CHAPTER 1

INTRODUCTION

As the Internet of Things (IoT) becomes real, several problems, inherent to energy-constrained nodes and mesh networks, arise. One of these issues is related to the optimal transmission power for each node of the network, for which the whole network is connected but no energy is wasted. New challenges appear with the ever increasing heterogeneity of such networks to form the IoT, requiring new analysis of previous solutions and also whole new solutions. Although a centralized algorithm may present an efficiency handicap in distributed scenarios like IoT due to increased communication overhead with coordinator nodes, the centralized optimal solution still needs to be known, even if only theoretically, in order to measure how close to the optimal solution a proposal result can be. Therefore, several studies in the literature apply centralized solutions to wireless sensor networks (WSNs). According to the previously mentioned work, PSO is easy to implement, computationally efficient, and fast to converge. Thus, it will be considered in this work.

The set of issues addressed by centralized solutions for WSNs include optimal nodes deployment, node localization, energy-aware clustering and data aggregation. In this work, communications inside a single cluster will be studied. Considering nodes connectivity and deployment, a PSO-based algorithm is proposed to find a quasi-optimal transmission power for each node in order to minimize energy consumption while keeping the cluster fully-connected. Then, it will be compared to a simplistic but commonly used method in order to assess the energy-saving gain. At the present research phase, real technologies have different bandwidth, power consumption and range, e.g., WiFi, ZigBee, Bluetooth , have not been considered, although the frequency parameter can be modified to match each technology.

Clustering in WSNs is an effective technique for extending the network lifetime. Most of the traditional routing methods in clustered WSNs assume that there is no obstacle in a field of interest. Recently, there is growing interest in using wireless sensor networks (WSNs) in many applications, in order to generate report parameters such as temperature, pressure, humidity, light and chemical activity. Transmitted reports from these sensors are collected by observers (*e.g.*, basestations). The dense deployment and autonomous nature of WSNs make it quite difficult to recharge them. Therefore, energy efficiency is an important goal of the project. In these networks, in order to improve the reliability of the reported measurements, there is data aggregation reducing the overhead on the communication network, thus leading to significant energy savings. In order to support data aggregation through efficient network organization, the sensors can be divided in a number of small groups called clusters. Each cluster has a coordinator, referred to as a head set, and a number of member nodes. Some of the difficulties to design WSNs include memory and energy constraints, limited capabilities and bandwidth unavailability. Most of these issues can be modeled as optimization problems, allowing meta heuristic approaches to be used on their solution. PSO is one of the options for solving these problems in WSNs from a centralized point of view , and some of its applications are summarized next. The optimal solution to find the minimal necessary power to connect every node of a wireless sensor network would be to calculate the minimal power for each sensor node to reach only its closest neighbor, if it is placed on the edge of the network, or a set of nodes to connect the edges, if it is an intermediate node. However, as the number of nodes in the cluster increase, checking for the closest nodes and keeping the whole network connected would lead to high computational overhead. Thus, a suboptimal, but less computing-intensive solution could be applied. PSO is one possible solution, which will be considered in this proposal. Its results will be compared to a simplistic solution, where all nodes adopt the highest needed power to connect a node. Hence, it will be possible to assess the distance from the PSO solution to the simplistic one and, also, the energy saving of the former over the latter. the test procedures applied to the simplistic method and the PSO proposal are presented. Algorithms presented in the next section, have been implemented using MATLAB scripts. The PSO proposal achieves different, but similar, results for each run. Hence, it has been run five times for each scenario, and the best result among these five runs has been considered. In this work, the use of particle swarm optimization (PSO) is proposed to calculate different transmission powers for each node, without creating disconnected areas in a sensors cluster. The achieved results show that the proposed PSO algorithm is able to save sensors energy when compared to the common deployment of nodes with a single transmission power.

CHAPTER 2

WIRELESS SENSOR NETWORKS

2.1. INTRODUCTION:

Wireless Sensor Networks (WSNs) can be defined as a self-configured and infrastructureless wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location or sink where the data can be observed and analysed. A sink or base station acts like an interface between users and the network. One can retrieve required information from the network by injecting queries and gathering results from the sink. Typically a wireless sensor network contains hundreds of thousands of sensor nodes. The sensor nodes can communicate among themselves using radio signals. A wireless sensor node is equipped with sensing and computing devices, radio transceivers and power components. The individual nodes in a wireless sensor network (WSN) are inherently resource constrained: they have limited processing speed, storage capacity, and communication bandwidth. After the sensor nodes are deployed, they are responsible for self-organizing an appropriate network infrastructure often with multi-hop communication with them. Then the onboard sensors start collecting information of interest. Wireless sensor devices also respond to queries sent from a "control site" to perform specific instructions or provide sensing samples. The working mode of the sensor nodes may be either continuous or event driven. Global Positioning System (GPS) and local positioning algorithms can be used to obtain location and positioning information. Wireless sensor devices can be equipped with actuators to "act" upon certain conditions. These networks are sometimes more specifically referred as Wireless Sensor and Actuator Networks as described in (Akkaya et al., 2005).

Wireless sensor networks (WSNs) enable new applications and require non-conventional paradigms for protocol design due to several constraints. Owing to the requirement for low device complexity together with low energy consumption (i.e. long network lifetime), a proper balance between communication and signal/data processing capabilities must be found. This motivates a huge effort in research activities, standardization process, and industrial investments on this field since the last decade (Chiara et. al. 2009). At present time, most of the research on WSNs has concentrated on the design of energy- and computationally efficient algorithms and protocols, and the application domain has been restricted to simple data-oriented monitoring and reporting applications (Labrador et. al. 2009). The authors in

(Chen et al., 2011) propose a Cable Mode Transition (CMT) algorithm, which determines the minimal number of active sensors to maintain K-coverage of a terrain as well as K-connectivity of the network. Specifically, it allocates periods of inactivity for cable sensors without affecting the coverage and connectivity requirements of the network based only on local information. In (Cheng et al., 2011), a delay-aware data collection network structure for wireless sensor networks is proposed. The objective of the proposed network structure is to minimize delays in the data collection processes of wireless sensor networks which extends the lifetime of the network. In (Matin et al., 2011), the authors have considered relay nodes to mitigate the network geometric deficiencies and used Particle Swarm Optimization (PSO) based algorithms to locate the optimal sink location with respect to those relay nodes to overcome the lifetime challenge. Energy efficient communication has also been addressed in (Paul et al., 2011; Fabbri et al. 2009). In (Paul et al., 2011), the authors proposed a geometrical solution for locating the optimum sink placement for maximizing the network lifetime. Most of the time, the research on wireless sensor networks have considered homogeneous sensor nodes. But nowadays researchers have focused on heterogeneous sensor networks where the sensor nodes are unlike to each other in terms of their energy. In (Han et al., 2010), the authors addresses the problem of deploying relay nodes to provide fault tolerance with higher network connectivity in heterogeneous wireless sensor networks, where sensor nodes possess different transmission radii. New network architectures with heterogeneous devices and the recent advancement in this technology eliminate the current limitations and expand the spectrum of possible applications for WSNs considerably and all these are changing very rapidly.

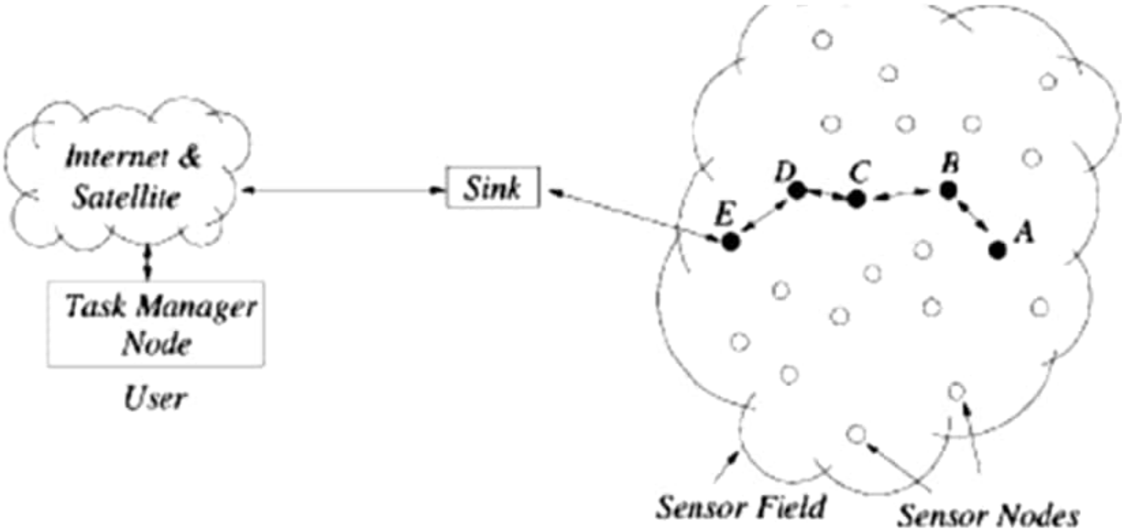

Figure 2.1. A typical Wireless Sensor Network

2.2. APPLICATIONS OF WIRELESS SENSOR NETWORK:

Wireless sensor networks have gained considerable popularity due to their flexibility in solving problems in different application domains and have the potential to change our lives in many different ways. WSNs have been successfully applied in various application domains (Akyildiz et al. 2002; Bharathidasan et al., 2001), (Yick et al., 2008; Boukerche, 2009), (Sohraby et al., 2007), and (Chiara et al., 2009;Verdone et al., 2008), such as:

Military applications: Wireless sensor networks be likely an integral part of military command, control, communications, computing, intelligence, battlefield surveillance, reconnaissance and targeting systems. Area monitoring: In area monitoring, the sensor nodes are deployed over a region where some phenomenon is to be monitored. When the sensors detect the event being monitored (heat, pressure etc), the event is reported to one of the base stations, which then takes appropriate action.

Transportation: Real-time traffic information is being collected by WSNs to later feed transportation models and alert drivers of congestion and traffic problems.

Health applications: Some of the health applications for sensor networks are supporting interfaces for the disabled, integrated patient monitoring, diagnostics, and drug administration in hospitals, tele-monitoring of human physiological data, and tracking & monitoring doctors or patients inside a hospital.

Environmental sensing: The term Environmental Sensor Networks has developed to cover many applications of WSNs to earth science research. This includes sensing volcanoes, oceans, glaciers, forests etc. Some other major areas are listed below:

1. Air pollution monitoring
2. Forest fires detection
3. Greenhouse monitoring
4. Landslide detection

Structural monitoring: Wireless sensors can be utilized to monitor the movement within buildings and infrastructure such as bridges, flyovers, embankments, tunnels etc enabling Engineering practices to monitor assets remotely with out the need for costly site visits.

Industrial monitoring: Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionalities. In wired systems, the installation of enough sensors is often limited by the cost of wiring.

Agricultural sector: using a wireless network frees the farmer from the maintenance of wiring in a difficult environment. Irrigation automation enables more efficient water use and reduces waste.

2.3. DESIGN ISSUES OF A WIRELESS SENSOR NETWORK:

There are a lot of challenges placed by the deployment of sensor networks which are a superset of those found in wireless ad hoc networks. Sensor nodes communicate over wireless, lossy lines with no infrastructure. An additional challenge is related to the limited, usually non-renewable energy supply of the sensor nodes. In order to maximize the lifetime of the network, the protocols need to be designed from the beginning with the objective of efficient management of the energy resources (Akyildiz et al., 2002). Wireless Sensor Network Design issues are mentioned in (Akkaya et al., 2005), (Akyildizet al., 2002), for simulation and testing of routing protocols for WSNs are discussed in (NS-2, Zeng et al., 1998, SensorSim, Tossiim). Let us now discuss the individual design issues in greater detail.

Fault Tolerance: Sensor nodes are vulnerable and frequently deployed in dangerous environment. Nodes can fail due to hardware problems or physical damage or by exhausting their energy supply. We expect the node failures to be much higher than the one normally considered in wired or infrastructure-based wireless networks. The protocols deployed in a sensor network should be able to detect these failures as soon as possible and be robust enough to handle a relatively large number of failures while maintaining the overall unctionality of the network. This is especially relevant to the routing protocol design, which has to ensure that alternate paths are available for rerouting of the packets. Different deployment environments pose different fault tolerance requirements.

Scalability: Sensor networks vary in scale from several nodes to potentially several hundred thousand. In addition, the deployment density is also variable. For collecting high-resolution data, the node density might reach the level where a node has several thousand neighbors in their transmission range. The protocols deployed in sensor networks need to be scalable to these levels and be able to maintain adequate performance.

Production Costs: Because many deployment models consider the sensor nodes to be disposable devices, sensor networks can compete with traditional information gathering approaches only if the individual sensor nodes can be produced very cheaply. The target price envisioned for a sensor node should ideally be less than $1.

Hardware Constraints: At minimum, every sensor node needs to have a sensing unit, a processing unit, a transmission unit, and a power supply. Optionally, the nodes may have several built-in sensors or additional devices such as a localization system to enable location-aware routing. However, every additional functionality comes with additional cost and increases the power consumption and physical size of the node. Thus, additional functionality needs to be always balanced against cost and low-power requirements.

Sensor Network Topology: Although WSNs have evolved in many aspects, they continue to be networks with constrained resources in terms of energy, computing power, memory, and communications capabilities. Of these constraints, energy consumption is of paramount importance, which is demonstrated by the large number of algorithms, techniques, and protocols that have been developed to save energy, and thereby extend the lifetime of the network. Topology Maintenance is one of the most important issues researched to reduce energy consumption in wireless sensor networks.

Transmission Media: The communication between the nodes is normally implemented using radio communication over the popular ISM bands. However, some sensor networks use optical or infrared communication, with the latter having the advantage of being robust and virtually interference free.

