadcast consumes more power than data processing. As the size of data or the distance between communicators increases, the power consumption increases. In order to reduce the data size and the distance between communicators, some routing protocols are designed[4]. These designed protocols uses a hierarchical technique and clustering the nodes this save more energy because the cluster heads can do some aggregation and reduction ,Each one of these protocols have a different routing method[5]. This study analyzes the performance of the most used protocols: LEACH (Low Energy Adaptive Clustering Hierarchy), and PEGASIS (Power Efficient Gathering in Sensor Information System) and TEEN (Threshold sensitive Energy Efficient sensor Network).

1.2. Problem Statement

Though different protocols are available, each of them are based on different assumptions and intuitions. Very little information is known about predicting performance of the three protocols (LEACH, PEGASIS, TEEN) and no attempt has previously been made to directly compare them in a realistic manner. Because WSNs is one of new fields of research, there are no many studies and researches in this domain. Yet there are a few studies related to the subject of this study.

1.3. Definition of the Problem

In its simplest form, the wireless sensor network consists of number of sensor nodes and a base station. These sensor nodes perform three functions: sensing, processing, and communicating. To do these functions, the sensor nodes need a source of sufficient and sustained energy. To supply it with energy, every sensor node has a battery. When this battery dies, the sensor is useless. Because of that, the power consumption is one of the most important issues during designing of wireless sensor networks[6].

In wireless communication, data transmission consumes more power than data processing. The battery power of the node will be reduced whenever it transmits more amounts of data. Also the power consumption increases as the distance between the two parties of the connection increases. So, to reduce power consumption of the network, the size of data transmitted and the distance between communicators should be reduced.

In [7], the thesis intended to develop a methodology for WSN routing protocols to increase the sensor lifetime and decrease the energy consumption rate of wireless sensor nodes. It planned an approach to reflect on the energy level of sensor nodes through the establishment and routing processes, and distribute the power consumption among the wireless sensor nodes.

In [8], the main issue was increase the network lifetime of WSNs by reducing energy consumption during routing process from hop number point of view. This thesis proposes a Hop-based Energy Aware Routing (HEAR) algorithm. And showed that the proposed protocol (HEAR) has a better performance than other algorithms.

In [9],the leach protocol to improve the wireless sensor network performance has expanded. The main objective of the research is to investigate the mechanism to conserve and balance the energy consumption in wireless sensor networks. The research proposed a Fixed-LEACH routing technique and it has claimed that the proposed technique reduces the consumption of power among the network.

An energy efficient hierarchical routing technique has proposed in [10].In this technique, the cluster nodes are selected based on the prediction of transmission energy via a shortest path to the base station. The study found that the proposed protocol offer a better result to energy efficiency comparing to LEACH and other non-hierarchical techniques.

In [11],the thesis compared between hierarchical (TEEN) and non-hierarchical routing protocols (Directed Diffusion & Flooding) of WSNs

when using moving sink. According to the study, Directed Diffusion has the best performance, TEEN came in second and Flooding came in third.

An approach to minimize the energy consumption in WSNs and increase network's lifetime in heterogeneous environments using protocol named ARESEP was proposed in [12].Unlike protocols such as LEACH, M-GEAR, and SEP, in this approach, nodes are selected as cluster heads based on their residual energy and system average energy.

A hierarchical based routing protocol which improves the scalability and increases the network lifetime of the wireless sensor network by distribute the power dissipation load evenly among all the sensor nodes within the network was in presented [13]. In [14], the authors have considered different clustering based energy efficient routing protocols of wireless sensor networks. They focused on energy efficient clustering based routing protocols for wireless sensor networks. Recently, a review on network structure based routing protocol in WSNs was covered in [15], where energy consumption and network life time has been considered as the major issues.

1.4. Objectives

The objectives of this study are:

1. Studying the performance of three of hierarchical routing protocols (LEACH, PEGASIS, TEEN) according to the following factors:
    - End to end delay.
    - Load in the network.
    - Amount of traffic received by the sink (base station).

- Throughput.

2. Determining the strength and weakness of each protocol.
3. Comparing between protocols to determine which is better according to each factor.

1.5. Methodology

In this research, three hierarchical routing protocols of wireless sensor networks are studied. Each one of these protocols has different attributes and different way of work. To analyze the performance of these protocols and to compare between them, the following steps were performed:

1- Study the attributes of each protocol.
2- Perform a simulation for each protocol by OPNET 14.5 modeler.
3- Analyze the information gathered from simulation and obtains results.
4- Compare between the performances of the three protocols, and draw findings.

1.6. Thesis Layout

Chapter two covers the theoretical background of the study. It describes the architecture, the applications, and the standers of WSNs, and also describes the hierarchical routing protocols of WSNs. Chapter Three discusses the configuration of the simulation performed in the study, illustrating the comparing factors used to study the performance of the protocols . The results obtained from the simulation, the discussion and the analysis of these results are covered in Chapter Four. Chapter Five contains the conclusion, recommendations, and further researches.

Chapter Two

Routing Protocols In Wireless Sensor Networks

2.1 Overview of Wireless Sensor Networks

A sensor network are used to exchange information between an application platform and one or more sensor nodes, it's an infrastructure comprised of sensing ,measuring, computing, The administrator typically is a social, commercial, industrial entity or governmental. The environment can be biological system or physical world, a or an information technology (IT) framework[16, 17].

There are four essential components in a sensor network: (1) number of nodes which it an assembly of distributed or restricted sensors; (2) an interconnecting network, (3)a sink or base station which it's a central point of information (4) a software which its set of computing resources to handle data correlation and analyses [18].

2.2 WSN Architecture and Protocol Stack

The sensor node has the ability to collect data and route data back to the sink/gateway and the end-users. Data are routed back to the end-user by a multi-hop infrastructure less architecture through the sink as shown in Figure 2.1[19]. The sink may speak with the task manager/end-user via the Internet or satellite or any type of wireless network such as Wi-Fi, mesh networks, cellular systems, and WiMax, or without any of these networks where the sink can be directly connected to the end-users[20]. Note that there may be multiple base stations and multiple end-users in the architecture In WSNs,

the sensor play two roles in the network data originators and data routers. Hence, communication is performed for two reasons:

- Source function: Source nodes with event information perform communication in order to send their packets to the base station .
- Router function: Sensor nodes also contribute in forward the packets received from other nodes to the next hop in the multi-hop path to the BS.

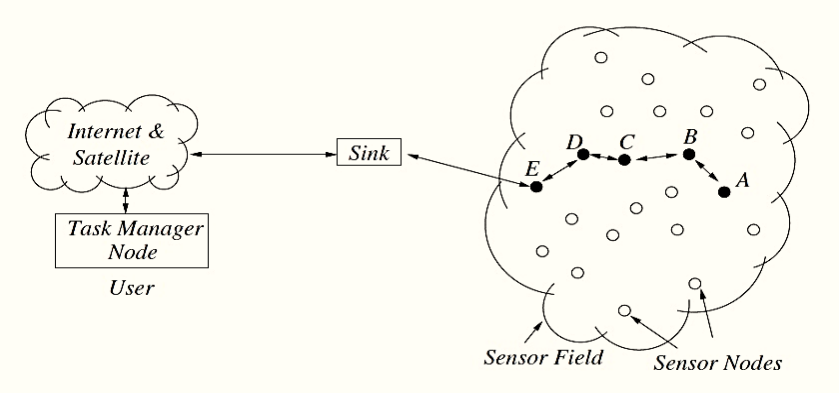

Figure 2.1: Data flow in wireless sensor networks

The protocol stack used by the wireless sensor network devices include sink and nodes combines routing awareness and power, integrates data with networking protocols, promotes cooperative efforts of sensor nodes and communicates power efficiently through the wireless medium[21, 22]. in Figure 2.2, the protocol layers consists of the physical layer, data link layer, network layer, transport layer, application layer. The physical layer address the requirements of simple but robust modulation, transmission. Since the environment is noisy and sensor nodes can be movable, the link layer is responsible for ensuring reliable communication through error control techniques and manages channel access through the media access control

