A Major Project Report On

ADAPTIVE ENERGY-EFFICIENT PROTOCOL (AEFP)

Submitted in partial fulfillment of the requirements

For the award of the degree of

MASTER OF TECHNOLOGY

IN

SOFTWARE TECHNOLOGY

By

BISWAJEET MOHANTY

(Roll No. 2K11/ST/06)

Under the guidance of

Mr. R.K. Yadav

Asst. Professor, Department of COE

Department of Computer Engineering

Delhi Technological University, Delhi

DELHI TECHNOLOGICAL UNIVERSITY

CERTIFICATE

This is to certify that the project report entitled is a bona fide record of work carried out by **BISWAJEET MOHANTY (2K11/ST/06)** under my guidance and supervision, during the academic session 2014 in partial fulfillment of the requirement for the degree of Master of Technology in Software Technology from Delhi Technological University, Delhi.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

R.K. Yadav

Asst. Professor

Department of Computer Engineering

Delhi Technological University

Delhi

DELHI TECHNOLOGICAL UNIVERSITY

ACKNOWLEDGEMENT

With due regards, I hereby take this opportunity to acknowledge a lot of people who have supported me with their words and deeds in completion of my research work on **AEFP** as part of Master of Technology in Software Technology.

To start with I would like to thank the almighty for being with me in each and every step of my life. Next, I thank my parents and family for their persistent support.

I would like to express my deepest sense of gratitude and indebtedness to my guide and motivator, **R.K. Yadav**, Professor, Department of Computer Engineering, Delhi Technological University for her valuable guidance and support in all the phases from conceptualization to final completion of the project.

I humbly extend my grateful appreciation to my friends whose moral support made this project possible.

Last but not the least; I would like to thank all the people directly and indirectly involved in successfully completion of this project.

Biswajeet Mohanty

Roll No. 2K11/ST/06

TABLE OF CONTENTS

Chapter 1

Chapter 2

Chapter 3

List of Fig.s

ABSTRACT

This report is concerned with designing and implementation of energy efficient sensor network as generally, Sensor Network is deployed to continuously perform in low energy environment. Routing Protocol is the key for efficiency of a sensor Network.

Wireless Sensor Network (WSN) consists of irreplaceable nodes which are equipped with limited energy resources. Necessity of power consumption becomes a prior importance for various pervasive and ubiquitous applications.

Wireless Sensor Network (WSN) consists of several nodes in count of hundred or thousand operating in remote location and harsh environment. Recent advancements in the field of digital signal processors, short-range radio electronics, and micro electromechanical systems (MEMS) based sensor technology and low power RF designs have enabled the development of inexpensive low power sensors with significant computational capability.

A cluster of wirelessly communicating cells, to cope with scalability problems, if combined with in-network data aggregation with help of an energy efficient routing algorithm i.e. Hierarchical routing in which sensors organize themselves into clusters and each cluster has a cluster head where the low energy nodes are used to perform the sensing in the proximity of the phenomenon, can be deployed as an energy efficient sensor Network.

Introduction

Wireless sensor networks (WSNs) are an important technology for large-scale monitoring, in which a set of sensor nodes that can vary from few hundreds to thousands in numbers, with capabilities of sensing, establishing wireless communication between each other and doing computational and processing operations by providing sensor measurements at high temporal and spatial resolution.

Sensor network is deployed in random fashion (e.g., dropped from an airplane) or planted manually (e.g., fire alarm sensors in a facility). For example, in a disaster management application, a large number of sensors can be dropped from a helicopter. Networking these sensors can assist rescue operations by locating survivors, identifying risky areas, and making the rescue team more aware of the overall situation in the disaster area .Scalability and temporal processing capability of Sensor networks make WSN usable for variety of applications and systems with vastly varying requirements and characteristics. Sensor nodes can be used in applications involving in Military environment, Disaster management, Habitat monitoring, Medical and health care, Industrial fields, Home networks, detecting chemical, biological, radiological, nuclear, and explosive material etc.