Power Consumption: As we have already seen, many of the challenges of sensor networks revolve around the limited power resources. The size of the nodes limits the size of the battery. The software and hardware design needs to carefully consider the issues of efficient energy use. For instance, data compression might reduce the amount of energy used for radio transmission, but uses additional energy for computation and/or filtering. The energy policy also depends on the application; in some applications, it might be acceptable to turn off a subset of nodes in order to conserve energy while other applications require all nodes operating simultaneously.

2.4. STRUCTURE OF A WIRELESS SENSOR NETWORK

Structure of a Wireless Sensor Network includes different topologies for radio communications networks. A short discussion of the network topologies that apply to wireless sensor networks are outlined below:

2.4.1. STAR NETWORK (SINGLE POINT-TO-MULTIPOINT)

A star network is a communications topology where a single base station can send and/or receive a message to a number of remote nodes. The remote nodes are not permitted to send messages to each other. The advantage of this type of network for wireless sensor networks includes simplicity, ability to keep the remote node's power consumption to a minimum. It also allows low latency communications between the remote node and the base station. The disadvantage of such a network is that the base station must be within radio transmission range of all the individual nodes and is not as robust as other networks due to its dependency on a single node to manage the network.

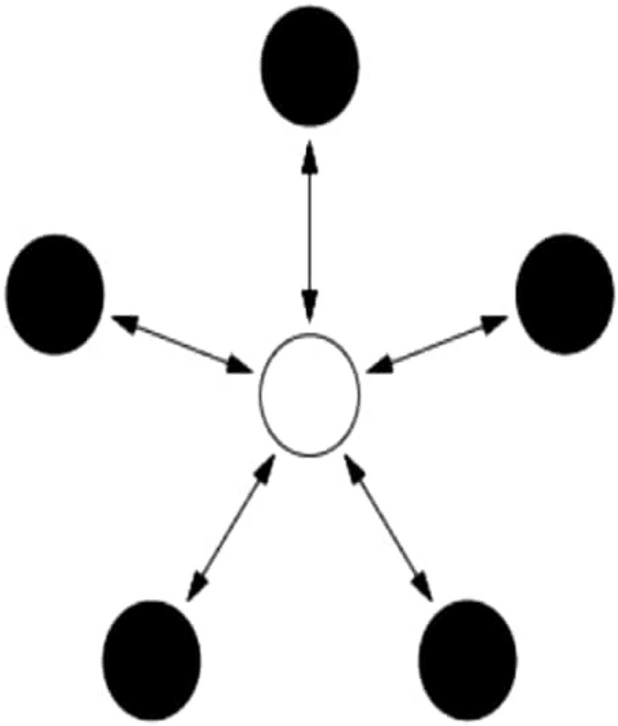

Figure 2.2. A Star network topology

2.4.2. MESH NETWORK

A mesh network allows transmitting data to one node to other node in the network that is within its radio transmission range. This allows for what is known as multi-hop communications, that is, if a node wants to send a message to another node that is out of radio communications range, it can use an intermediate node to forward the message to the desired node. This network topology has the advantage of redundancy and scalability. If an individual node fails, a remote node still can communicate to any other node in its range, which in turn, can forward the message to the desired location. In addition, the range of the network is not necessarily limited by the range in between single nodes; it can simply be extended by adding more nodes to the system. The disadvantage of this type of network is in power consumption for the nodes that implement the multi-hop communications are generally higher than for the nodes that don't have this capability, often limiting the battery life. Additionally, as the number of communication hops to a destination increases, the time to deliver the message also increases, especially if low power operation of the nodes is a requirement.

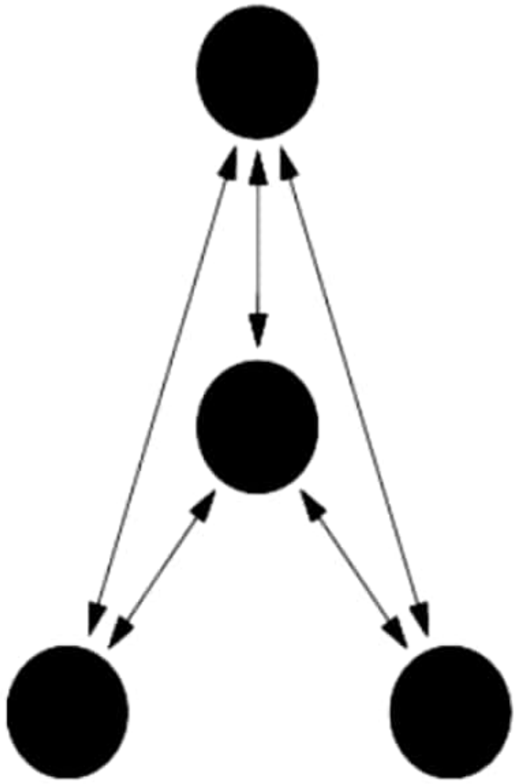

Figure 2.3. A Mesh network topology

2.4.3. HYBRID STAR – MESH NETWORK

A hybrid between the star and mesh network provides a robust and versatile communications network, while maintaining the ability to keep the wireless sensor nodes power consumption to a minimum. In this network topology, the sensor nodes with lowest power are not enabled with the ability to forward messages. This allows for minimal power consumption to be maintained. However, other nodes on the network are enabled with multi-hop capability, allowing them to forward messages from the low power nodes to other nodes on the network. Generally, the nodes with the multi-hop capability are higher power, and if possible, are often plugged into the electrical mains line. This is the topology implemented by the up and coming mesh networking standard known as ZigBee.

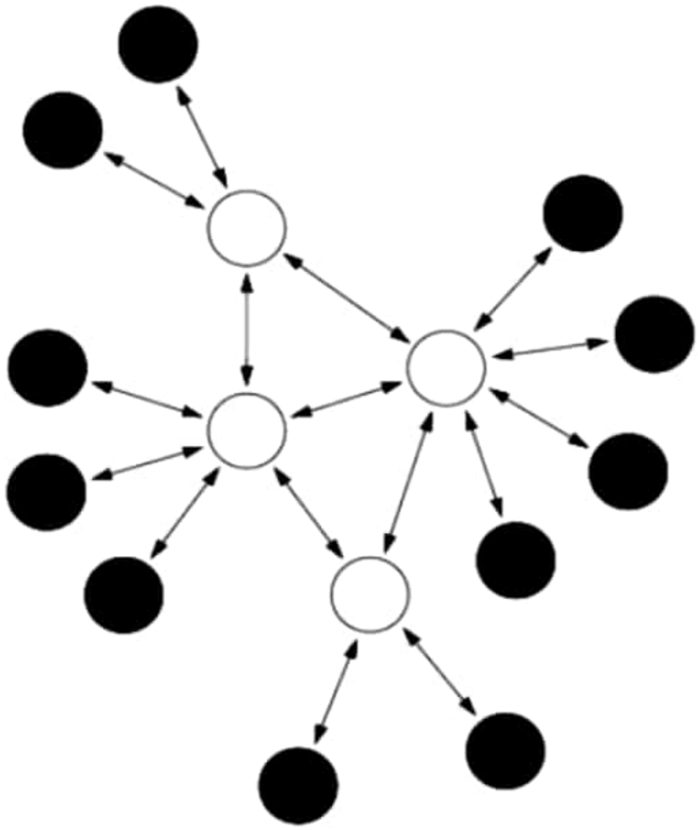

Figure 2.4. A Hybrid Star – Mesh network topology

2.5. STRUCTURE OF A WIRELESS SENSOR NODE

A sensor node is made up of four basic components such as sensing unit, processing unit, transceiver unit and a power unit which is shown in Fig. 5. It also has application dependent additional components such as a location finding system, a power generator and a mobilizer. Sensing units are usually composed of two subunits: sensors and analogue to digital converters (ADCs) (Akyildiz et al., 2002). The analogue signals produced by the sensors are converted to digital signals by the ADC, and then fed into the processing unit.

The processing unit is generally associated with a small storage unit and it can manage the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks. A transceiver unit connects the node to the network. One of the most important components of a sensor node is the power unit. Power units can be supported by a power scavenging unit such as solar cells. The other subunits, of the node are application dependent.

A functional block diagram of a versatile wireless sensing node is provided in Fig. 6. Modular design approach provides a flexible and versatile platform to address the needs of a wide variety of applications. For example, depending on the sensors to be deployed, the signal conditioning block can be re-programmed or replaced. This allows for a wide varietyof different sensors to be used with the wireless sensing node. Similarly, the radio link may be swapped out as required for a given applications' wireless range requirement and the need for bidirectional communications.

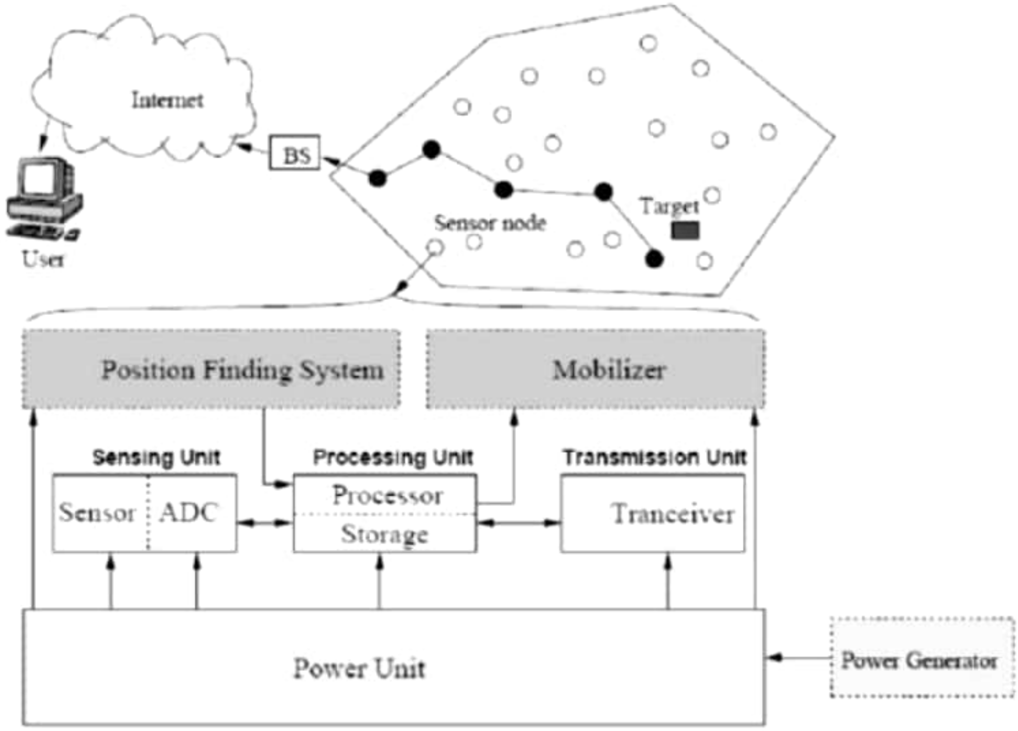

Figure 2.5. The components of a sensor node

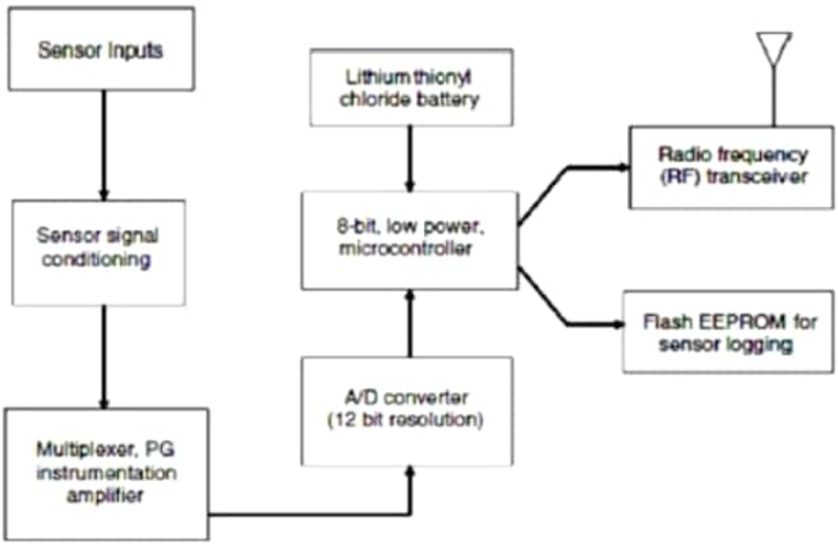

Figure 2.6. Functional block diagram of a sensor node

Using flash memory, the remote nodes acquire data on command from a base station, or by an event sensed by one or more inputs to the node. Moreover, the embedded firmware can be upgraded through the wireless network in the field.

The microprocessor has a number of functions including:

- Managing data collection from the sensors
- performing power management functions
- interfacing the sensor data to the physical radio layer
- managing the radio network protocol

A key aspect of any wireless sensing node is to minimize the power consumed by the system. Usually, the radio subsystem requires the largest amount of power. Therefore, data is sent over the radio network only when it is required. An algorithm is to be loaded into the node to determine when to send data based on the sensed event. Furthermore, it is important to minimize the power consumed by the sensor itself. Therefore, the hardware should be designed to allow the microprocessor to judiciously control power to the radio, sensor, and sensor signal conditioner .

2.6. COMMUNICATION STRUCTURE OF A WIRELESS SENSOR NETWORK

The sensor nodes are usually scattered in a sensor field as shown in Fig. 1. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink 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 Fig. 1. The sink may communicate with the task manager node via Internet or Satellite.