(MAC) which is a sub layer of the data link layer, to minimize collision with neighbors' broadcasts. [23]

Different types of application software can be built and used on the application layer depending on the sensing tasks,. The network layer takes care of routing the data supplied by the transport layer . The transport layer responsible for reliable data delivery required by the application layer and to maintain the flow of data, the power, mobility, and task management planes monitor the power, movement, and task distribution among the sensor nodes . These planes help the sensor nodes coordinate the sensing task and lower the overall power consumption. The power management plane manages how a sensor node uses its power. For example, the sensor node may turn off its receiver after receiving a message from one of its neighbors. This is to avoid getting duplicated messages. Also, when the power level of the sensor node is low, the sensor node broadcasts to its neighbors that it is low in power and cannot participate in routing messages. The remaining power is reserved for sensing. The mobility management plane detects and registers the movement of sensor nodes, so a route back to the user is always maintained, and the sensor nodes can keep track of their neighbors. By knowing these neighbor sensor nodes, the sensor nodes can balance their power and task usage. The task management plane balances and schedules the sensing tasks given to a specific region. Not all sensor nodes in that region are required to perform the sensing task at the same time[19]. As a result, some sensor nodes perform the task more than others, depending on their power level. These management planes are needed so that sensor nodes can work together in a power-efficient way, route data in a mobile sensor network, and share resources between sensor nodes[24]. Without them, each sensor node will just work individually.

From the standpoint of the whole sensor network, it is more efficient if sensor nodes can collaborate with each other, so the lifetime of the sensor networks can be prolonged [25].

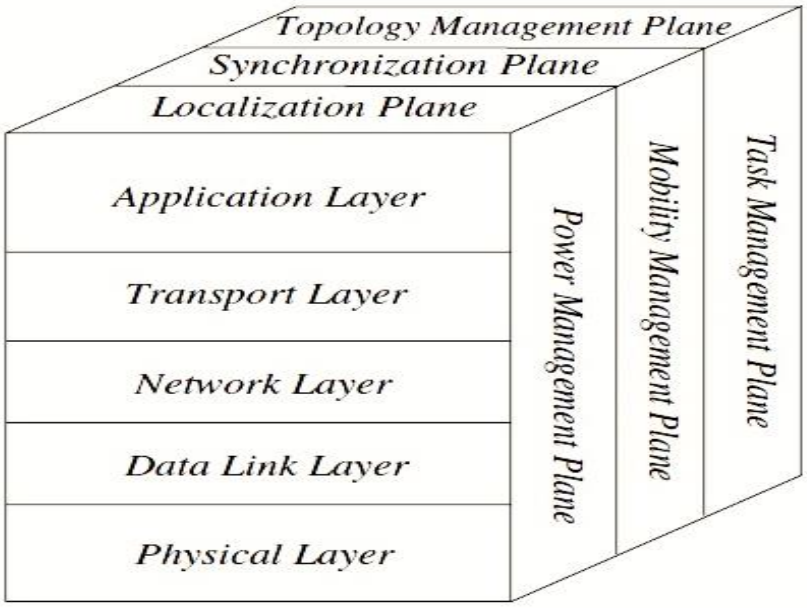

Figure 2.2: Architecture of wireless sensor networks

2.3 Applications of Wireless Sensor Networks

WSNs may consist of many different types of sensors including radar ,seismic, thermal ,magnetic, acoustic , visual, infrared, , which are able to monitor a wide variety of ambient conditions. So, there are many applications of WSNs[26]:

2.3.1 Military applications

wireless sensor networks are base on the dense operation of disposable and low-cost sensor nodes, damage of some nodes by aggressive action does not affect a services operation as much as the damage of a traditional sensor,

which makes the sensor network concept a better approach for battlefields. Some of the military applications of sensor networks are monitoring friendly forces, equipment, and ammunition; targeting , battlefield surveillance; reconnaissance of opposing forces battle damage assessment ,and terrain.

2.3.2 Environmental applications

The independent coordination capability of WSNs are utilized in the realization of a wide variety of environmental applications. environmental monitoring is a domain in which they may have a huge impact. Some of this applications include tracking of birds, and pollution studies ,monitoring environmental conditions that affect crops and livestock , forest fire detection ,the movements of small animals and environmental monitoring in marine, soil, and atmospheric contexts;; meteorological or geophysical research; flood detection; bio-complexity mapping of the environment[26].

2.3.3 Health applications

The developments in implanted biomedical devices and smart integrated sensors make the usage of sensor networks for biomedical applications possible. Some of the health applications for sensor networks are the provision of interfaces for the disabled; integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of insects or other small animals; monitoring of human physiological data; and tracking and monitoring doctors and patients inside a hospital[27].

2.3.4 Home applications

As technology advances, smart sensor nodes and actuators can be buried in appliances such as vacuum cleaners, microwave ovens, refrigerators, and DVD players as well as water monitoring systems. These sensor nodes inside domestic devices can interact with each other and with the external network via the Internet or satellite. They allow end-users to more easily manage home devices both locally and remotely[28].

2.3.5 Industrial applications

Networks of wired sensors have long been used in industrial fields such as industrial sensing and control applications, building automation, and access control. However, the cost associated with the deployment of wired sensors limits the applicability of these systems. Moreover, even if a sensor system were deployed in an industrial plant, upgrading this system would cost almost as much as a new system. In addition to sensor-based monitoring systems, manual monitoring has also been used in industrial applications for preventive maintenance. Manual monitoring is generally performed by experienced personnel using handheld analyzers that are collected from a central location for analysis. While sensor-based systems incur high deployment costs, manual systems have limited accuracy and require personnel. Instead, WSNs are a promising alternative solution for these systems due to their ease of deployment, high granularity, and high accuracy provided through battery-powered wireless communication units [29].

2.4 Architecture of the Sensor Node

Basically a sensor node is made by four components as shown in Figure 2.3: a sensing unit, a processing unit, a communication unit, a power unit. The sensing unit is made up of one or many sensors and analog to digital convertor. Where the sensor nodes sense the physical phenomenon and generate the analog signal. The ADC converts this analog signal to digital signal. After the conversion of the signal it is fed into processing unit. The processing unit has limited memory (storage) and processor (microprocessor) provides full control to sensor nodes. A communication unit use radio for data transmission. The most important component or unit of a sensor node is power unit which supplies power to the nodes. There can be more components or units can be added to the sensor node, depending on different applications [30].

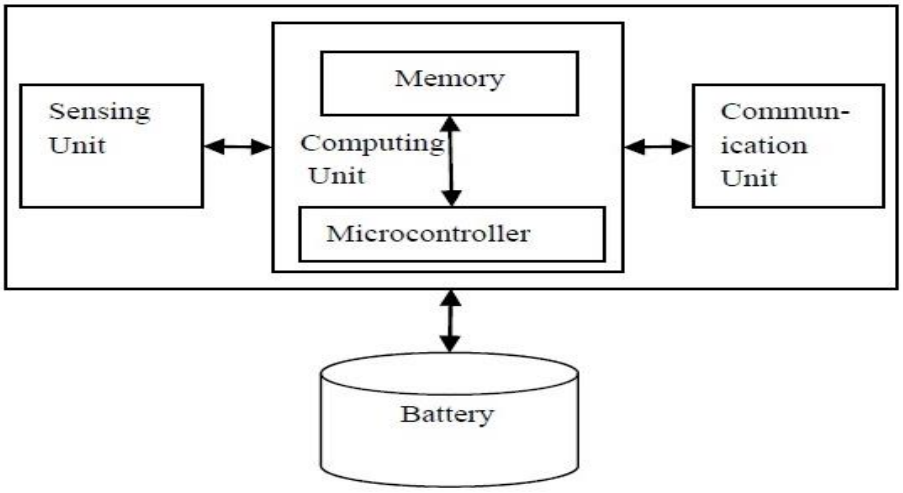

Figure 2.3: Architecture of sensor node

2.5 Standards of WSNs

Standardization in WSNs has been a major challenge. Over the years, various types of platforms have been developed, mainly for research purposes. However, due to the independent development of these platforms, interdependency has been a problem among them. Recently, the IEEE has undertaken a worldwide effort to develop a standard for low-power wireless communication. As a result, the IEEE 802.15.4 standard was developed. Since its establishment, this standard has been adopted by the majority of the platforms and industrial applications as the de facto standard for WSNs[31].