Fig. 1 shows the schematic diagram of sensor node components in which sensor nodes are shown as small circles. Basically, each sensor node comprises sensing, processing, transmission, mobilize, position finding system, and power units (some of these components are optional like the mobilize). The same Fig. shows the communication architecture of a WSN. Sensor nodes are usually scattered in a sensor field, which is an area where the sensor nodes are deployed.

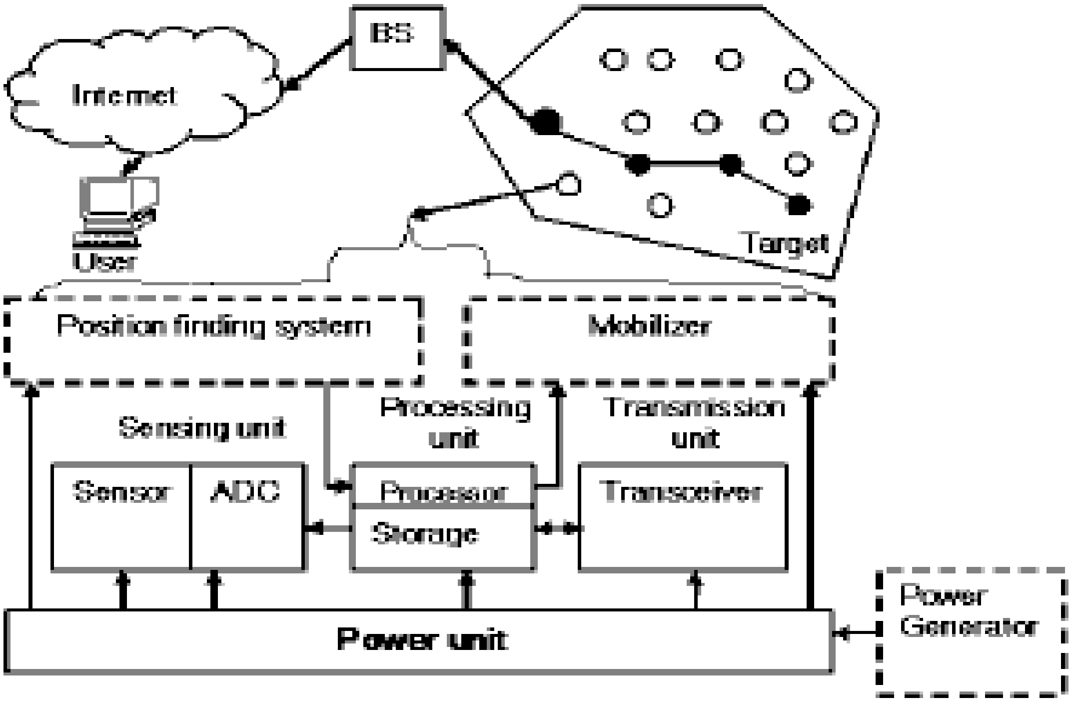


Fig.1 WSN

A wireless sensor network (WSN) of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

In general, classification of a WSN routing methodology can be done into two main categories;

- Based on network structure
- Based on the protocol operation.

Depending on the network structure, different routing schemes fall into this category. A sensor network can be non hierarchical or flat in the sense that every sensor has the same role and functionality. Therefore the connections between the nodes are set in short distance to establish the radio communication. Alternatively, a sensor network can be hierarchical or cluster-based hierarchical model, where the network is divided into clusters comprising of number of nodes.

Cluster head i.e. master node, within each respective cluster is responsible for routing the information to other cluster head. Another class of routing protocols is based on the location information of the sensor nodes either estimated on the basis of incoming signal strengths or obtained by small low-power GPS receivers or even by combination of the two previous methods. Location-based protocols use this information to reduce the latency and energy consumption of the sensor network.

This report is based on hierarchal schema to analyze the network performance.