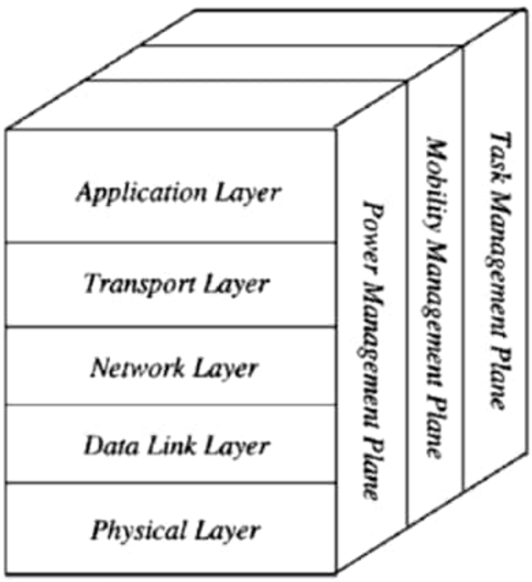

Figure 2.7. Wireless Sensor Network protocol stack

The protocol stack used by the sink and the sensor nodes is given in Fig. 7. This protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium and promotes cooperative efforts of sensor nodes. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane (Akyildiz et al., 2002). Different types of application software can be built and used on the application layer depending on the sensing tasks. This layer makes hardware and software of the lowest layer transparent to the end-user. The transport layer helps to maintain the flow of data if the sensor networks application requires it. The network layer takes care of routing the data supplied by the transport layer, specific multi-hop wireless routing protocols between sensor nodes and sink. The data link layer is responsible for multiplexing of data streams, frame detection, Media Access Control (MAC) and error control. Since the environment is noisy and sensor nodes can be mobile, the MAC protocol must be power aware and able to minimize collision with neighbours' broadcast. The physical layer addresses the needs of a simple but robust modulation, frequency selection, data encryption, transmission and receiving techniques. In addition, 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 energy consumption.

2.7. ENERGY CONSUMPTION ISSUES IN WIRELESS SENSOR NETWORK

Energy consumption is the most important factor to determine the life of a sensor network because usually sensor nodes are driven by battery. Sometimes energy optimization is more complicated in sensor networks because it involved not only reduction of energy consumption but also prolonging the life of the network as much as possible. The optimization can be done by having energy awareness in every aspect of design and operation. This ensures that energy awareness is also incorporated into groups of communicating sensor nodes and the entire network and not only in the individual nodes.

A sensor node usually consists of four sub-systems :

 a computing subsystem : It consists of a microprocessor(microcontroller unit, MCU) which is responsible for the control of the sensors and implementation of communication protocols. MCUs usually operate under various modes for power management purposes. As these operating modes involves consumption of power, the energy consumption levels of the various modes should be considered while looking at the battery lifetime of each node.

 a communication subsystem: It consists of a short range radio which communicate with neighboring nodes and the outside world. Radios can operate under the different modes. It is important to completely shut down the radio rather than putting it in the Idle mode when it is not transmitting or receiving for saving power.

 a sensing subsystem : It consists of a group of sensors and actuators and link the node to the outside world. Energy consumption can be reduced by using low power components and saving power at the cost of performance which is not required. a power supply subsystem : It consists of a battery which supplies power to the node. It should be seen that the amount of power drawn from a battery is checked because if high current is drawn from a battery for a long time, the battery will die faster even though it could have gone on for a longer time. Usually the rated current capacity of a battery being used for a sensor node is less than the minimum energy consumption. The lifetime of a battery can be increased by reducing the current drastically or even turning it off often.

To minimize the overall energy consumption of the sensor network, different types of protocols and algorithms have been studied so far all over the world. The lifetime of a sensor network can be increased significantly if the operating system, the application layer and the network protocols are designed to be energy aware. These protocols and algorithms have to be aware of the hardware and able to use special features of the micro-processors and transceivers to minimize the sensor node's energy consumption. This may push toward a custom solution for different types of sensor node design. Different types of sensor nodes deployed also lead to different types of sensor networks. This may also lead to the different types of collaborative algorithms in wireless sensor networks arena.

2.8. PROTOCOLS & ALGORITHMS OF WIRELESS SENSOR NETWORK

In WSN, the main task of a sensor node is to sense data and sends it to the base station in multi hop environment for which routing path is essential. For computing the routing path from the source node to the base station there is huge numbers of proposed routing protocols exist (Sharma et al., 2011). The design of routing protocols for WSNs must consider the power and resource limitations of the network nodes, the time-varying quality of the wireless channel, and the possibility for packet loss and delay.

The first class of routing protocols adopts a flat network architecture in which all nodes are considered peers. Flat network architecture has several advantages, including minimal overhead to maintain the infrastructure and the potential for the discovery of multiple routes between communicating nodes for fault tolerance.

A second class of routing protocols imposes a structure on the network to achieve energy efficiency, stability, and scalability. In this class of protocols, network nodes are organized in clusters in which a node with higher residual energy, for example, assumes the role of a cluster head. The cluster head is responsible for coordinating activities within the cluster and forwarding information between clusters. Clustering has potential to reduce energy consumption and extend the lifetime of the network.

A third class of routing protocols uses a data-centric approach to disseminate interest within the network. The approach uses attribute-based naming, whereby a source node queries an attribute for the phenomenon rather than an individual sensor node. The interest dissemination is achieved by assigning tasks to sensor nodes and expressing queries to relative to specific attributes. Different strategies can be used to communicate interests to the sensor nodes, including broadcasting, attribute-based multicasting, geo-casting, and any casting.

A fourth class of routing protocols uses location to address a sensor node. Location-based routing is useful in applications where the position of the node within the geographical coverage of the network is relevant to the query issued by the source node. Such a query may specify a specific area where a phenomenon of interest may occur or the vicinity to a specific point in the network environment.

In the rest of this section we discuss some of the major routing protocols and algorithms to deal with the energy conservation issue in the literatures.

1. Flooding: Flooding is a common technique frequently used for path discovery and information dissemination in wired and wireless ad hoc networks. The routing strategy of flooding is simple and does not rely on costly network topology maintenance and complex route discovery algorithms. Flooding uses a reactive approach whereby each node receiving a data or control packet sends the packet to all its neighbors. After transmission, a packet follows all possible paths. Unless the network is disconnected, the packet will eventually reach its destination. Furthermore, as the network topology changes, the packet transmitted follows the new routes. Fig. 8 illustrates the concept of flooding in data communications network. As shown in the figure, flooding in its simplest form may cause packets to be replicated indefinitely by network nodes.

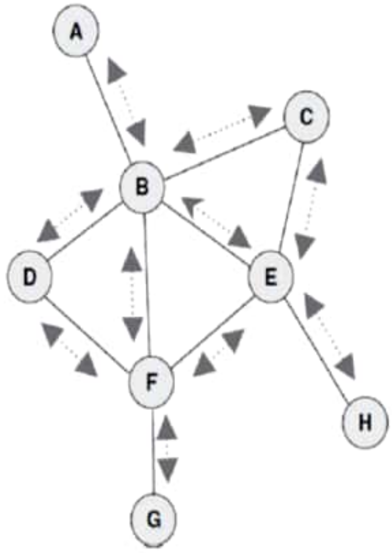

Figure 2.8. Flooding in data communication networks

1. Gossiping:

To address the shortcomings of flooding, a derivative approach, referred to as gossiping, has been proposed in (Braginsky et al., 2002). Similar to flooding, gossiping uses a simple forwarding rule and does not require costly topology maintenance or complex route discovery algorithms. Contrary to flooding, where a data packet is broadcast to all neighbors, gossiping requires that each node sends the incoming packet to a randomly selected neighbor. Upon receiving the packet, the neighbor selected randomly chooses one of its own neighbors and forwards the packet to the neighbor chosen. This process continues iteratively until the packet reaches its intended destination or the maximum hop count is exceeded.

2. Protocols for Information via Negotiation (SPIN):

Sensor protocols for information via negotiation (SPIN), is a data-centric negotiation-based family of information dissemination protocols for WSNs (Kulik et al., 2002). The main objective of these protocols is to efficiently disseminate observations gathered by individual sensor nodes to all the sensor nodes in the network. Simple protocols such as flooding and gossiping are commonly proposed to achieve information dissemination in WSNs. Flooding requires that each node sends a copy of the data packet to all its neighbors until the information reaches all nodes in the network. Gossiping, on the other hand, uses randomization to reduce the number of duplicate packets and requires only that a node receiving a data packet forward it to a randomly selected neighbor.

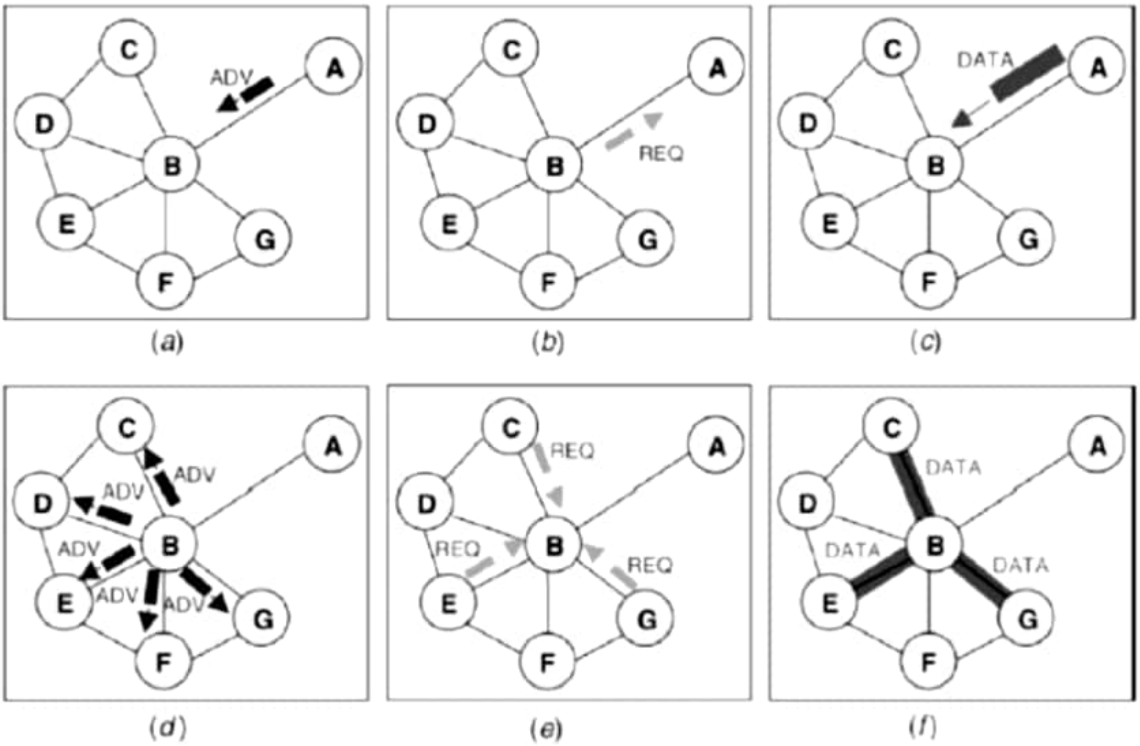

Figure 2.9. SPIN basic protocol operation

3. Low-Energy Adaptive Clustering Hierarchy (LEACH):

Low-energy adaptive clustering hierarchy (LEACH) is a routing algorithm designed to collect and deliver data to the data sink, typically a base station (Heinzelman et. al. 2000).

The main objectives of LEACH are:

- Extension of the network lifetime
- Reduced energy consumption by each network sensor node
- Use of data aggregation to reduce the number of communication messages

To achieve these objectives, LEACH adopts a hierarchical approach to organize the network into a set of clusters. Each cluster is managed by a selected cluster head. The cluster head assumes the responsibility to carry out multiple tasks. The first task consists of periodic collection of data from the members of the cluster. Upon gathering the data, the cluster head aggregates it in an effort to remove redundancy among correlated values. The second main task of a cluster head is to transmit the aggregated data directly to the base station over single hop. The third main task of the cluster head is to create a TDMA-based schedule whereby each node of the cluster is assigned a time slot that it can use for transmission. The cluster head announces the schedule to its cluster members through broadcasting. To reduce the likelihood of collisions among sensors within and outside the cluster, LEACH nodes use a code-division multiple access–based scheme for communication.

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 steadystate phase to minimize the protocol overhead.

At the beginning of the setup phase, a round of cluster-head selection starts. To decide whether a node to become cluster head or not a threshold T(s) is addressed in which is as follows:

$$T(s) = \begin{cases} \dfrac{p_{opt}}{1 - p_{opt} \cdot (r.\text{mod}.\dfrac{1}{p_{opt}})}, ifs \in G' \\ 0, otherwise \end{cases}$$

Where r is the current round number and G is the set of nodes that have not become cluster head within the last 1/popt rounds. At the beginning of each round, each node which belongs to the set G selects a random number 0 or 1. If the random number is less than the threshold T(s) then the node becomes a cluster head in the current round.

4. Threshold-sensitive Energy Efficient Protocols (TEEN and APTEEN):

Two hierarchical routing protocols called TEEN (Threshold-sensitive Energy Efficient sensor Network protocol), and APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) are proposed in (Manjeshwar et al., 2001) and (Manjeshwar et al., 2002) , respectively. These protocols were proposed for time-critical applications. In TEEN, sensor nodes sense the medium continuously, but the data transmission is done less frequently. A cluster head sensor sends its members a hard threshold, which is the threshold value of the sensed attribute and a soft threshold, which is a small change in the value of the sensed attribute that triggers the node to switch on its transmitter and transmit. Thus the hard threshold tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions that might have otherwise occurred when there is little or no change in the sensed attribute. A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption. Thus, the user can control the trade-off between energy efficiency and data accuracy. When cluster-heads are to change, new values for the above parameters are broadcast. The main drawback of this scheme is that, if the thresholds are not received, the nodes will never communicate, and the user will not get any data from the network at all.