2.5.1 IEEE 802.15.4 standard

The IEEE 802.15.4 standards body was formed for the specification of low-data-rate wireless transceiver technology with long battery life and very low complexity. The IEEE 802.15.4 standard can be viewed as the low-power counterpart of the IEEE 802.11 standard developed for WLANs.

The IEEE 802.15.4 standard defines the PHY and MAC layers and provides flexibility for higher layer solutions. More specifically, the wireless spectrum to be used, wireless communication techniques, and MAC algorithms are defined. This allows for compliant transceivers to communicate with each other even though they may have been produced by different vendors. At the PHY layer, three different bands are chosen for communication: 2.4 GHz (global), 915MHz (the Americas), and 868MHz (Europe). The IEEE 802.15.4 standard also defines a MAC layer, which is based on a super frame structure and relies on carrier sense multiple accesses with collision avoidance (CSMA/CA) techniques[32].

The MAC layer provides communication for star, mesh, and cluster tree-based topologies with controllers. As part of these topologies, two types of devices are defined as a part of the standard. Full function devices (FFDs) are implemented with all the functionalities defined in the standard. They can function in any topology and can be used as a network coordinator or a router. An FFD can communicate with any other device in the network. On the other hand, reduced function devices (RFDs) are defined for very simple implementation in the network. These types are shown in Figure 2.4.

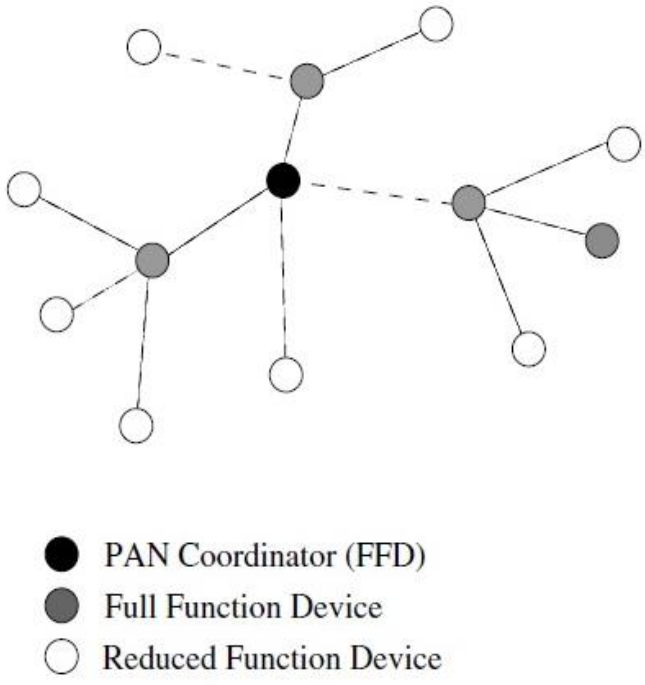

Figure 2.4: Types of IEEE 802.15.4 devices

Applications for the IEEE 802.15.4 standard include sensor networks, industrial sensing and control devices, building and home automation products, and even networked toys. Most of the recent platforms developed for WSN research comply with the IEEE 802.15.4 standard .

2.5.2 ZigBee standard

The ZigBee standard has been developed by the ZigBee Alliance, which is an international, nonprofit industrial consortium of leading semiconductor manufacturers and technology providers. The ZigBee standard was created to address the market need for cost-effective, standard-based wireless networking solutions that support low data rates[33].

Low power consumption, security, and reliability. The ZigBee standard is defined specifically in conjunction with the IEEE 802.15.4 standard. Therefore, both are usually confused. However, each standard defines specific layers of the protocol stack. As shown in Figure 2.5 below, the PHY and MAC layers are defined by the IEEE 802.15.4 standard while the ZigBee standard defines the network layer (NWK) and the application framework [34].

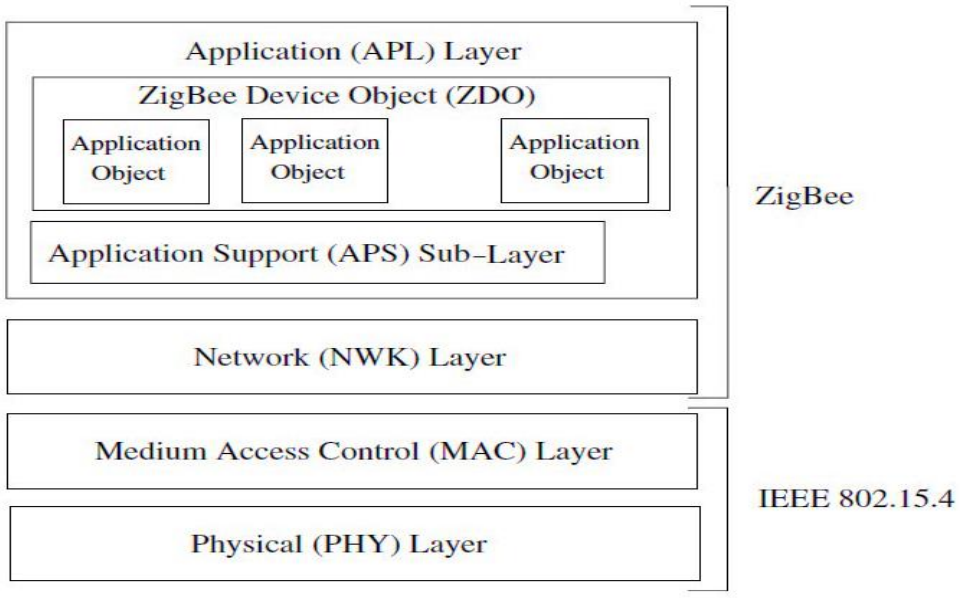

Figure 2.5: IEEE 802.15.4 and the ZigBee protocol stack

2.6 Power Consumption

A wireless sensor node can only be equipped with a limited power source due to the several hardware constraints such as the small size and mobility. Moreover, for most applications, replenishment of power resources is impossible. WSN lifetime, therefore, shows a strong dependence on battery lifetime. Thus the sources that consume energy during the operation of each node should be analyzed and maintained efficiently. The sensor node performs three functions: sensing, processing, and Communicating[35].

2.6.1 Sensing

Sensing power varies with the nature of applications and the specific sensors used. Sporadic sensing might consume less power than constant event monitoring. The complexity of event detection also plays a crucial role in determining energy expenditure. Higher ambient noise levels might cause significant corruption and increase detection complexity. While the energy consumption for sensing varies significantly with the type of sensor used, the sensor system is generally associated with an ADC subsystem. This subsystem usually consists of the sensor, a low-noise preamplifier, an anti-aliasing filter, an ADC, and a DSP. The energy consumption of an ADC depends on the performance of each of these components[36].

2.6.2 Data processing

The energy expenditure for data processing is similar to that for sensing. However, computation requires much less energy compared to data communication. As an example, assuming Rayleigh fading and fourth-power distance loss, the energy cost of transmitting a 1 kb packet over a distance of

100m is approximately equal to executing 3 million instructions by a typical microprocessor. This drastic difference between communication and computation signifies the importance of local data processing in minimizing power consumption in a multi-hop sensor network[37].

2.6.3 Communicating

Of the three domains, a sensor node expends the maximum energy in data communication. Communication is performed by the transceiver circuitry during both receiving and transmitting data[38] .