Chapter 1. WSN Characteristics

In a WSN, sensor nodes are scattered randomly or with a static strategy in the network field varying from hundreds to thousands in number. There are many operational characteristics of WSNs, few can be listed as:

- Self Configuration: The topology of WSNs is not static and is supposedly changed with no traceable patterns. So the sensor nodes of the network have to be adaptable to such changes while keeping the power efficiency intact. Self configuration also has to handle the situations like node failures and node additions to the network or any other obstacles.
- Efficient Energy Usage: Power is very critical issue in increasing the performance of the network. So energy in a WSN should be expended in optimal manner like sensor nodes can switch off their sensors for a particular period of time and can switch on their transmitters only when any event occurs or after every frame time.
- Single Hop Communication: Traditional WSN protocol used single hop mechanism to send the information to the BS. In this mode of communication all the sensor senses the information and then transmits all the information it has sensed to the BS directly without the involvement of other sensor nodes on its own. This mode of communication is very inefficient and power consuming with lots of redundant information to the BS. Thus modern protocols rely on the multi hop mode of communication for better efficiency.

- Multi Hop Communication: In case of larger networks where the distance of the BS from the node is greater than the transmission range of the sensor node. The single hop communication fails. Multi hop communication uses packet forwarding to increase the efficiency of the network. The nodes send the information to the sink with the help of other intermediate sensor nodes which receives the information from the node and transfers the information to other node along the path of the BS or to BS itself. Multi hop communication saves transmission energy and is proved to be useful for energy efficiency and network lifetime.
- Automatic Load Balancing: Nodes in the network must decide who will be the parent node to transmit the information based on the hop count to the respective node, signal strength, link quality and present load quantity of the parent node. Automatic load balancing is dynamic in nature as the number of nodes in the network can run out of power anytime in between the network lifetime.

1.1 WSN Classification

Sensor networks can be employed to benefit a variety of applications in diverse environments. To benefit the applications, sensor nodes differ greatly in technical requirements for these various applications. Such networks require specific application oriented sensors to be developed and deployed. Although there are many specific properties of the sensors but some of the properties are common in these networks even for some very discrete applications. Like one group of application uses randomness for node distribution as the network area may be remote and isolated from human involvement and the remaining group of applications may use strategic static deployment. Hence a network can be classified based on the deployment strategy.

The sensor nodes are deployed densely in the desired environment. Depending upon the routing protocol of the WSN the sensors senses the environment continuously or after every fixed quantum. Then the sensor node transmits the sensed information either periodically or on the occurrence of some event to the BS. The transmission of information to the sensor nodes to the BS can be done in a variety of ways as proposed by many routing protocols to increase the efficiency of the network. Traditional way of transmitting the information directly from the sensor node to BS is very energy consuming and inefficient.

It uses multi-hop transmissions to transmit the information to the BS which distributes the energy expended amongst many nodes and hence prolongs the lifetime of whole of the network. Use of such mechanisms and development of energy efficient routing protocol is a need for the proficient use of WSNs. One other class of routing protocols in which lot of research has been done is clustered routing protocols. Clustered routing protocols are used in many WSN applications and is proved efficient than the traditional routing models. These protocols are very different from protocols used for wireless ad-hoc networks (WANETs) because of the differences amongst them which can be listed as:

- Network Density: The density of sensor nodes in WSNs is very high and the nodes are close to each other as compared to WANETs. Also the size of the nodes in WSNs may be even smaller than coins whereas for WANET nodes are large like laptops, cellular phones etc.
- Network Size: Number of nodes in WSNs can vary from hundreds to thousands but for WANETs number of nodes is lesser and can be up to few hundred at max.
- Frequency of Topology Change: Topology changes quite often in WSNs because of node failures, addition of new nodes, node movements and environmental interference. Topology may change as fast as in milliseconds in WSNs and network has to adapt with the change in the topology. While in WANETs nodes will join the network after request and can leave the network after some time and usually topology change is after a larger period of time as compared to WSNs.
- Failure of Nodes: In WSNs nodes are employed in remote areas which are sometimes isolated and from the human involvements like in disaster areas. The nodes once deployed cannot be replaced or recharged by the changing requirements and can be damaged from the environmental interferences. But in WANETs node can have rechargeable batteries and they are not imposed to difficult conditions as sensor nodes in WSNs.
- Communication Paradigm: In WANETs point to point communication is done in order establish communication between nodes whereas in WSNs, nodes broadcast in order to communicate with each other.
- Limited Resources: Memory storage of sensor nodes in WSNs is few kilobytes and for WANETs nodes can have gigabytes of storage. Power as mentioned is very limited in WSNs while WANETs can have rechargeable batteries for the nodes.