5. Power-Efficient Gathering in Sensor Information Systems (PEGASIS):

Power-efficient gathering in sensor information systems (PEGASIS) (Lindsey et al., 2002) and its extension, hierarchical PEGASIS, are a family of routing and information-gathering protocols for WSNs. The main objectives of PEGASIS are twofold. First, the protocol aims at extending the lifetime of a network by achieving a high level of energy efficiency and uniform energy consumption across all network nodes. Second, the protocol strives to reduce the delay that data incur on their way to the sink.

The network model considered by PEGASIS assumes a homogeneous set of nodes deployed across a geographical area. Nodes are assumed to have global knowledge about other sensors' positions. Furthermore, they have the ability to control their power to cover arbitrary ranges. The nodes may also be equipped with CDMA-capable radio transceivers.

The nodes' responsibility is to gather and deliver data to a sink, typically a wireless base station. The goal is to develop a routing structure and an aggregation scheme to reduce energy consumption and deliver the aggregated data to the base station with minimal delay while balancing energy consumption among the sensor nodes. Contrary to other protocols, which rely on a tree structure or a cluster-based hierarchical organization of the network for data gathering and dissemination, PEGASIS uses a chain structure.

6. Directed Diffusion:

Directed diffusion (Intanagonwiwat et al., 2000) is a data-centric routing protocol for information gathering and dissemination in WSNs. The main objective of the protocol is to achieve substantial energy savings in order to extend the lifetime of the network. To achieve this objective, directed diffusion keeps interactions between nodes, in terms of message exchanges, localized within limited network vicinity. Using localized interaction, direct diffusion can still realize robust multi-path delivery and adapt to a minimal subset of network paths. This unique feature of the protocol, combined with the ability of the nodes to aggregate response to queries, results into significant energy savings.

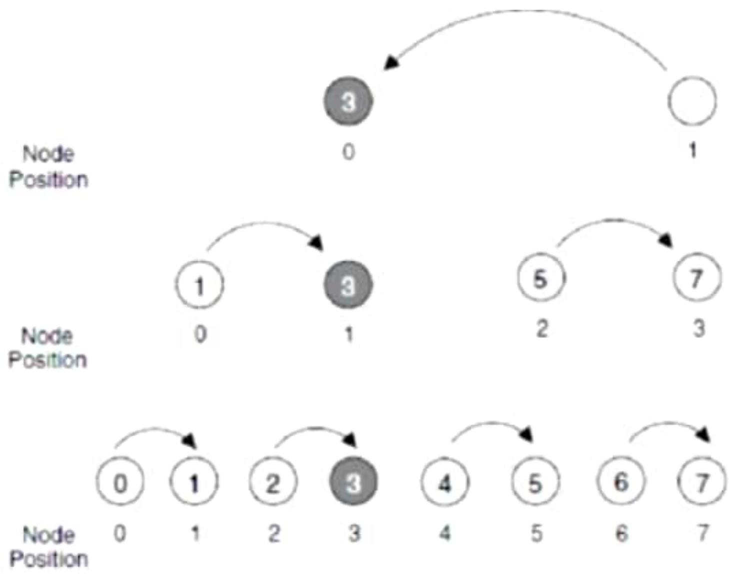

Figure 2.10. Chain-based data gathering and aggregation scheme

The main elements of direct diffusion include interests, data messages, gradients, and reinforcements. Directed diffusion uses a publish-and-subscribe information model in which an inquirer expresses an interest using attribute–value pairs. An interest can be viewed as a query or an interrogation that specifies what the inquirer wants.

7. Geographic Adaptive Fidelity (GAF):

GAF (Xu et al., 2001) is an energy-aware location-based routing algorithm designed mainly for mobile ad hoc networks, but may be applicable to sensor networks as well. The network area is first divided into fixed zones and forms a virtual grid. Inside each zone, nodes collaborate with each other to play different roles. For example, nodes will elect one sensor node to stay awake for a certain period of time and then they go to sleep. This node is responsible for monitoring and reporting data to the BS on behalf of the nodes in the zone. Hence, GAF conserves energy by turning off unnecessary nodes in the network without affecting the level of routing fidelity.

CHAPTER 3

DEPLOYMENT IN WIRELESS SENSOR NETWORKS

The performance of a wireless sensor network is greatly influenced by the process of deploying the sensor nodes. The issue of deployment and positioning of sensor nodes in a WSN is a strategy which is used in defining the topology of the network, the number and the position of the sensor nodes.

Quality monitoring, connectivity, and power consumption are also directly affected by the network topology. The problem of optimal placement of nodes is proven NP-hard for most deployment formulations [1]. The deployment activities can be grouped under three main phases. A pre-deployment and deployment phase that concerns the manual placement of the nodes by a human or a robot, or launching them from a plane (a helicopter or a drone). A post-deployment phase which is necessary if the network topology has been evolved due to a displacement of nodes, or a change of radio propagation conditions. The third phase is the redeployment which consists in adding new nodes to the network to replace some broken down or damaged nodes. Fig illustrates the different deployment phases.

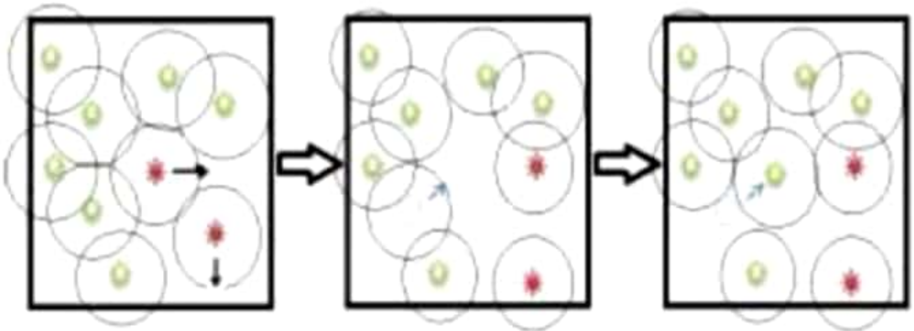

Figure 3.1. Post-Deployment phase

Different issues are discussed at the deployment of sensor nodes in a WSN. These studies concern mainly stationary and mobile case, mono and multi-objective case, deterministic and stochastic case, and finally the static and dynamic case. In the dynamic deployment context for example, authors in present and discuss different research works that aim to provide repositioning schemes nodes and some related problems. Authors in [2] propose a detailed study of the deployment in the static case. They distinguish two deployment methodologies depending on the distribution of nodes (either random or controlled). Different primary objectives are treated:

 Coverage: it is among the most predominant issues to ensure the quality of service in a WSN. Several types of coverage are presented: area coverage, barrier coverage and point (event or moving target) coverage. Fig. 4 presents the different coverage types.

 Optimization of the energy consumption by nodes andassurance of the energy efficiency,

 Network connectivity,

 Lifetime of the network,

 Network traffic,

 Reliability of data

 Cost of deployment (the number of deployed nodes)

 Fault tolerance and load balancing between nodes.

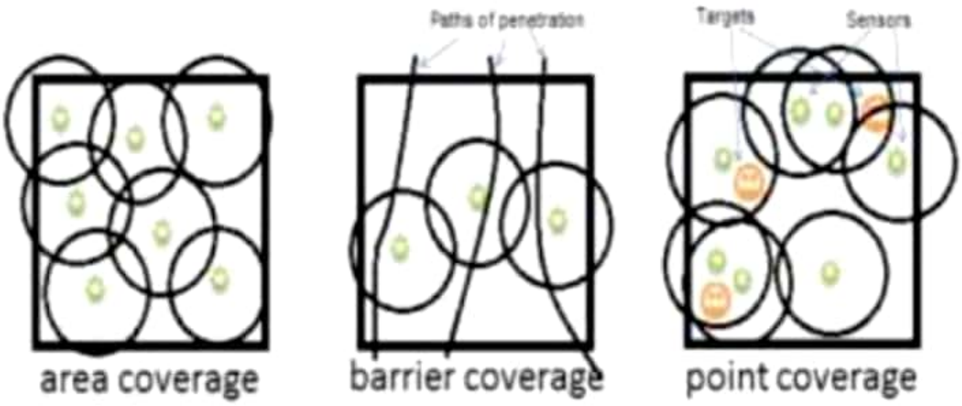

Figure. 3.2. Coverage types

In what follows, we discuss centralized, decentralized and hybrid approaches to solve the problem of deployment. Then, we present some similar problems and applications of the deployment.

CHAPTER 4

LEACH ROUTING PROTOCOL FOR WSN

4.1. INTRODUCTION

A wireless sensor networks consist of tiny sensor nodes to monitor physical or environmental conditions such as temperature, pressure, sound, humidity etc. The network must possess self configuration capabilities as the positions of the individual sensor nodes are not predetermined.

Routing strategies and security issues are a great research challenge now days in WSN but in this paper we will emphasize on the routing protocol. A number of routing protocols have been proposed for WSN but the most well known are hierarchical protocols like LEACH and PEGASIS.

Hierarchical protocols are defined to reduce energy consumption by aggregating data and to reduce the transmissions to the Base Station. LEACH is considered as the most popular routing protocol that use cluster based routing in order to minimize energy consumption. In this paper firstly we analyze LEACH protocol and then in the third section we will discuss the phases of LEACH protocol. In the fourth section we define various possible attacks on it and in the fifth section there are the advantages and disadvantages of LEACH. In the last section we compare LEACH with other protocols.

Low Energy Adaptive Clustering Hierarchy (LEACH) protocol is a TDMA based MAC protocol. The principal aim of this protocol is to improve the lifespan of wireless sensor networks by lowering the energy consumption required to create and maintain Cluster Heads. The operation of LEACH protocol consists of several rounds with two phases : Set-up Phase and Steady Phase. In the Set-up phase the main goal is to make cluster and select the cluster head for each of the cluster by choosing the sensor node with maximum energy. Steady Phase which is comparatively longer in duration than the set-up deals mainly with the aggregation of data at the cluster heads and transmission of aggregated data to the Base station.

PHASES OF LEACH

As described earlier the operation of LEACH consists of several rounds with two phases in each round. Working of LEACH starts with the formation of clusters based on the received signal strength.

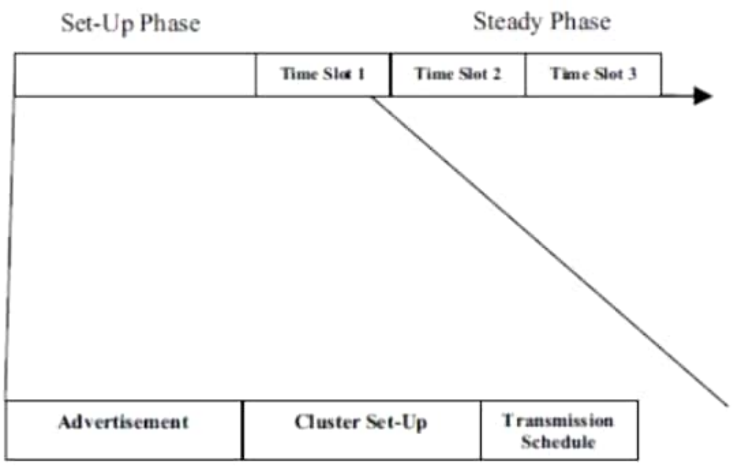

Fig. 1 Time Line Operation of LEACH

The algorithm for LEACH protocol is as follows:

The first phase of LEACH is Set-up phase and it has three fundamental steps.

1. Cluster Head advertisement
2. Cluster setup
3. Creation of Transmission Schedule

During the first step cluster head sends the advertisement packet to inform the cluster nodes that they have become a cluster head on the basis of the following formula [5]:

Let x be any random number between 0 and 1.

Where n is the given node, p is the probability, r is the current round, G is the set of nodes that were not cluster heads in the previous round, T(n) is the Threshold.

$$T(n) = \begin{cases} \dfrac{P}{1 - P[r * \mathrm{mod}(1/P)]} & if \ n \in G \\ 0 & otherwise, \end{cases}$$

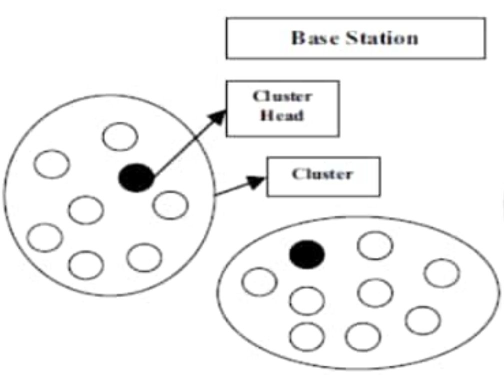

Fig. 2 Cluster Formation in LEACH.

The node becomes cluster head for the current round if the number is less than threshold T(n). Once the node is elected as a cluster head it cannot become cluster head again until all the nodes of the cluster have become cluster head once. This helps in balancing the energy consumption.

In the second step, the non cluster head nodes receive the cluster head advertisement and then send join request to the cluster head informing that they are the members of the cluster under that cluster head as shown in Fig. 2 [6].

These non cluster head nodes saves a lot of energy by turning off their transmitter all the time and turn it ON only when they have something to transmit to the cluster head.

In the third step, each of the chosen cluster head creates a transmission schedule for the member nodes of their cluster. TDMA schedule is created according to the number of nodes in the cluster. Each node then transmits its data in the allocated time schedule.