2.7 Hierarchical Routing Protocols

As mentioned before, it is proved that communication is the part that consumes the most energy, so, to reduce the power consumption of WSNs to minimum, the way of communication must be altered. In order to do that, Hierarchical routing protocols are designed. They use some routing techniques to minimize the consumption of energy. These techniques rearrange the way that sensors communicate with each other and with the base station. The major and most popular hierarchical routing protocols of WSNs are LEACH, PEGASIS, TEEN and APTEEN[39].

2.7.1 Low Energy Adaptive Clustering Hierarchy(LEACH) Protocol

Low Energy Adaptive Clustering Hierarchy is a hierarchical cluster based routing protocol. In this protocol, the network is distributed to clusters. The cluster is a group of adjacent sensors. As shown in Figure 2.6, each cluster has a node called cluster head (CH). Cluster head is responsible of

transmitting data between cluster nodes and base station. Every node in the cluster sends its data to the CH, and then the CH aggregates data and sends it to the base station (BS) or the sink[40].

As shown in Figure 2.7, the basic operations of LEACH are organized in two distinct phases. The first phase, the setup phase, consists of two steps, cluster-head selection and cluster formation. The second phase, the steady-state phase, focuses on data collection, aggregation, and delivery to the base station. The duration of the setup is assumed to be relatively shorter than the steady-state phase to minimize the protocol overhead.

At the beginning of the setup phase, a round of cluster-head selection starts. The cluster-head selection process ensures that this role rotates among sensor nodes, thereby distributing energy consumption evenly across all network nodes.

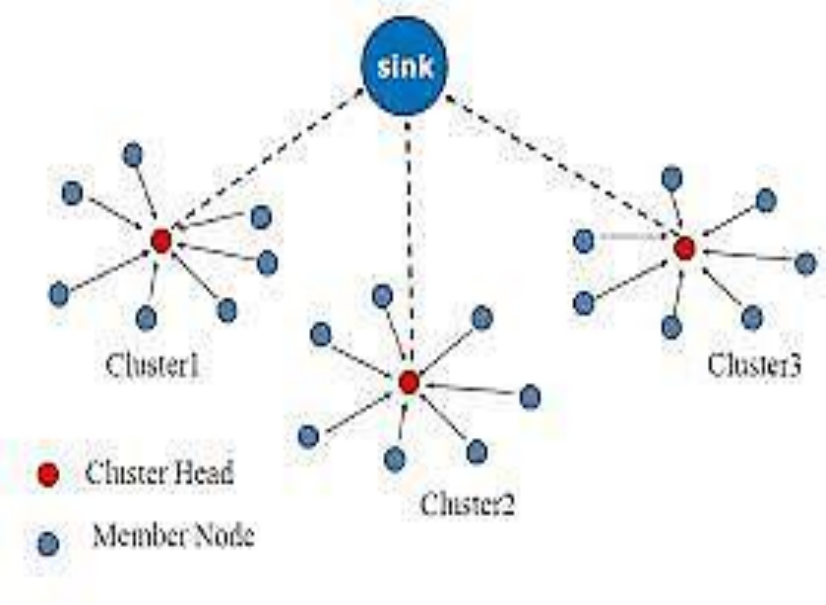

Figure 2.6: LEACH protocol

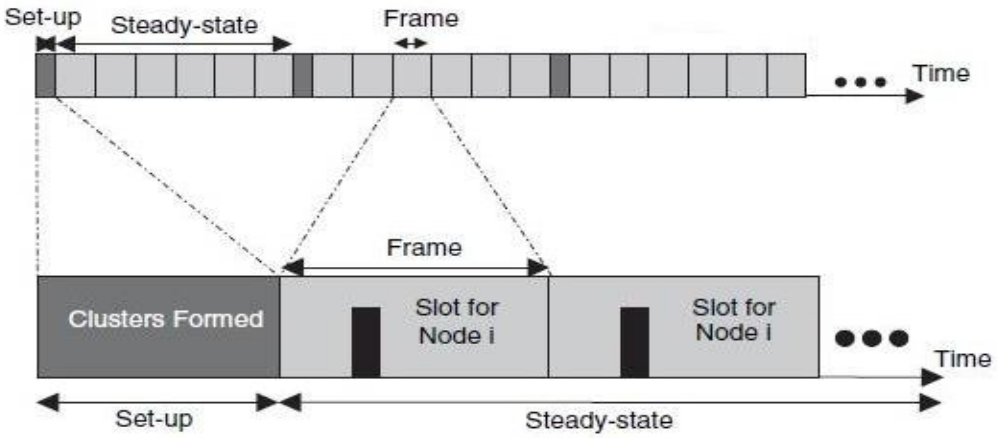

Figure 2.7: Setup phase and steady state face in LEACH

To determine if it is its turn to become a cluster head, a node, n, generates a random number, v, between 0 and 1 and compares it to the cluster-head selection threshold, T (n). The node becomes a cluster head if it's generated value, v, is less than T (n).

The cluster-head selection threshold is designed to ensure with high probability that predetermined fraction of nodes, P, is elected cluster heads at each round. Further, the threshold ensures that nodes which served in the last 1/P rounds are not selected in the current round. To meet these requirements, the threshold T (n) of a competing node n can be expressed as shown in the equation below[41].

$$T(n) = \begin{cases} 0 & if\ n \notin G \\ \frac{P}{1-P(r\ mod(1/p)} & \forall n \in G \end{cases} \quad (2.1)$$

The variable *G* represents the set of nodes that have not been selected to become cluster heads in the last (*1/P*) rounds, and *r* denotes the current round. The predefined parameter, *P*, represents the cluster-head probability. It

is clear that if a node has served as a cluster head in the last (*1/P*) rounds, it will not be elected in this round.

After the cluster heads are selected, the cluster-heads advertise to all sensor nodes in the network that they are the new cluster-heads. Once the sensor nodes receive the advertisement, they determine the cluster that they want to belong based on the signal strength of the advertisement from the cluster-heads to the sensor nodes. The sensor nodes inform the appropriate cluster-heads that they will be a member of the cluster. Afterwards, the cluster-heads assign the time on which the sensor nodes can send data to the cluster-heads.

LEACH exhibits several properties which enable the protocol to reduce energy consumption. Energy requirement in LEACH is distributed across all sensor nodes, as they assume the cluster head role in a round-robin fashion based on their residual energy.

LEACH is a completely distributed algorithm, requiring no control information from the base station. The cluster management is achieved locally, which obliterates the need for global network knowledge. Furthermore, data aggregation by the cluster also contributes greatly to energy saving, as nodes are no longer required to send their information directly to the sink[42].

This protocol is most suited for constant monitoring such as monitor machinery for fault detection and diagnosis. Many researches and studies about LEACH have been performed, so, there are several enhancements proposed on LEACH:

- **E-LEACH**

Enhanced Low-Energy Adaptive Clustering Hierarchy proposes a cluster head selection algorithm for sensor networks that have non-uniform starting energy level among the sensors. It also determines that the required number of cluster heads has to scale as the square root of the total number of sensor nodes to minimize the total energy consumption[43].

- **LEACH-C**

LEACH-Centralized uses a centralized clustering algorithm and same steady-state protocol. During the set-up phase of LEACH-C, each node sends information about current location and energy level to base station (BS). The BS will determine clusters, CH and non-CHs of each cluster. The BS utilizes its global information of the network to produce better clusters that require less energy for data transmission. The number of CHs in each round of LEACH-C equals a predetermined optimal value[44].

- **M-LEACH**

Multi-hop LEACH modifies LEACH allowing sensor nodes to use multi-hop communication within the cluster in order to increase the energy efficiency of the protocol. This work extends the existing solutions by allowing multi-hop inter-cluster communication in sparse WSNs in which the direct communication between CHs or the sink is not possible due to the distance between them [45].