- Identification of Node: In WSNs it is very difficult to maintain a global identifier for the identification of the nodes because of high number of sensor nodes in a network. Also the nodes in WSNs are deployed in remote areas and hence can be damaged or moved outside network area frequently. The nodes of WANETs consists of unique identifier like IP (Internet Protocol) addresses.

1.2 Sensor Node Architecture

Sensor nodes are the outcome of the recent advances in low power highly integrated digital electronics and micro-electro-mechanical systems. The sensing circuitry is capable of measuring ambient conditions related to surroundings of the sensor which is then transformed into an electric signal.

Sensors are hardware devices that produce a measurable response to a change in a physical condition like temperature or pressure. Sensors measure physical data of the parameter to be monitored. The continual analog signal produced by the sensors is digitized by an analog-to-digital converter and sent to controllers for further processing. A sensor node should be small in size, consume extremely low energy, operate in high volumetric densities, be autonomous and operate unattended, and be adaptive to the environment. As wireless sensor nodes are typically very small electronic devices, they can only be equipped with a limited power source of less than 0.5-2 ampere-hour and 1.2-3.7 volts.

Sensors are classified into three categories: passive, Omni-directional sensors; passive, narrow-beam sensors; and active sensors. Passive sensors sense the data without actually manipulating the environment by active probing. They are self powered; that is, energy is needed only to amplify their analog signal. Active sensors actively probe the environment, for example, a sonar or radar sensor, and they require continuous energy from a power source. Narrow-beam sensors have a well-defined notion of direction of measurement, similar to a camera. Omni-directional sensors have no notion of direction involved in their measurements.

The overall theoretical work on WSNs works with passive, Omni-directional sensors. Each sensor node has a certain area of coverage for which it can reliably and accurately report the particular quantity that it is observing. Several sources of power consumption in sensors are: signal sampling and conversion of physical signals to electrical ones, signal conditioning, and analog-to-digital conversion. Spatial density of sensor nodes in the field may be as high as 20 nodes per cubic meter.

Sensor nodes often make use of ISM band, which gives free radio, spectrum allocation and global availability. The possible choices of wireless transmission media are radio frequency (RF), optical communication (laser) and infrared. Lasers require less energy, but need line-of-sight for communication and are sensitive to atmospheric conditions. Infrared, like lasers, needs no antenna but it is limited in its broadcasting capacity. Radio frequency-based communication is the most relevant that fits most of the WSN applications. WSNs tend to use license-free communication frequencies: 173, 433, 868, and 915 MHz; and 2.4 GHz. The functionality of both transmitter and receiver are combined into a single device known as a transceiver. Transceivers often lack unique identifiers. The operational states are transmitting, receive, idle, and sleep. Current generation transceivers have built-in state machines that perform some operations automatically.

Most transceivers operating in idle mode have a power consumption almost equal to the power consumed in receive mode. Thus, it is better to completely shut down the transceiver rather than leave it in the idle mode when it is not transmitting or receiving. A significant amount of power is consumed when switching from sleep mode to transmit mode in order to transmit a packet.

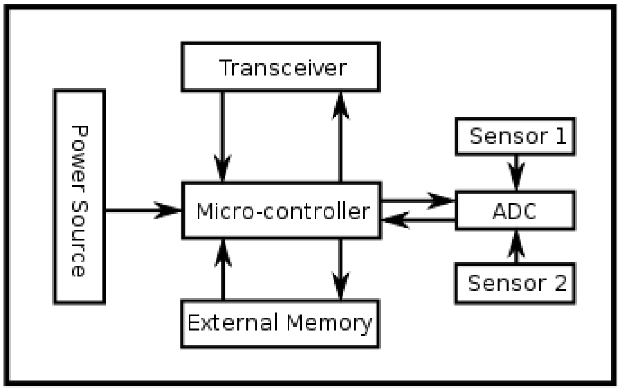


Fig1.1 Sensor Node Architecture

1.3 Sensor Network Protocol Stack

The sensor network protocol stack is different from the standard TCP/IP and along with the layers used in traditional protocols it has additional planes to handle the issues in the sensor nodes.