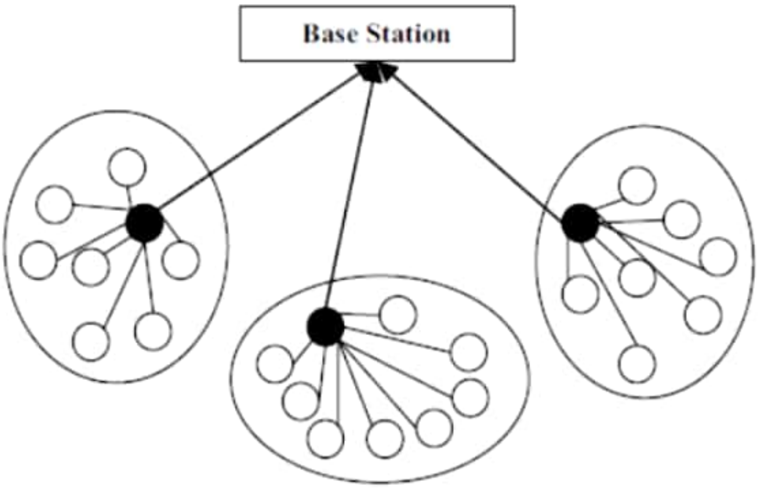

Fig. 3 Steady Phase in LEACH.

The second phase of LEACH is the Steady phase during which the cluster nodes send their data to the cluster head. The member sensors in each cluster communicate only with the cluster head via a single hop transmission. The cluster head then aggregates all the collected data and forwards this data to the base station either directly or via other cluster head along with the static route defined in the source code as shown in Fig. 3[7]. After the certain predefined time, which is decided beforehand, the network again goes back to the Set-up phase.

4.2. ANALYSIS OF LEACH ENERGY PARAMETERS

Wireless sensor networks (WSNs) are a current topic of research. Although the field has not developed as fast as some had expected [1], intense research continues. WSNs are proposed for applications as diverse as military, business, and healthcare [2]. For example, a recent study on WSNs in the oil, gas, and resource industries is presented by the authors of [3]; a healthcare monitoring application is presented in [4]. The LEACH protocol [5], remains a popular protocol for researching WSNs. It assumes that the network is subdivided into groups of nodes, called clusters. Each cluster contains a cluster head, which receives data from the other nodes, aggregates the data, and transmits to a base station. The LEACH protocol contains models for the energy used in receiving, transmitting, and aggregating data. The authors of [5] specified parameter values for transmission energy expended by sensor nodes based on assumptions about their electronics and physical properties. In this paper we analyse the effect of varying the parameter values used in the LEACH model. We select parameter values based on the varied physical properties of the networks developed and the introduction of new transceiver modules in the decade since the introduction of the LEACH protocol. Section 2 addresses related work. Section 3 summarizes the LEACH energy model. Section 4 examines the effect of different parameter values. Section 5 presents experimental results, and Section 6 contains the conclusions and discussion.

4.2.1. ENERGY MODEL

The energy model used in LEACH-like protocols assumes that transmission energy is composed of a constant amount of energy consumed by the electronics and a propagation energy proportional to the transmitter – receiver separation distance raised to a power of 2 or 4, depending on whether the distance is larger or smaller than the crossover distance [5]. The transmission energy depends on the number of bits transmitted.

The energy required to transmit an *l*-bit message is

$$E_{tx} = lE_{elect} + l\mu d^{n}$$

where

l is the number of bits; Eelect is the energy consumed per bit by the electronics;

□dn is the propagation energy per bit ;

d is the distance of transmission;

n is the propagation loss exponent.

For distances smaller than d0, the value of □ is based on the Friis free-space equation. The propagation
energy is therefore

$$E_{tx1} = lE_{elect} + l\varepsilon_{fs}d^2 \quad \text{for } d \le d_0$$

where □*fs* = □ Ñs the free space factor.

For distances smaller than *d0*, the Friis free space equation is used for the transmission power:

$$Pt(d) = \frac{\Pr(4\pi d)^2 L}{G_t G_r \lambda^2}$$

where

Pr is the minimum receive power;

Gt and *Gr* are the gains of the transmitting and receiving antennas respectively;

□ is the wavelength of the carrier;

d is the receiver-transmitter separation distance;

L is a system loss factor.

For distances greater than the crossover distance, the two-ray model, which uses a path loss exponent
of 4, is used [11]:

$$Pt(d) = \frac{P_r d^4}{G_t G_r h_t^2 h_r^2}$$

where *ht* and *hr* are the heights of the transmitting and receiving antennas above ground respectively. At the crossover distance, (3) and (4) yield the same value. The LEACH model derives the crossover distance based on this observation. The value of □*fs* = □ is determined by observing that the transmission power is equal to bit rate times the propagation energy per bit:

$$P_t(d) = R_b \varepsilon_{fs} d^2$$

where *Rb* is the bit rate.

An expression for □*fs* is found by equating (3) and (5):

$$\varepsilon_{fs} = \frac{\Pr(4\pi)^2 L}{\lambda^2 G_t G_r R_b}$$

The energy equations and the value of □ for distances greater than *d0* are found using a similar derivation.

The value of □ is called the multi-path factor, □*mp*. Therefore

$$E_{tx2} = lE_{elect} + l\varepsilon_{mp} d^4$$

$$P_t(d) = R_b \varepsilon_{mp} d^4$$

and an expression for □*mp* is found by equating (4) and (8):

$$\varepsilon_{mp} = \frac{P_r}{R_b G_t G_r h_t^2 h_r^2}$$

The crossover distance, *d0*, is found by equating (5) and (8):

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$$

CHAPTER 5

PARTICLE SWARM OPTIMIZATION

5.1. INTRODUCTION

PSO has been proposed by Eberhart and Kennedy in 1995, subsequently developed in thousands of scientific papers, and applied to many diverse problems, for instance neural networks training, data mining, signal processing, and optimal design of experiments.

PSO is a **swarm intelligence** meta-heuristic inspired by the group behavior of animals, for example bird flocks or fish schools. Similarly to genetic algorithms (GAs), it is a population-based method, that is, it represents the state of the algorithm by a population, which is iteratively modified until a termination criterion is satisfied. In PSO algorithms, the population P={p1,…,pn} of the feasible solutions is often called a **swarm**. The feasible solutions p1,…,pn are called **particles**. The PSO method views the set Rd of feasible solutions as a "space" where the particles "move". For solving practical problems, the number of particles is usually chosen between 10 and 50.

5.2. PARTICLE SWARM OPTIMIZATION THEORY

The Particle Swarm Optimization (PSO) algorithm has been introduced by Kennedy and Eberhart to deal with optimization problems of non-linear functions. Through an iterative process, the algorithm performs the search for an optimal solution of a given problem in a d-dimensional space.

In PSO, the particles have a social behavior through the interaction among themselves, and through the exchange of experiences with the swarm. In each step of the algorithm, the particles are evaluated through a fitness function and have their positions and speeds updated through the individual and collective results. The algorithm is stopped when a quasioptimal solution is found or by reaching the maximum number of execution steps.

Equations (1) and (2) define the speed and position updating of each particle in the tth step of the algorithm. For a determined particle i in the d-dimensional search space, it can be considered that: xi=[xi1, xi2, xi3, ...,xid] is the particle position vector; vi=[vi1, vi2, vi3, ..., vid] is the particle velocity vector; pBesti is the individual position of particle i that has the best fitness value; gBest is the global position that has the best swarm fitness value; c1 and c2 are the social and cognitive parameters, and; r1 and r2 are values randomly generated in the interval [0,1]. The term ω is known as the inertia factor, used to achieve a better control of the particle velocity in the search space. It will be initialized with a maximum value ωMax and decreased linearly to a minimum value ωMin, subject to the maximum number of steps of the algorithm, according to Shi and Eberhart .

$$v_i(t+1) = \omega v_i(t) + c_1 r_1[pBest_i - x_i(t)] \\ + c_2 r_2[gBest - x_i(t)]$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

The position and the speed of each particle is limited by the constraints [xMin; xMax] and [vMin; vMax], respectively, in order to prevent paths outside the search space from being explored, according to Poli et al.

5.3. SWARM TOPOLOGY

Each particle i has its **neighborhood** Ni (a subset of P). The structure of the neighborhoods is called the **swarm topology**, which can be represented by a graph. Usual topologies are: **fully connected** topology and **circle** topology.

Characteristics of particle i at iteration t:

 xi(t) … the**position** (a d-dimensional vector)

 pi(t) … the "historically"**best position**

 li(t) … the historicaly **best position of the neighboring particles**; for the fully connected topology it is the historically best known position of the entire swarm

 vi(t) … the**speed**; it is the step size between xi(t) and xi(t+1)

At the beginning of the algorithm, the particle positions are randomly initialized, and the velocities are set to 0, or to small random values.

Parameters of the algorithm:

 w(t) … **inertia weight**; a damping factor, usually decreasing from around 0.9 to around 0.4 during the computation

 φ1, φ2 … **acceleration coefficients**; usually between 0 and 4.

Update of the speed and the positions of the particles

Many versions of the particle speed update exist, for example:

vi(t+1)=w(t)vi(t)+ φ1u1(pi(t)-xi(t))+ φ2u2(li(t)-xi(t)).

The symbols u1 and u2 represent random variables with the U(0,1) distribution. The first part of the velocity formula is called "**inertia**", the second one "**the cognitive** (personal)

component", the third one is "**the social** (neighborhood) **component**". Position of particle i changes according to

xi(t+1)= xi(t)+vi(t+1).

Stopping rule

The algorithm is terminated after a given number of iterations, or once the fitness values of the particles (or the particles themselves) are close enough in some sense.

PSO variants

There is a plethora of different versions of PSOs, which usually modify the formula for the change of velocity (e.g., instead of u1 and u2 they use diagonal matrices **U**1 and **U**2, in other variants they use no inertia, but enforce an upper **limit on the particle speed**, there is the so-called "**fully informed**" PSO, and there is also a popular modification using a "**constriction coefficient**"). There exist versions of the PSO for constrained optimization, for discrete optimization, and for multi-objective optimization.

5.4. ADVANTAGES AND DISADVANTAGES

- The fitness function can be non-differentiable (only values of the fitness function are used). The method can be applied to optimization problems of large dimensions, often producing quality solutions more rapidly than alternative methods.

- There is no general convergence theory applicable to practical, multidimensional problems. For satisfactory results, tuning of input parameters and experimenting with various versions of the PSO method is sometimes necessary. Stochastic variability of the PSO results is very high for some problems and some values of the parameters. Also, some versions of the PSO method depend on the choice of the coordinate system.

CHAPTER 6

PSO IMPLEMENTATION FOR MINIMAL TRANSMISSION POWER

6.1. INTRODUCTION

Some of the difficulties to design WSNs include memory and energy constraints, limited capabilities and bandwidth unavailability. Most of these issues can be modeled as optimization problems, allowing met heuristic approaches to be used on their solution. PSO is one of the options for solving these problems in WSNs from a centralized point of view, and some of its applications are summarized next.

Several solutions focus on the energy saving problem. For instance, Latiff *et al.* proposed a clustering algorithm that uses PSO to achieve an energy-efficient network management. This is carried out by PSO through the election of cluster heads (CHs) according to their remaining energy, which is proven to significantly increase the nodes lifetime. Furthermore, Latiff *et al.* proposed an energy-efficient protocol for the movement of mobile base station using a Particle Swarm Optimization method in order to improve energy efficiency, lifetime and data delivery in WSN networks. Considering the mobility of the base station, the objective was to determine the path and the locations of a base station to stop for data collection in such a way that all sensor nodes can send the data to the base station via single hop communications. The PSO algorithm was applied as the optimization method when selecting the optimal sites for the base station to visit based on the distance between the base station and sensor nodes. Results shown that the proposed protocol can extend the network lifetime significantly compared to other protocol in the literature and it is also able to increase the data delivery at the base station compared to other protocols.

In another approach, particle swarm optimization has been applied to a routing protocol in order to also achieve energy efficiency. The work by Sarangi and Thankchan shows how PSO is able to calculate cheaper energy routes, outperforming the genetic algorithm.

Moreover, nodes deployment to maximize coverage has been considered by Jain and Sharma. In their proposal, a PSO algorithm is used to modify a previous method, increasing an area coverage without spending more nodes.

Regarding power adjustment carried out by the PSO algorithm, to the best of the authors' knowledge, Ho et al. have proposed a joint optimization of BERs, determination of an Unmanned Aerial Vehicle (UAV) route, and network energy consumption. The last objective includes changing the nodes transmission power but, since the considered scenarios are not similar, that proposal cannot be compared to the one presented in this work yet.

In this work, however, cluster heads will not be considered yet since a single cluster will be studied. Moreover, this research proposes to study sensors connectivity, providing the foundation to consider nodes deployment and energy-efficient routing in future iterations.

6.2. PROPOSED PSO ALGORITHM

N particles will be considered, each representing its corresponding node transmission power. The fitness function of the PSO algorithm will determine if all nodes are connected according to the values of each node transmission power. If all the nodes are connected, the fitness function will return the sum of the nodes power. Otherwise, it will return an infinite value.

This method is shown in Algorithm 2.