2.7.2 Power Efficient Gathering in Sensor Information System (PEGASIS) Protocol

Power Efficient Gathering in Sensor Information System (PEGASIS) is a chain based protocol. The idea of cluster formation and cluster head is discarded in PEGASIS as shown in Figure 2.8. Instead of multiple nodes, a single node in the chain communicates with the base-station [6]. In this protocol, the sensor node sends its data to the closest neighbor as next hop. Then the neighbor adds this data to its own and sends it to next, until the BS is reached.

The network model considered by PEGASIS assumes a homogeneous set ofnodes deployed across a geographical area. Nodes are assumed to have global knowledge about other sensors' positions[46].

The construction of the chain starts with the farthest node from the sink. Network nodes are added to the chain progressively, starting from the closest neighbor to the end node. Nodes that are currently outside the chain are added to the chain in a greedy fashion, the closest neighbor to the top node in the current chain first, until all nodes are included. To determine the closest neighbor, the node uses the signal strength to measure the distance to all its neighboring nodes. Using this information, the node adjusts the signal strength so that only the closest node can be heard. A node within the chain is selected to be the chain leader. Its responsibility is to transmit the aggregated data to the base station.

The chain leader role shifts in positioning the chain after each round. Rounds can be managed by the data sink, and the transition from one round to

the next can be issued by the data sink. Rotation of the leadership role among nodes of the chain ensures on average a balanced consumption of energy among all the network nodes[47].

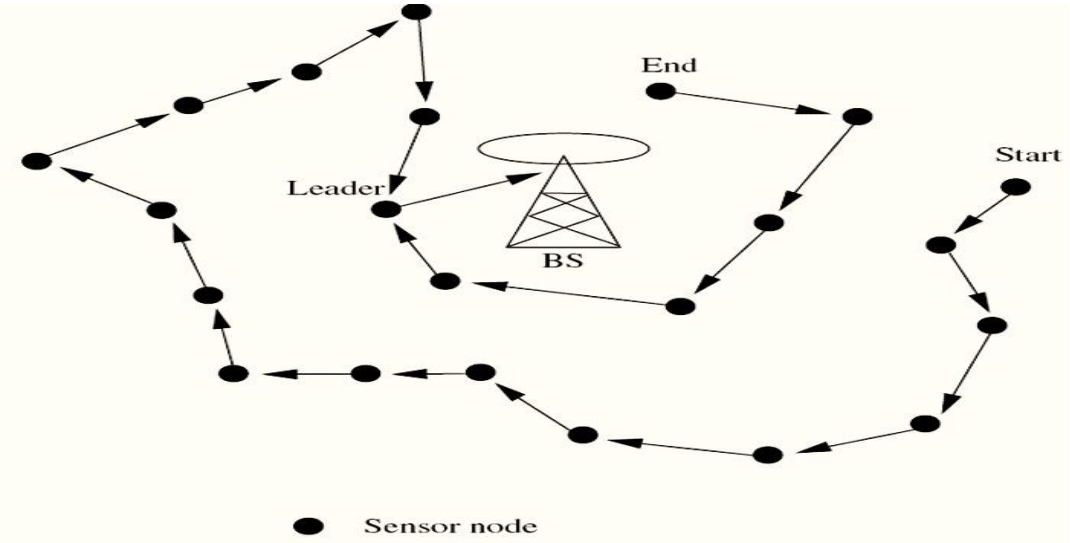

Figure 2.8: PEGASIS protocol

It is worth noting, however, that nodes assuming the role of chain leadership may be arbitrarily far away from the data sink. Such a node may be required to transmit with high power in order to reach the base station. Data aggregation in PEGASIS is achieved along the chain. In its simplest form, the aggregation process can be performed sequentially as follows: first, the chain leader issues a token to the last node in the right end of the chain. Upon receiving the token, the end node transmits its data to its downstream neighbor in the chain toward the leader. The neighboring node aggregates the data and transmits them to its downstream neighbor. This process continues until the aggregated data reach the leader. Upon receiving the data from the right side of the chain, the leader issues a token to the left end of the chain, and the same aggregation process is carried out until the data reach the leader.

Upon receiving the data from both sides of the chain, the leader aggregates the data and transmits them to the data sink [46].

2.7.3 TEEN & APTEEN

Threshold sensitive Energy Efficient sensor Network protocol (TEEN) is a hierarchical cluster based routing protocol. It organizes the sensor nodes into multiple levels of hierarchy. Here the data is transmitted from end nodes to CHs, which collect, aggregate, and transmit this data to higher level cluster heads until the BS is reached. As shown in Figure 2.9.

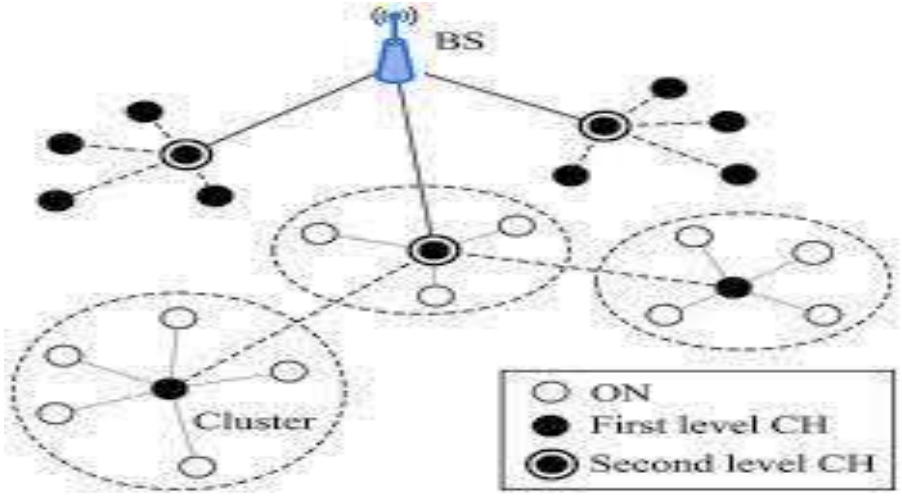

Figure 2.9: TEEN protocol

Based on this hierarchical network structure, TEEN provides event-based communication through two thresholds: hard threshold (HT) and soft threshold (ST).The sensor nodes are programmed to respond to sensed-attribute changes, such as temperature or magnetic flux, by comparing the measured value to the hard threshold. If the hard threshold HT is exceeded, the sensor node sends its observed data to the cluster head. Consequently, data are collected only if an event of interest occurs[48].

It is clear that events can last for a long time, which requires frequent data transmission. In order to reduce the redundancy in this transmission, the soft threshold ST is used. Whenever the hard threshold is exceeded, the sensor node also checks the soft threshold for consequentobservations. If the difference between consecutive observations does not exceed the soft threshold, the sensor node does not transmit this information. This informs the cluster head that similar values are observed. New observations are only transmitted if the soft threshold is exceeded. Consequently, the hard threshold limits the transmissions to those observations that match the sink's interests (exceed the hard threshold) and the soft threshold further limits the transmitted information when there is no or little change in the sensed value.

Since TEEN is based on fixed threshold limits, it is not suitable for periodic reports required by some applications. In order to provide periodic information retrieval, the adaptive threshold-sensitive energy efficient sensor network (APTEEN) protocol has been developed as an advancement of TEEN. APTEEN provides a TDMA-based structure for information transmission in each cluster. Consequently, each node transmits its information periodically to the cluster head. Moreover, the hard and soft threshold values control when and how frequently to send the data. As a result, both event-based and monitoring applications can be served [49].

CHAPTER THREE

PERFORMANCE MEASUREMENTS FOR THE WSN HIERARCHICAL ROUTING PROTOCOLS

3.1. Comparison Factors and Simulation Platform

To analyze and evaluate the performance of protocols and compare between them, the following main factors were configured in the simulation:

- End to end delay.
- Load in the network.
- Amount of traffic received by the sink (base station).
- Throughput.

To compare between the different hierarchical routing protocols of WSNs, a simulation was performed using OPNET 14.5 modeler. Because TEEN and APTEEN have the same routing technique, the simulation was achieved to LEACH, PEGASIS, and TEEN.