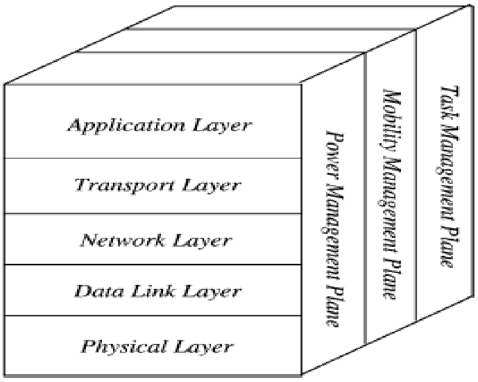


Fig. 1.2 : Sensor Network Protocol Stack

The front plane in the above Fig. has five layers which work as traditional protocols. The application layer is involved in usage and development of application software(s) based on the specified sensing tasks. The flow of data in sensor networks is managed by transport layer of the protocol stack. The transport layer provides the data to the network layer and it is the responsibility of the network layer to route the data as desired. Data link layer handles the noise and mobility of the nodes in the network and are power aware. Data link layer is also responsible for reducing the collision with the neighbors' broadcast. Transmission, modulation and receiving techniques are handled by physical layer of the protocol stack. The additional planes of power, mobility and task management are specially created to cater the needs of sensor node related to battery, movement and task distribution. Main objective of these planes is to carry out the collaborative sensing task as desired with least power consumption. The power management plane (PMP) has the responsibility to decide how a node will use its power like when it has to switch on their sensors and when they have to switch off them. Power management plane also takes care of when to transmit the information and when not to. Mobility management plane (MMP) is responsible to know the route back to the BS in case of movement of the sensor nodes from their initial positions and inform neighbors about the newly moved sensor node. With the updated knowledge of neighbors now they can contribute and use the power of the nodes efficiently.

Task management plane (TMP) performs the balancing of sensing tasks amongst the sensor nodes. It is not necessary that all the sensor nodes in the vicinity should sense and transmit the information. Thus TMP takes such decisions as per the application used in WSNs. So the additional planes are very essential for the objective of WSNs as without them a node will have no capability and will act as an individual transmitting the unwanted information and wasting the constrained power of the sensor node. These planes are essential for simple computations at the sensor nodes as per the changes in power, mobility and required tasks efficiently by the sensor nodes.

1.4 WSNs Application Areas

With the increase in growth and feasibility of WSNs, there are numbers of fields of applications which now use WSNs. Wide varieties of sensors are used for such applications. The sensors used can be thermal, visual, infrared, radar, seismic, low sampling rate magnetic and these sensors are capable of monitoring variety of diversified environmental conditions like temperature, pressure, soil makeup, vehicular movement, noise levels. Sensors are also used to detect the presence or absence of certain objects, to measure the mechanical stress levels on attached objects and to measure some of the dynamics characteristics like speed of the target object along with its direction and size. As the numbers of applications of WSNs are enormous, one cannot list an exhaustive list of the application areas but can be briefly categorize in military, health, home, environment and commercial applications.

1.4.1 Military applications

Most extensive use of WSNs is done in military applications right from the inception of WSN. The category of military applications can be in computing, intelligence, surveillance, military command, control etc. The characteristics of WSNs like rapid deployment, cheap sensors, self-configuration and fault tolerance makes WSNs the most suitable option in military operations as in the battlefield there is a need to quick deployment of the nodes so as to setup a network as fast as possible in high tension areas. The number of nodes may be subject to change frequently due to intrusions in the field and hence self-configuration is necessary to keep the network working. Cheap cost of sensors in WSN is another factor which is of importance to military applications as the sensors once deployed may be destroyed and can be subjected to redeployment quite frequently.