⊳Initialize the parameters of algorithm: the factors c1 and c2; the values of inertia factor [ωMin; ωMax]; the maximum number of steps tmax; the swarm size N and the constraints [xMin; xMax] and [vMin; vMax]

Algorithm. pso model:

```
procedure NodePowerOptimization
for i ∈ {1,2,...,N} do
        x_i ← rand(xMin, xMax)
        v_i ← rand(0,1)
end for
for i ∈ {1,2,...,N} do
    fitness_i ← f(x_i, position_scenario)
    f(pBest_i) ← fitness_i ⊳ individualfitness
    pBest_i ← x_i
end for
f(gBest)←min(f(pBest))⊳ global fitness
gBest ←pBest

for t ∈ {1,2,...,t_max} do
    ω ←ωMin-( (ωMax-ωMin) / tmax)*t
    for i ∈ {1,2,...,N} do
        ⊳Update speed of particle i according to Equation (1)
          for j ∈ {1,2,...,d} do
            if (v_ij>vMax)
                v_ij←vMax;
            end if
            if (v_ij<vMin)
                v_ij←vMin;
            end if
          end for
    end for

    for i ∈ {1,2,...,N} do
        ⊳Update position of particle i according to Equation (2)
          for j ∈ {1,2,...,d} do
            if (x_ij>xMax)
```

```
                $x_{ij}$←xMax;
            end if
            if ($x_{ij}$<xMin)
                $x_{ij}$←xMin;
            end if
        end for
    end for

    for i ∈ {1,2,...,N} do
        fitness_i ← f(x_i, position_scenario)
        if(fitness_i < f(pBest_i))then
        f(pBest_i) ← fitness_i
        pBest_i ← x_i
    end if
        if ( f(pBest_i) < f(gBest))then
        f(gBest) ← f(pBest_i)
        gBest ← pBest_i
        end if
    end for
end for
```

The values of the node transmission power that resulting in the least sum (energy saving) will be obtained from the values stored in *gBest*.

end procedure

CHAPTER-7

RESULTS

DEPLOYMENT:

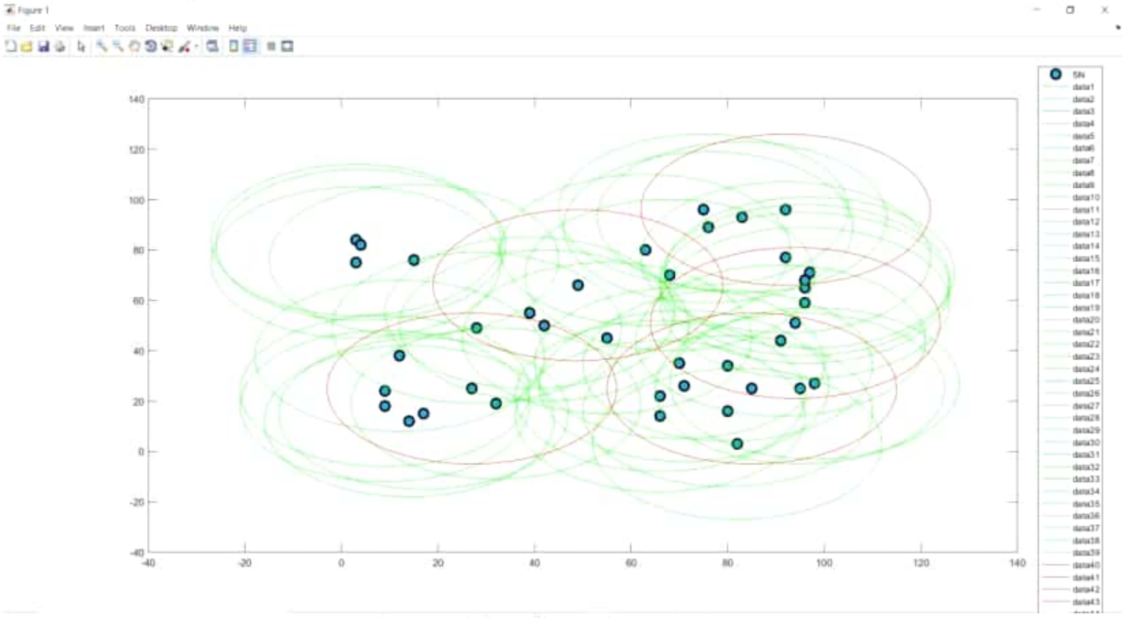

Figure 7.1. DEPLOYMENT

LEACH:

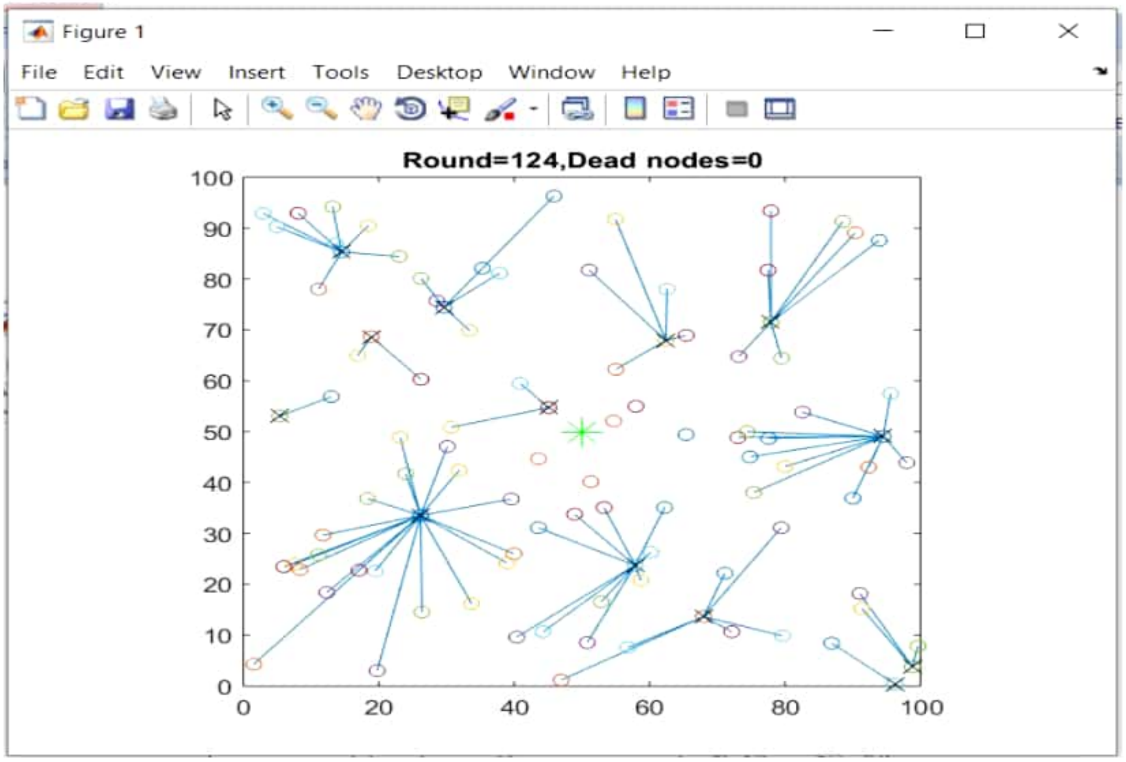

Figure 7.2. LEACH

PSO:

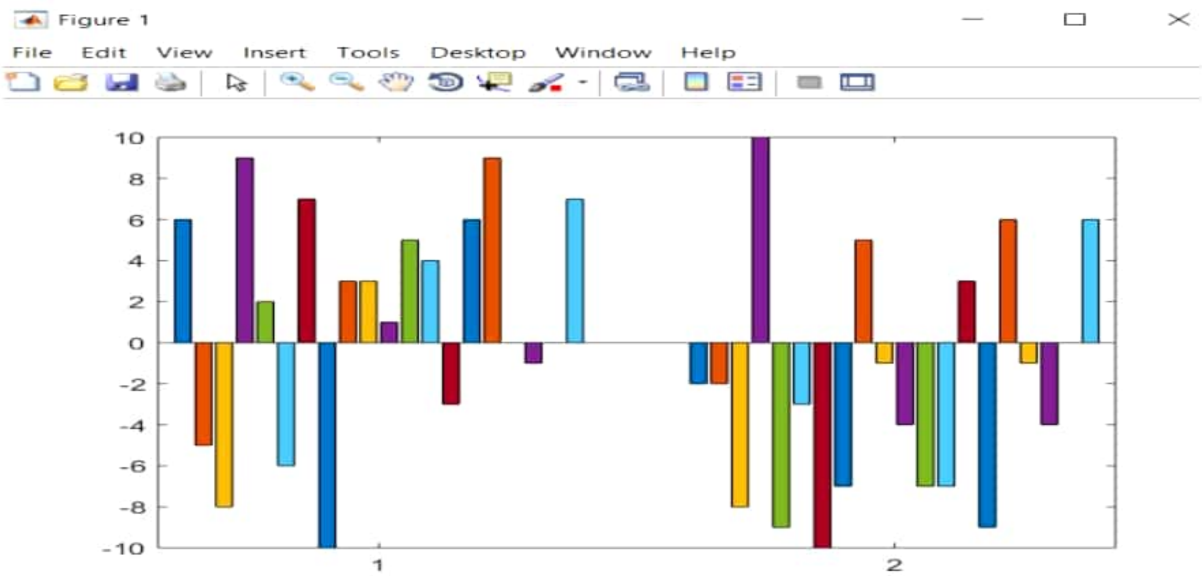

Figure 7.3.PSO SPACE 1 LEVELS

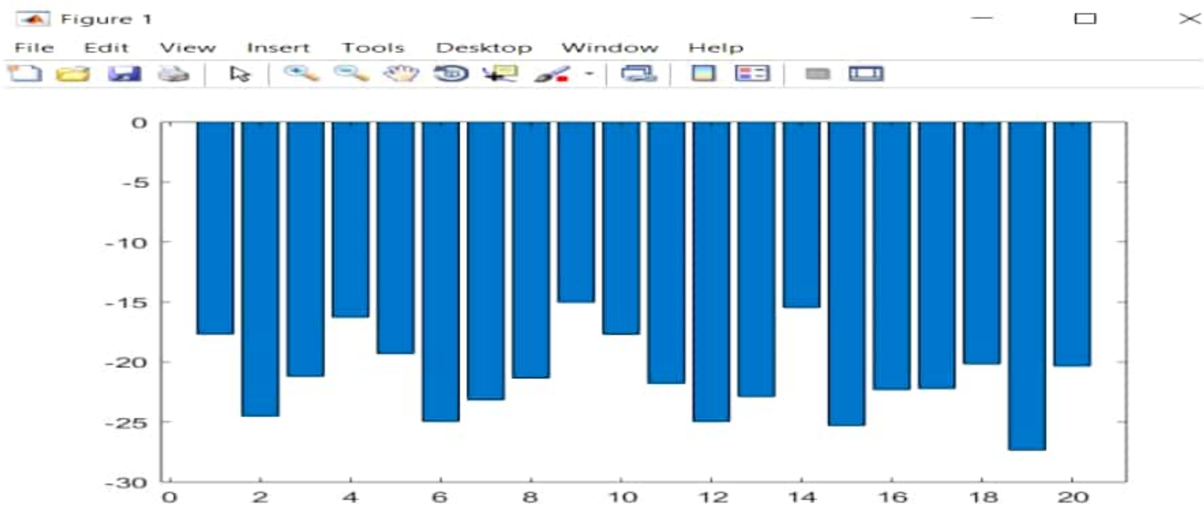

Figure 7.4.PSO ENERGY LEVELS

```
Landspace = space1

#################################################################
#################
                                    Test = 1
#################################################################
#################
configurations of PSO
Number of particles = 30.000000
Maximum number of steps  = 500.000000
Size of the problem = 20.000000
Coefficient C1 = 2.000000
Coefficient C2 = 2.000000
Inaction (wmax) = 0.900000 / (wmin) = 0.400000

power of each node

Node  1= -18.121238
Node  2= -23.031830
Node  3= -21.335420
Node  4= -16.288606
Node  5= -18.065299
Node  6= -21.546926
Node  7= -19.863372
Node  8= -20.772208
Node  9= -15.926684
Node  10= -18.144255
Node  11= -18.347691
Node  12= -27.992580
Node  13= -27.964423
Node  14= -15.605931
Node  15= -24.389304
Node  16= -22.309206
Node  17= -27.417277
Node  18= -20.246489
Node  19= -24.083106
Node  20= -20.125192


timetaken: 3.037046e+00
```

```
Landspace = space2

#############################################################
##################
                            Test = 1
#############################################################
##################
configurations of PSO
Number of particles = 30.000000
Maximum number of steps  = 500.000000
Size of the problem = 20.000000
Coefficient C1 = 2.000000
Coefficient C2 = 2.000000
Inaction (wmax) = 0.900000 / (wmin) = 0.400000



power of each node

Node  1= -18.121238
Node  2= -23.031830
Node  3= -21.335420
Node  4= -16.288606
Node  5= -18.065299
Node  6= -21.546926
Node  7= -19.863372
Node  8= -20.772208
Node  9= -15.926684
Node  10= -18.144255
Node  11= -18.347691
Node  12= -27.992580
Node  13= -27.964423
Node  14= -15.605931
Node  15= -24.389304
Node  16= -22.309206
Node  17= -27.417277
Node  18= -20.246489
Node  19= -24.083106
Node  20= -20.125192


timetaken: 3.037046e+00
```