A tool by MIL3, Inc., OPNET (Optimized Network Engineering Tools) is an engineering system capable of simulating large networks with detailed protocol representation and performance analysis. Its features include graphical specification of models, a dynamic, event-scheduled Simulation Kernel, integrated data analysis tools and hierarchical, object based modeling. “It is a network simulation tool that permit the description of a network topology, the nodes, and the links that in the network. The processes that may happen in a node can be user defined, as can the properties of the transmission links. A simulation can then be performed, and the results analyzed for any network element in the simulated network [50].

The OPNET provides powerful tools that assist the user in the design phase of a modeling and simulation project, i.e. the execution of a simulation the building of models ,and the analysis of the output data. OPNET employs a hierarchical structure to modeling, that is, each level of the hierarchy describes different aspects of the complete model being simulated. It has a detailed library of models that provide support for existing protocols and allow researchers and developers to either modify these existing models or develop new models of their own. Furthermore, OPNET models can be compiled into executable code. An executable discrete-event simulation can be debugged or simply executed, resulting in output data. OPNET has three main types of tools - the Model Development tool, the Simulation Execution tool and the Results Analysis tool. These three types of tools are used together to model, simulate and analyze a network[51].

i. The Model Development Tool

The model tools consist of the Network Editor, the Node Editor, the Parameter Editor and the Process Editor. The Network Editor is used to propose the network models, and may consist of none or more subnets. The Node Editor is used to place the models of the nodes into the network. A node in OPNET consists of modules, such as a packet generator, connected to other modules such as processors and packet sinks, by packet streams and statistic lines. The Process Editor is used to define the processes that run inside these modules. The processes themselves are designed using State Transition Diagrams along with some textual specifications using Proto-C, an OPNET variant on the C language. The Parameter Editor allows the definition of parameters used in the input for the node modules and process models, such as the packet format, and probability density functions[52] .

ii. The Simulation Execution Tool

The *simulation execution tools* consist of the *Probe Editor* and the *Simulation Tool*. The *Probe Editor* is used to put probes at a variety of points of interest in the network model. These probes can be used to check any of the statistics computed during simulation. The *Simulation Tool* allows the user to specify a order of simulations, along with any input and output options, and many different runtime options.

iii. The Results Analysis Tool

The *results analysis* tools consist of the Analysis Tool and the Filter Editor. The *Analysis Tool* will present the results from a simulation or series of simulations as graphs. The *Filter Editor* is used to define filters to scientifically process, reduce, or combine statistical data [53].

3.2. Set the Simulation Environment

A network model was designed firstly, where the Project Editor (Figure 3.1) was used to create it and process model. After perform the set simulation, the generated statistics were collect from each network objects or from the network as whole, and interoperated.

3.2.1. The Node Editor

The Node Editor (Figure 3.2) is used to generate models of nodes. The node models are then used to produce node instances within networks in the Project Editor. Internally, OPNET node models have a modular structure. You define a node by linking various modules with packet streams and statistic wires. The relatives between modules allow packets and status information to be switch over between modules. Each module placed in a

node serves a specific purpose, such as , queuing packets ,generating packets, transmitting and receiving packets or processing packets.

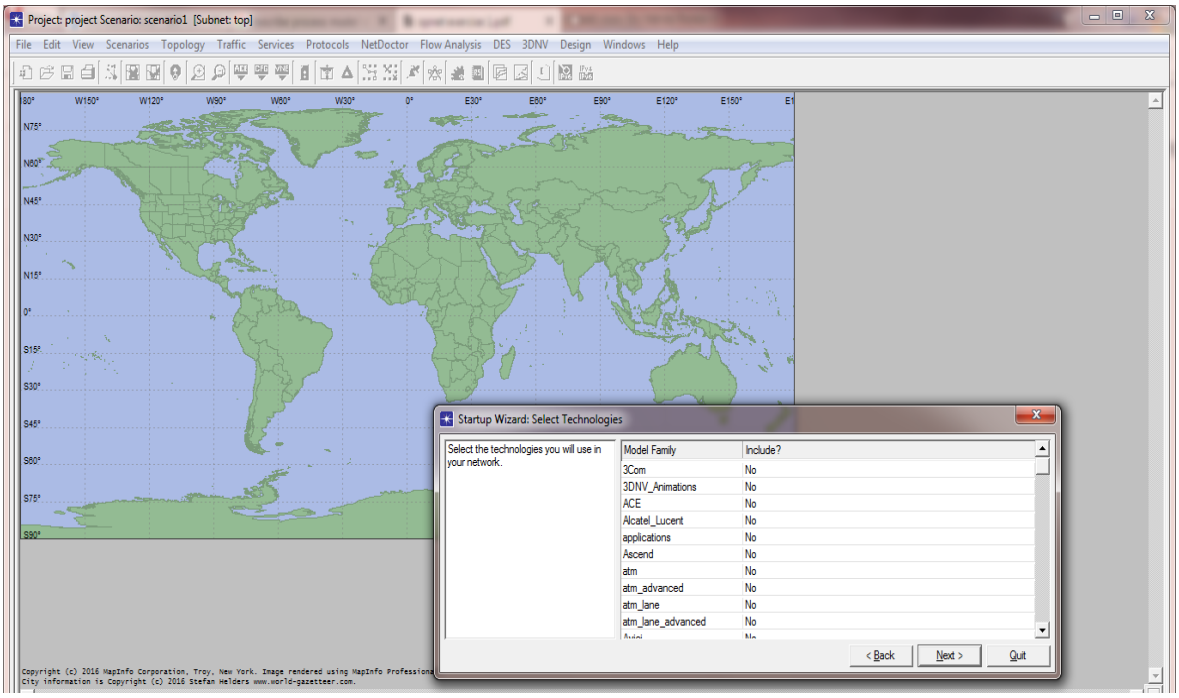

Figure 3.1: The Project Editor

3.2.2. The Process Model Editor

The process models which control the functions of the node models created in the Node Editor. Process models are represented by finite state machines (FSMs) and are created with icons that represent states and lines that represent transitions between states. procedure performed in each state or for a move are described in embedded C or C++ code blocks[54]. The process model in OPNET simulator (Figure 3.3 and Figure 3.4) have three state : **firstly** the initial state which it is the position where execution begins in a process **second** a forced state which does not permit a pause during the process **finally** an unforced state which allows a break during the process.