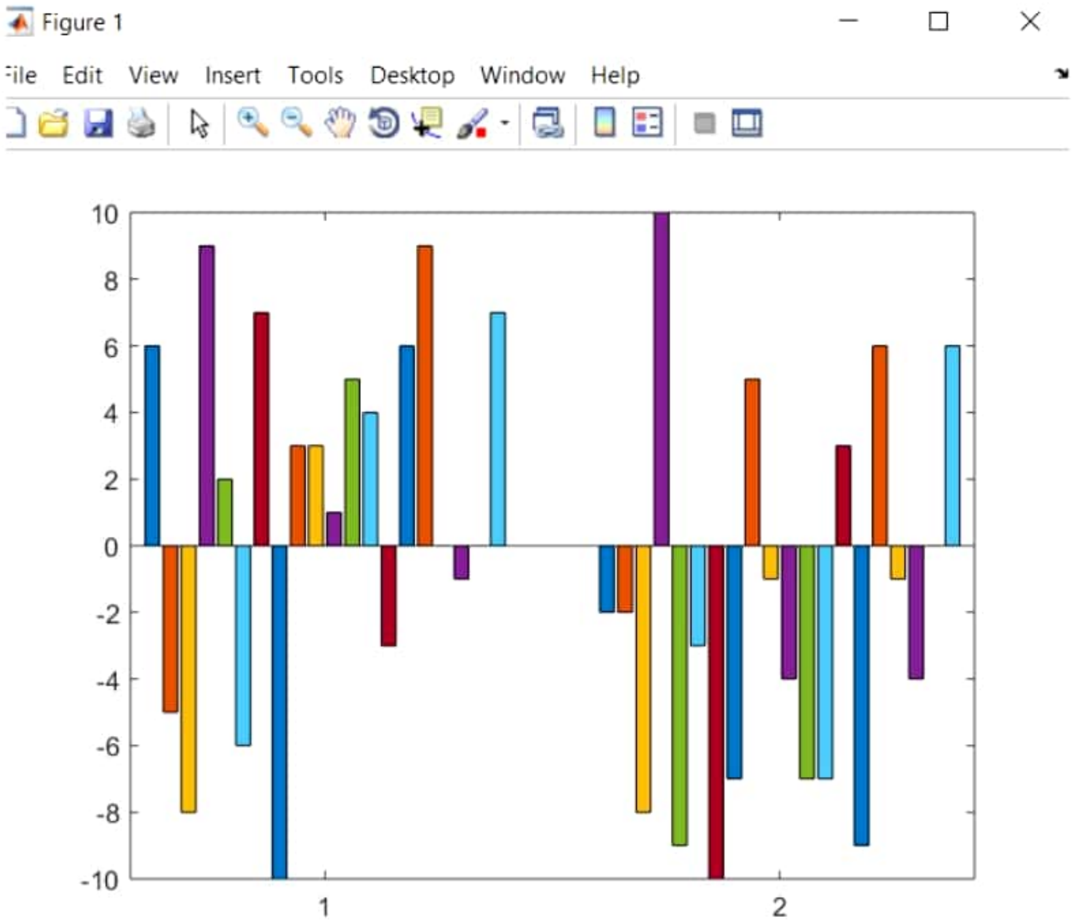

Figure 7.5.PSO SPACE 2 LEVELS

```
Landspace = space3

##########################################################
#################
                           Test = 1
##########################################################
#################
configurations of PSO
Number of particles = 30.000000
Maximum number of steps  = 500.000000
Size of the problem = 20.000000
Coefficient C1 = 2.000000
Coefficient C2 = 2.000000
Inaction (wmax) = 0.900000 / (wmin) = 0.400000
```

```
power of each node

Node  1= -16.804076
Node  2= -6.914861
Node  3= -28.840769
Node  4= -4.554207
Node  5= -1.161841
Node  6= -17.884274
Node  7= -10.343820
Node  8= -6.304830
Node  9= -22.835799
Node  10= -20.558685
Node  11= -13.782148
Node  12= -14.978147
Node  13= -8.147158
Node  14= -5.885420
Node  15= -12.929537
Node  16= -15.149033
Node  17= -19.511894
Node  18= -1.064325
Node  19= -13.438212
Node  20= -18.514719

timetaken: 2.910563e+00
```

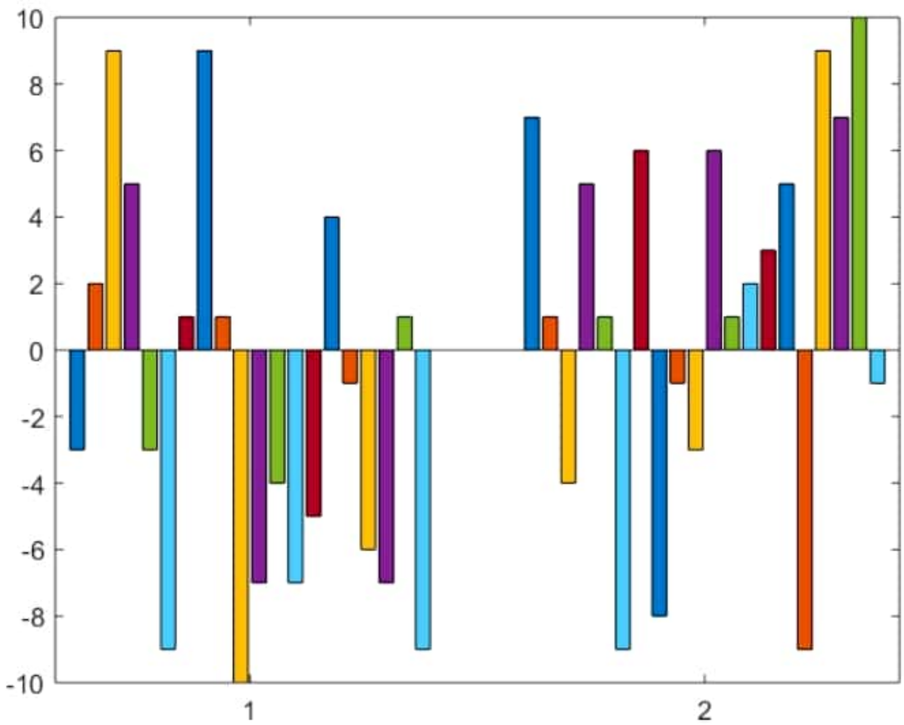

Figure 7.6.PSO SPACE 3 LEVELS

```
Landspace = space4

#################################################################
##################
                              Test = 1
#################################################################
##################
configurations of PSO
Number of particles = 30.000000
Maximum number of steps  = 500.000000
Size of the problem = 20.000000
Coefficient C1 = 2.000000
Coefficient C2 = 2.000000
Inaction (wmax) = 0.900000 / (wmin) = 0.400000
power of each node

Node  1= -21.340106
Node  2= -21.509586
Node  3= -28.167606
Node  4= -17.153547
Node  5= -20.296189
Node  6= -22.276665
Node  7= -23.892114
Node  8= -14.350406
Node  9= -24.175166
Node  10= -24.938333
Node  11= -27.718140
Node  12= -17.002399
Node  13= -9.161972
Node  14= -17.360955
Node  15= -23.168480
Node  16= -21.340022
Node  17= -20.911415
Node  18= -21.060363
Node  19= -20.705082
Node  20= -15.700941


timetaken: 2.985437e+00
```

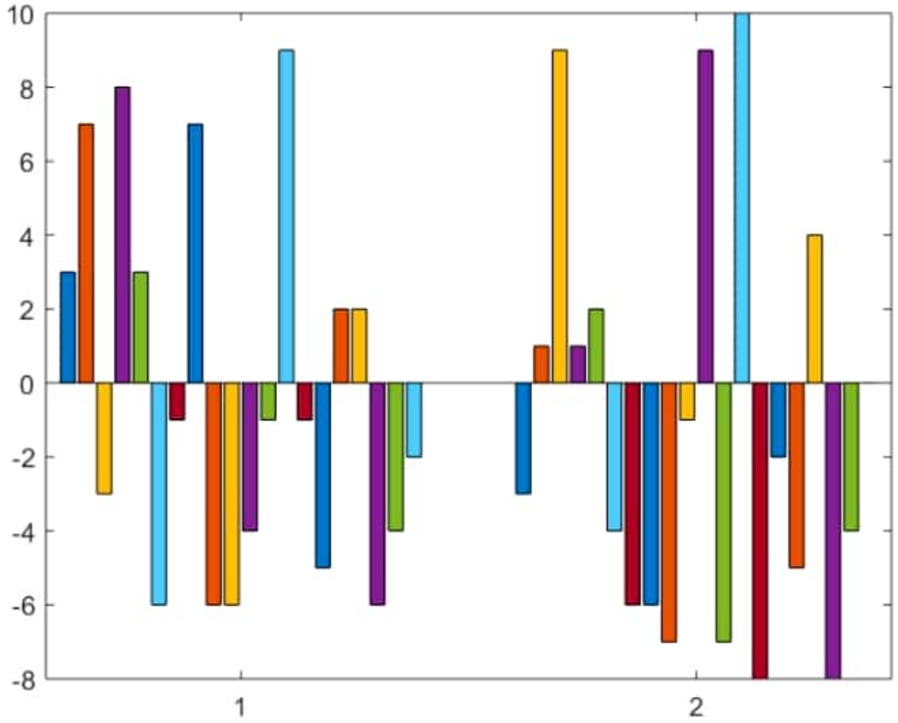

Figure 7.7.PSO SPACE 4 LEVELS

```
Landspace = space5

#################################################################
                                   Test = 1
#################################################################
configurations of PSO
Number of particles = 30.000000
Maximum number of steps   = 500.000000
Size of the problem = 20.000000
Coefficient C1 = 2.000000
Coefficient C2 = 2.000000
Inaction (wmax) = 0.900000 / (wmin) = 0.400000
power of each node

Node  1= -26.851060
Node  2= -5.019395
Node  3= -22.391034
Node  4= -21.508637
Node  5= -29.813196
Node  6= -15.293859
Node  7= -7.105997
Node  8= -16.547496
Node  9= -18.427525
```

```
Node  10= -12.977773
Node  11= -6.098343
Node  12= -8.608221
Node  13= -16.942057
Node  14= -24.415496
Node  15= -28.536744
Node  16= -23.515323
Node  17= -15.717123
Node  18= -16.629574
Node  19= -27.202144
Node  20= -29.221115
timetaken: 2.676099e+00
```

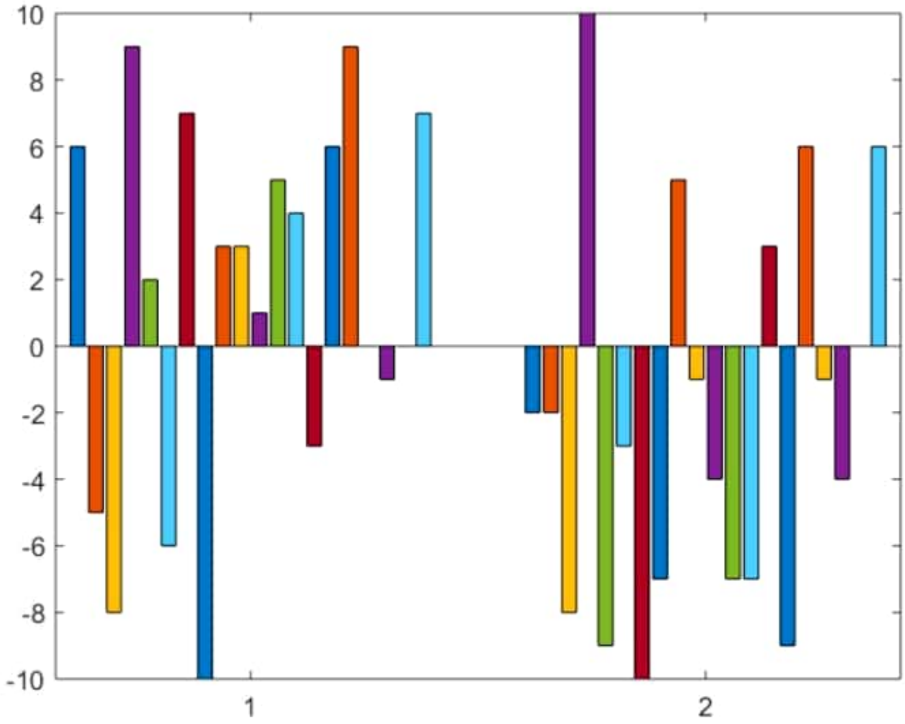

Figure 7.8:PSO SPACE 5 LEVELS

CHAPTER-8

CONCLUSIONS

- More sensor node become inter connected to make IOT real.
- WSN comprise energy-limited device to save energy become a new trend
- Cluster formation using LEACH algorithm.
- PSO method has been proposed to calculate different sensor nodes transmission power in order to save sensors energy while keeping all nodes connected.

CHAPTER-9

REFERENCES

1. Boukerche. Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks, John Wiley& Sons, Inc., 2009.
2. Manjeshwar and D. P. Agarwal, "APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, pp. 195-202.
3. Manjeshwar and D. P. Agarwal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," In 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, April 2001.
4. Paul, M. A. Matin," Optimal Geometrical Sink Location Estimation for Two Tiered Wireless Sensor Networks" IET Wireless Sensor Systems, vol.1, no.2, pp.74-84, June 2011,doi: 10.1049/iet-wss.2010.0073, IET UK.
5. Bharathidasan, A., Anand, V., Ponduru, S. (2001), Sensor Networks: An Overview, Department of Computer Science, University of California, Davis 2001. Technical Report
6. X. Cheng, DZ Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," ACM/Springer Journal of Wireless Networks, 14: pp. 347-355, 242, 2008.
7. M. Younis, and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks A survey," Ad Hoc Networks 6 pp. 621–655, 2008.
8. W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "*Energyefficient communication protocol for wireless sensor networks*", in: Proceeding of the Hawaii International Conference System Sciences, Hawaii, January 2000.
9. S. Lindsey, C.S. Raghavendra, "*PEGASIS: power efficient gathering in sensor information systems*", in: Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 2002.
10. Rajesh Patel, Sunil Pariyani, Vijay Ukani," *Energy and hroughput Analysis of Hierarchical Routing Protocol(LEACH) for Wireless Sensor Networks*", International Journal of Computer Applications Volume 20- No. 4 (April 2011).
11. Rodoplu, V., &Meng, T. H. (1999). *Minimum energy mobile wireless networks. IEEE Journal on Selected Areas in Communications*, 3, 1333– 1344. doi:10.1109/49.779917 Kulkarni, R. V., & Venayagamoorthy, G. K. (2011). *Particle swarm optimization in wireless-sensor networks: A brief survey. IEEE Transactions on Systems, Man and*

Cybernetics Part C: Applications and Reviews, 41, 262–267. doi:10.1109/TSMCC.2010.2054080

12. Latiff, N. A., Tsimenidis, C. C., & Sharif, B. S. (2007, September).*Energy-aware clustering for wireless sensor networks using particle swarm optimization. In Personal, Indoor and Mobile Radio Communications*, 2007. PIMRC 2007. *IEEE 18th International*

13. *Symposium on* (pp. 1-5). *IEEE.*

14. Latiff, N. A. A., Latiff, N. M. A. A., & Ahmad, R. B. (2011). Prolonging lifetime of wireless sensor networks with mobile base station using particle swarm optimization. In Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on (pp. 1-6). IEEE.