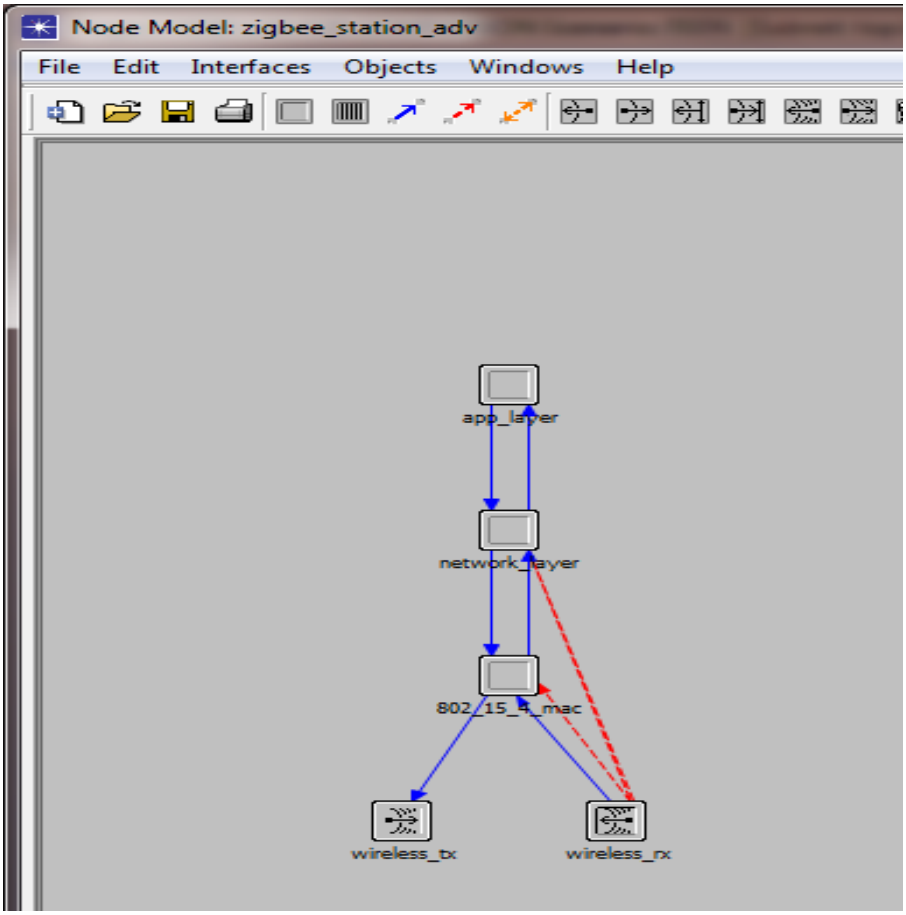

Figure 3.2: The Node Editor

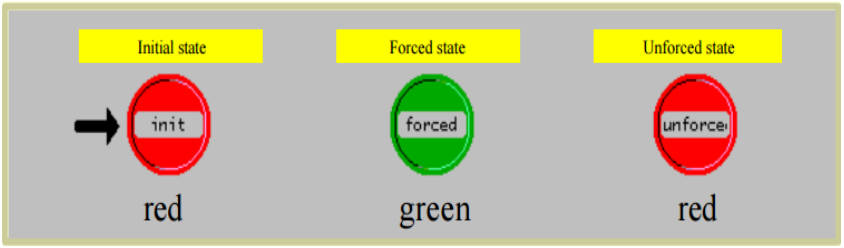

Figure 3.3: Process Model state in OPNET

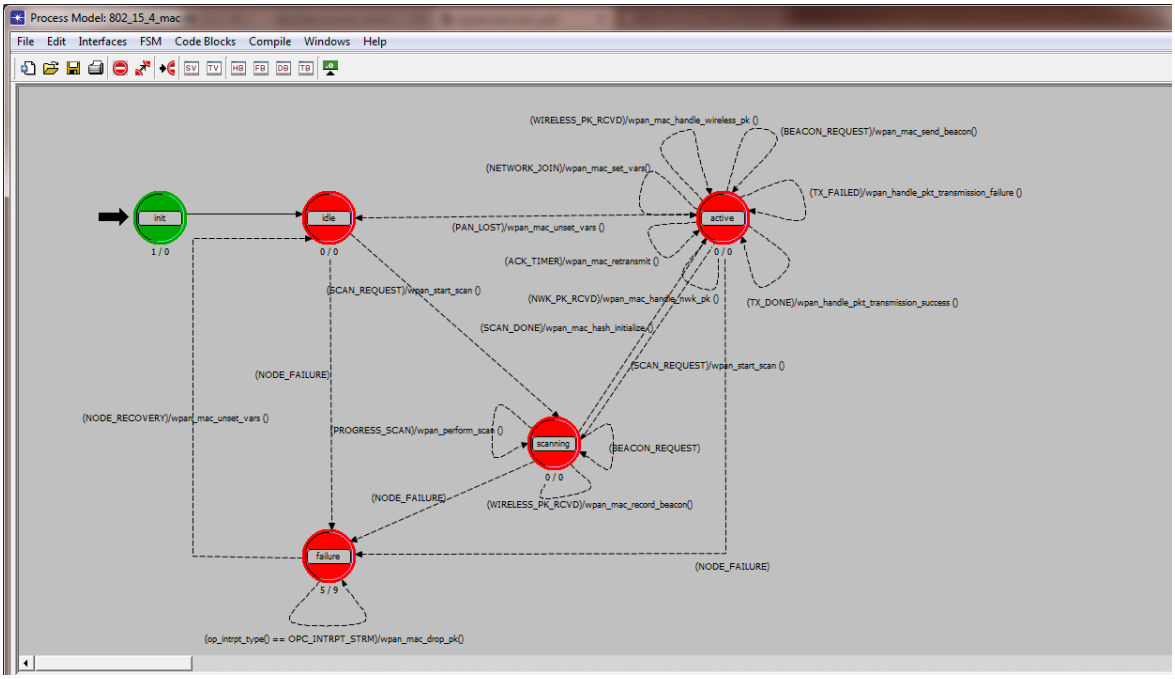

Figure 3.4: Node Process Model state in OPNET

3.2.3. Simulation Configuration

To compare between the three protocols (LEACH, PEGASIS, TEEN) according to the mentioned factors, three scenarios have been configured in one simulation project. Each scenario represents one protocol. The three scenarios have the same number of end nodes (sensors), same type of nodes, and same network topology. The only difference between them is the routing technique. In the simulation, the wireless connection protocol IEEE 802.15.4 (Zigbee) has three types of nodes:

i. Coordinator (Full Function Device):

This node coordinates other devices and provides synchronization services, it works as a base station. There is only one coordinator in the entire network. Figure 3.5 shows the shape of the coordinator in OPNET modeler.

Figure 3.5: ZigBee coordinator

ii. Router (Full Function Device)

Figure 3.6 shows the shape of router node. This node routes the data between nodes. In this simulation, it sometimes acts as a cluster head.

Figure 3.6: ZigBee router

iii. End device (Reduced Function Device)

It is the sensor node which observes the environment and sends the data to the router or the coordinator (base station) directly. This node cannot transfer data between nodes, it is shown in Figure 3.7.

Figure 3.7: ZigBee end device

All three scenarios have the same topology; it consists of 29 end devices, 4 routers, and 1 coordinator, table 3.1 below shows the simulation setting deployed in all scenarios.

Table 3.1: Simulation setting

Area	100x100 m
Wireless connection protocol	IEEE 802.15.4
Transmission band	2.4 GH
Acknowledgment mechanism	Yes/NO
Number of retransmissions when failing to send data	0/10
Packet distribution and size	Normal distribution and 1024 bits / uniform distribution and 1024 bits
Simulation duration	3 hours

3.3. Simulation Scenario for TEEN Protocol

Using this protocol, the network is divided to levels. The data travels from level to level until it reaches the base station. In this simulation, there are two clusters in level 1, and two routers in level 2. Here, the sensor nodes send the data to the CHs in level 1, the CHs aggregate the data and send it to the routers in level 2, and then the routers send the data to the coordinator (the base station) as shown in Figure 3.8.

3.4. Simulation Scenario for LEACH Protocol

In this scenario, there are 4 clusters. Each one consists of number nodes. Each end node observes the environment and sends data to the cluster head (the router). The router collects the data from the sensor nodes of its cluster and then sends the data directly to the coordinator (base station).

Figure 3.9 shows the WSN deployed in the simulation using LEACH. There is some line arrows added to show the paths of data.

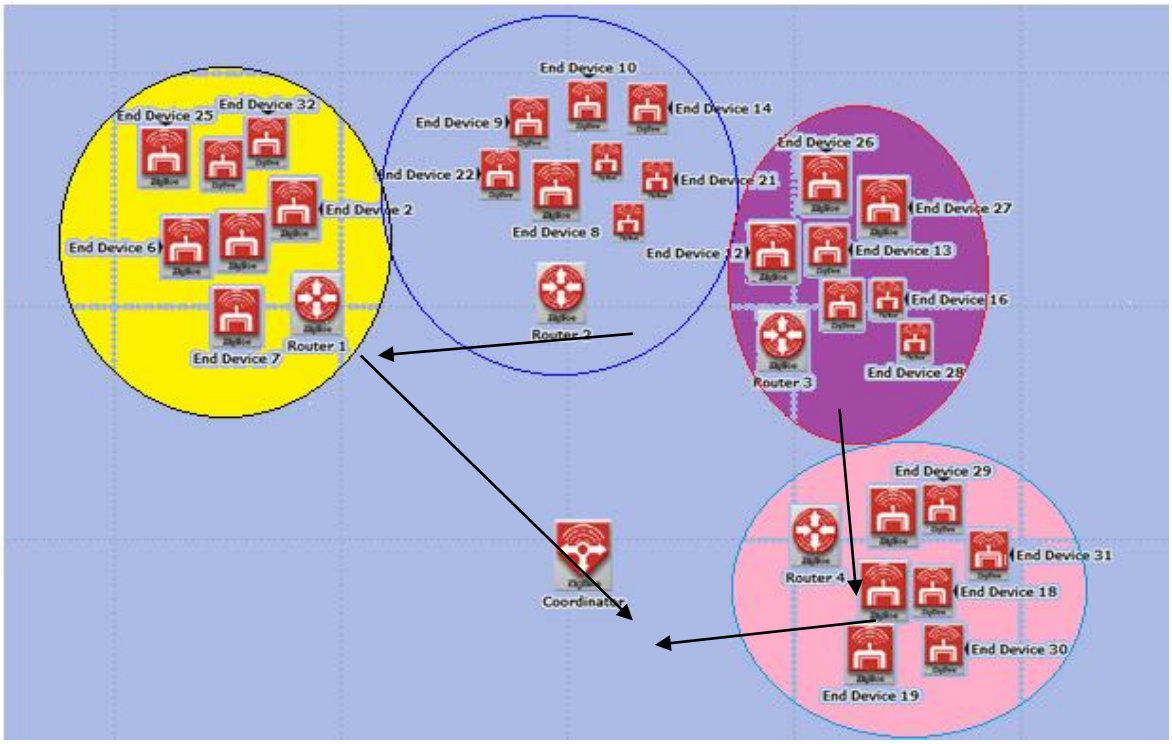

Figure 3.8 the network topology used in Teen protocol simulation

3.5. Simulation Scenario for PEGASIS Protocol

In this scenario, there are no clusters. The sensor node sends its data to its neighbor, and then the neighbor (the next hop) aggregates the data with its own and sends it to next until the aggregated data reaches the coordinator as in Figure 3.10.

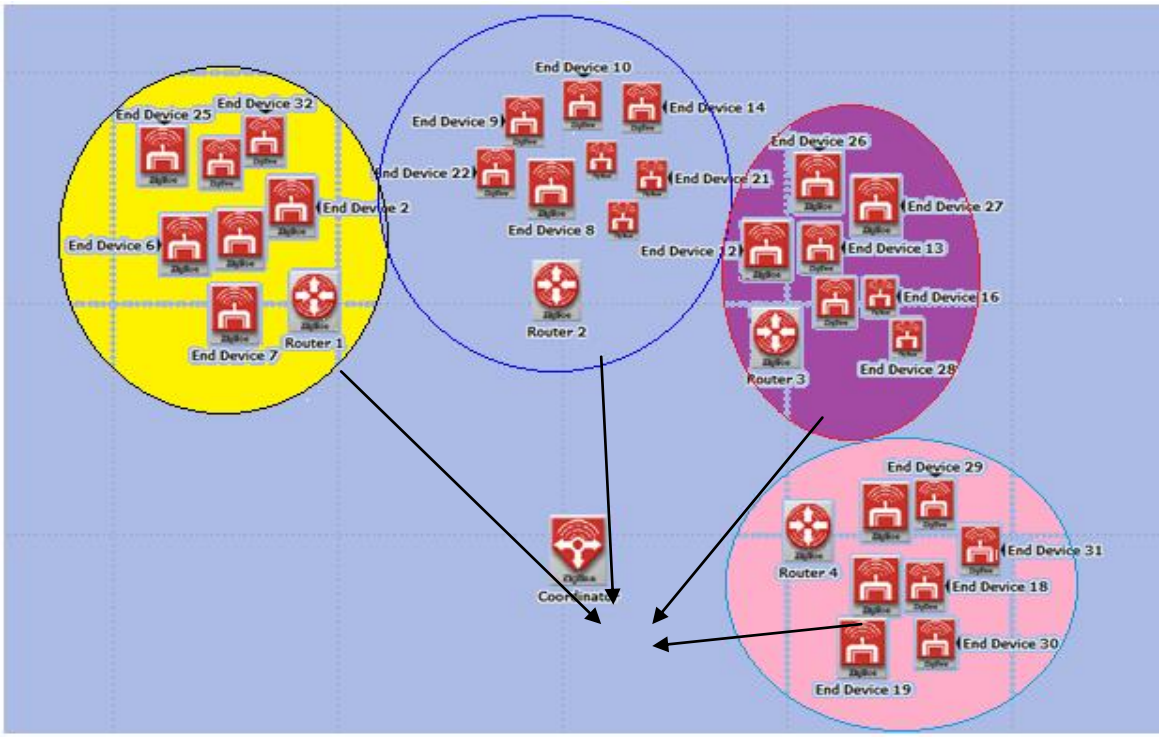

Figure 3.9: The network topology used in leach simulation

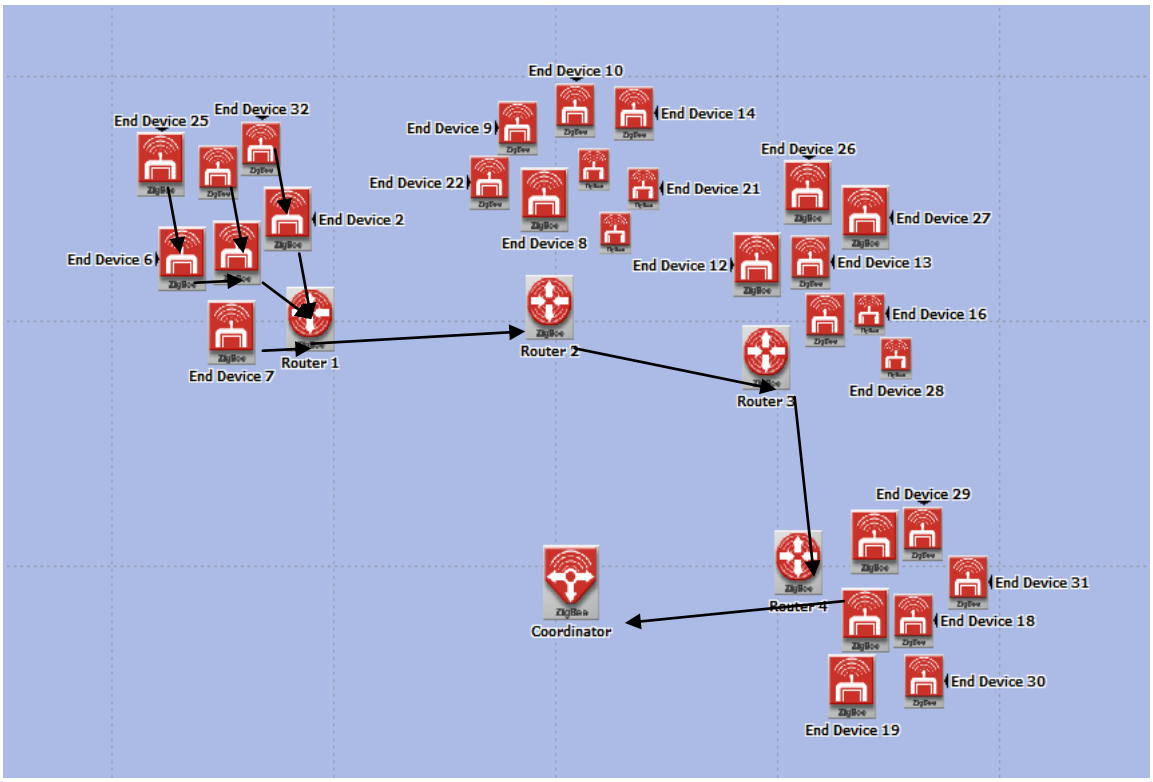

Figure 3.10: The network topology used in PEGASIS simulation