15. GUPTA, H. (2014).*Geographic Routing in Clustered Wireless Sensor Networks among Obstacles*, Sensors Journal, IEEE (Volume:PP , Issue: 99) doi:10.1109/JSEN.2014.2385734.

16. Heinzelman, W. R., Sinha, A., Wang, A., &Chandrakasan, A. P. (2000).*Energy-scalable algorithms and protocols for wireless microsensor networks*.2000 IEEE International Conference on Acoustics, Speech, and Signal Processing.Proceedings (Cat.No.00CH37100), 6. doi:10.1109/ICASSP.2000.860211

17. Mendes, L. D. P., Rodrigues, J. J. P. C., & Chen, M. (2010). *A cross-layer sleep and rate adaptation mechanism for slotted ALOHA wireless sensor networks*. In 2010 International Conference on Information and Communication Technology Convergence, ICTC 2010 (pp. 213–217). doi:10.1109/ICTC.2010.5674661

CHAPTER-10

SOURCE CODE

DEPLOYMENT:

```
clc;
clear;
mn=5; % malicious node
sn=39; % sensor node
x=randi([0,100],1,sn); % x coordinate of a sn
y=randi([0,100],1,sn); % y coordinate of a sn
z=randi([0,sn],1,mn); % malicious nodes index values (as per
percentage 20%)
plot(x,y,'mo',...
    'LineWidth',2,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor',[.17 0.66 .83],...
    'MarkerSize',10)
legend('SN')
for i=1:sn
      draw_circle1(x(i),y(i),30,'g')
end
for j=1:sn
    for k=1:mn
    if z(k)==j
    draw_circle1(x(j),y(j),30,'r')
    end
    end
end
grid on
figure(1) % Hold figure 1
hold on
```

LEACH:

```
clc;
clear;
close all;
warning off all;
tic;

%% Create sensor nodes, Set Parameters and Create Energy Model
%%%%%%%%%%%%%%%%%%%%%%%%% Initial Parameters
%%%%%%%%%%%%%%%%%%%%%%%
n=100;                                  %Number of Nodes in
the field
[Area,Model]=setParameters(n);          %Set Parameters
Sensors and Network

%%%%%%%%%%%%%%%%%%%%%%%%% configuration Sensors
%%%%%%%%%%%%%%%%%%%%
CreateRandomSen(Model,Area);            %Create a random
scenario
load Locations                          %Load sensor Location
Sensors=ConfigureSensors(Model,n,X,Y);
ploter(Sensors,Model);                  %Plot sensors

%%%%%%%%%%%%%%%%%%%%%%%%%% Parameters initialization
%%%%%%%%%%%%%%%%
countCHs=0;          %counter for CHs
flag_first_dead=0;   %flag_first_dead
deadNum=0;           %Number of dead nodes

initEnergy=0;        %Initial Energy
for i=1:n
      initEnergy=Sensors(i).E+initEnergy;
end

SRP=zeros(1,Model.rmax);    %number of sent routing packets
RRP=zeros(1,Model.rmax);    %number of receive routing packets
SDP=zeros(1,Model.rmax);    %number of sent data packets
RDP=zeros(1,Model.rmax);    %number of receive data packets

Sum_DEAD=zeros(1,Model.rmax);
CLUSTERHS=zeros(1,Model.rmax);
AllSensorEnergy=zeros(1,Model.rmax);

%%%%%%%%%%%%%%%%%%%%%%%%% Start Simulation
%%%%%%%%%%%%%%%%%%%%%%%%%
global srp rrp sdp rdp
srp=0;          %counter number of sent routing packets
rrp=0;          %counter number of receive routing packets
sdp=0;          %counter number of sent data packets
rdp=0;          %counter number of receive data packets
```

```
%Sink broadcast start message to all nodes
Sender=n+1;      %Sink
Receiver=1:n;    %All nodes
Sensors=SendReceivePackets(Sensors,Model,Sender,'Hello',Receiv
er);

% All sensor send location information to Sink .
 Sensors=disToSink(Sensors,Model);
% Sender=1:n;     %All nodes
% Receiver=n+1;   %Sink
%
Sensors=SendReceivePackets(Sensors,Model,Sender,'Hello',Receiv
er);

%Save metrics
SRP(1)=srp;
RRP(1)=rrp;
SDP(1)=sdp;
RDP(1)=rdp;

%% Main loop program
for r=1:1:Model.rmax

%%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialization
%%%%%%%%%%%%%%%%%%%%%
    %This section Operate for each epoch
    member=[];              %Member of each cluster in per
period
    countCHs=0;             %Number of CH in per period
    %counter for bit transmitted to Bases Station and Cluster
Heads
    srp=0;          %counter number of sent routing packets
    rrp=0;          %counter number of receive routing packets
    sdp=0;          %counter number of sent data packets to
sink
    rdp=0;          %counter number of receive data packets by
sink
    %initialization per round
    SRP(r+1)=srp;
    RRP(r+1)=rrp;
    SDP(r+1)=sdp;
    RDP(r+1)=rdp;
    pause(0.001)    %pause simulation
    hold off;       %clear figure

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
    Sensors=resetSensors(Sensors,Model);
    %allow to sensor to become cluster-head. LEACH Algorithm
    AroundClear=10;
    if(mod(r,AroundClear)==0)
```

```
        for i=1:1:n
            Sensors(i).G=0;
        end
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% plot sensors
%%%%%%%%%%%%%%%%%%%%%%%
    deadNum=ploter(Sensors,Model);

    %Save r'th period When the first node dies
    if (deadNum>=1)
        if(flag_first_dead==0)
            first_dead=r;
            flag_first_dead=1;
        end
    end

%%%%%%%%%%%%%%%%%%%%%%% cluster head election
%%%%%%%%%%%%%%%%%%%
    %Selection Candidate Cluster Head Based on LEACH Set-up
Phase
    [TotalCH,Sensors]=SelectCH(Sensors,Model,r);

    %Broadcasting CHs to All Sensor that are in Radio Rage CH.
    for i=1:length(TotalCH)

        Sender=TotalCH(i).id;
        SenderRR=Model.RR;
        Receiver=findReceiver(Sensors,Model,Sender,SenderRR);

Sensors=SendReceivePackets(Sensors,Model,Sender,'Hello',Receiv
er);

    end


    %Sensors join to nearest CH
    Sensors=JoinToNearestCH(Sensors,Model,TotalCH);

%%%%%%%%%%%%%%%%%%%%%% end of cluster head election phase
%%%%%%

%%%%%%%%%%%%%%%%%%%%%%% plot network status in end of set-up
phase

    for i=1:n

        if (Sensors(i).type=='N' &&
Sensors(i).dis2ch<Sensors(i).dis2sink && ...
                Sensors(i).E>0)
```

```
            XL=[Sensors(i).xd ,Sensors(Sensors(i).MCH).xd];
            YL=[Sensors(i).yd ,Sensors(Sensors(i).MCH).yd];
            hold on
            line(XL,YL)

        end

    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% steady-state phase
%%%%%%%%%%%%%%%%%
    NumPacket=Model.NumPacket;
    for i=1:1:1%NumPacket

        %Plotter
        deadNum=ploter(Sensors,Model);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% All sensor send data packet
to  CH
        for j=1:length(TotalCH)

            Receiver=TotalCH(j).id;
            Sender=findSender(Sensors,Model,Receiver);

Sensors=SendReceivePackets(Sensors,Model,Sender,'Data',Receive
r);

        end

    end


%%%%%%%%%%%% send Data packet from CH to Sink after Data
aggregation
    for i=1:length(TotalCH)

        Receiver=n+1;               %Sink
        Sender=TotalCH(i).id;       %CH

Sensors=SendReceivePackets(Sensors,Model,Sender,'Data',Receive
r);

    end
%%% send data packet directly from other nodes(that aren't in
each cluster) to Sink
    for i=1:n
        if(Sensors(i).MCH==Sensors(n+1).id)
            Receiver=n+1;               %Sink
            Sender=Sensors(i).id;       %Other Nodes
```

```
Sensors=SendReceivePackets(Sensors,Model,Sender,'Data',Receive
r);
        end
    end


%% STATISTICS

    Sum_DEAD(r+1)=deadNum;

    SRP(r+1)=srp;
    RRP(r+1)=rrp;
    SDP(r+1)=sdp;
    RDP(r+1)=rdp;

    CLUSTERHS(r+1)=countCHs;

    alive=0;
    SensorEnergy=0;
    for i=1:n
        if Sensors(i).E>0
            alive=alive+1;
            SensorEnergy=SensorEnergy+Sensors(i).E;
        end
    end
    AliveSensors(r)=alive; %#ok

    SumEnergyAllSensor(r+1)=SensorEnergy; %#ok

    AvgEnergyAllSensor(r+1)=SensorEnergy/alive; %#ok

    ConsumEnergy(r+1)=(initEnergy-SumEnergyAllSensor(r+1))/n;
%#ok

    En=0;
    for i=1:n
        if Sensors(i).E>0
            En=En+(Sensors(i).E-AvgEnergyAllSensor(r+1))^2;
        end
    end

    Enheraf(r+1)=En/alive; %#ok

    title(sprintf('Round=%d,Dead nodes=%d', r+1, deadNum))

   %dead
   if(n==deadNum)

       lastPeriod=r;
       break;
```

```
    end

end% for r=0:1:rmax

disp('End of Simulation');
toc;
disp('Create Report...')

filename=sprintf('leach%d.mat',n);

%% Save Report
save(filename);
```

PSO:

```
clear all;
for cen=1:10
landspace = strcat('space',int2str(cen));
load(landspace)
FileName = strcat(landspace,'.t2');
fid = fopen(FileName,'wt');
fprintf(fid,'################## initial training
##################\n\n');
fprintf(fid,'Landspace = %s\n\n',landspace);
for t=1:5
tic
fprintf(fid,'##################################################
##########################\n');
fprintf(fid,'                                        TESTE = %d
\n',t);
fprintf(fid,'##################################################
##########################\n');
n = 30;          % Size of cluster
num_steps = 500; % maximum number of steps
dim = size(X,2);    % Size of particles
wmax=0.9;  %The initial inertia factor
wmin=0.4;  %The final inertia factor
c1=2;      %coefficient of acceleration
c2=2;      %coefficient of acceleration

%Sets the search space limit
pMin = -30;
pMax = 0;

%Sets the speed limit
vMax = 5;
vMin = -vMax;

fprintf(fid,'configurations of PSO\n');
fprintf(fid,'Number of particles = %f\n',n);
fprintf(fid,'Maximum number of steps  = %f\n',num_steps);
fprintf(fid,'Size of the problem = %f\n',dim);
fprintf(fid,'Coefficient C1 = %f\n',c1);
fprintf(fid,'Coefficient C2 = %f\n',c2);
fprintf(fid,'Inaction (wmax) = %f / (wmin) = %f
\n\n',wmax,wmin);
fprintf(fid,'pMax = %f / pMin = %f\n',pMax,pMin);
fprintf(fid,'vMax = %f / vMin = %f\n',vMax,vMin);
fitness=0*ones(n,num_steps);
current_fitness =0*ones(n,1);


vector_power = pMin + (pMax - pMin)*rand(dim,n);
```

```
 velocity = rand(dim,n);

for i=1:n
current_fitness(i) = connect_report(X,vector_power(:,i)) ;
end

pbest_fitness  = current_fitness ;
pbest_position  = vector_power ;

[gbest_fitness,g] = min(pbest_fitness) ;
for i=1:n
    gbest_position(:,i) = pbest_position(:,g) ;
end

for iter=1:num_steps

w = wmax - ((wmax-wmin)/num_steps)*iter;

for i=1:n
    velocity(:,i) = w*velocity(:,i) +
c1*(rand*(pbest_position(:,i)-vector_power(:,i))) +
c2*(rand*(gbest_position(:,i)-vector_power(:,i)));
end

for k=1:n
    index = velocity(:,k)>vMax;
    velocity(index,k)=vMax;

    index = velocity(:,k)<vMin;
    velocity(index,k)=vMin;
end

vector_power = vector_power + velocity;

for k=1:n
    index = vector_power(:,k)>pMax;
    vector_power(index,k)=pMax;

    index = vector_power(:,k)<pMin;
    vector_power(index,k)=pMin;
end

for i=1:n
        current_fitness(i) =
connect_report(X,vector_power(:,i)) ;
end

for i=1:n
    if current_fitness(i) < pbest_fitness(i)
        pbest_fitness(i)  = current_fitness(i);
```

```
        pbest_position(:,i) = vector_power(:,i);
    end
end

[current_gbest_fitness,g] = min(pbest_fitness);

if current_gbest_fitness < gbest_fitness
    gbest_fitness = current_gbest_fitness;
    for i=1:n
        gbest_position(:,i) = pbest_position(:,g);
    end
end

 if(iter==1 || rem(iter,50)==0)
   sprintf('Iteration - %4d | Count =
%f\n',iter,gbest_fitness)
   fprintf(fid,'Iteration - %4d |Count =
%f\n',iter,gbest_fitness);
 end

end

gbest_fitness;
d=gbest_position(:,1);

fprintf(fid,'\n\n');
fprintf(fid,'power of each node\n\n');
for i=1:dim
    fprintf(fid,'Node  %i= %f ',i, d(i));
    fprintf(fid,'\n');
end
fprintf(fid,'\n\n');

timetaken = toc;
fprintf(fid,'timetaken: %d\n\n',timetaken);

end

fclose(fid);

end
```