Few military operations involving WSN is targeting, battlefield surveillance, battle damage assessment, monitoring friendly forces, equipment and ammunition etc. Sensor networks are used in the guidance systems of intelligent ammunition. Battle damage assessment is done with the help of WSNs in the areas which are not in the home range. For the battlefield surveillance military install WSNs on the critical terrains and the areas far from the home range and can closely track the activities of the opposing forces. Other than these military can always use WSNs for monitoring of the troops of the friendly forces to keep a close watch on them. WSNs are also used in managing the information related to the availability and conditions of the equipments used in the battlefield before and after the battles.

1.4.2 Health applications

Health industry has adopted the use of WSNs for variety of applications right from the simple diagnostics to providing interfaces for the disabled. Some other applications which are employing WSNs are patient monitoring, administration of drugs, monitoring the movements of internal processes for animals of interest, telemonitoring of the physiological human data and also tracking of the doctors in the hospital.

In telemonitoring of the physiological human data sensors give more freedom to the humans. It can retain the information from the sensed behavior of the individual which can be stored for long and can be used for exploration. Also, use of WSNs in telemonitoring provides greater flexibility and comfort to the patient. WSNs used to track the doctors and patient is done by attaching sensor nodes to them. The sensor nodes attached has specific tasks which may be checking heart beat, blood pressure etc.

1.4.3 Environmental applications

WSNs play a major role in environmental applications specially related to natural calamities and disasters. Some of the other applications using WSNs are monitoring environmental conditions which can affect the growth of crops, tracking the movements of small animals, insects and birds, monitoring in soil, marine and other atmospheric contexts, bio-complexity and pollution studies.

Precision in agriculture can be maintained by WSNs as it can monitor the level of soil erosion, the level of pesticides in water and the amount of air pollution. Flood detection is another area where WSNs are used and one of the examples of such sensors is ALERT which is used by US. Several sensors like weather, water level and rainfall sensors are deployed and send's the data packets to central authority. The central authority then monitors the values and takes appropriate actions and keeps the data for later purposes. Another critical area of environmental applications where WSNs are used is forest fire detection. There are situations when forest fire can broke large and can cause a lot of disaster. Thus to avoid such incidents sensor nodes are deployed either statically or dynamically of the stretch of the forest with high density. In case of early stages of fire in forest the sensor node communicate the exact location of the origin of fire and thus necessary action can be taken to stop it from spreading. If the fire has expanded and is beyond controls the information from the sensors helps in evacuating the place and thus saves any human losses.

1.4.4 Commercial applications

Many commercial applications use WSNs for better flexibility and versatility. Some of the commercial applications which use WSNs are monitoring quality of the products, construction of smart office places, robot control, automation and control in factories, machine transportation. WSNs are deployed in security services also like monitoring and detecting thefts, tracking of vehicles. Various factory applications are modeled using WSNs like factory instrumentation, local control of actuators, instrumentation of semiconductor processing chambers, rotating machinery, wind tunnels, and anechoic chambers.

Environmental control in the office buildings is widely handled using WSNs. As the heat and air conditioning of most of the buildings is centrally controlled there are places with differences in temperature. One place in the room may be cooler than the other as the central control does not distribute the energy evenly. Thus distributed WSNs are deployed in such building to control the temperature and air flow in different parts of the building. Detection and monitoring of vehicle thefts is another commercial application which uses WSNs. Sensors are spread within a geographical region and with the help of internet these changes reach the users in the remote areas. Also the location in case of the theft can be directly sent to the BS and theft vehicle can be recovered. In inventory control each item is attached to a sensor node which can tell the location of the item. Thus in the inventory the user can locate each and every item and can keep a count of items. Remote user can also track the exact position of the items and can be ensured of the count of the items.

1.5 WSN Properties

Most of the applications of WSNs share many common characteristics. Some of the similarities based on the interaction(s) between sources and BS are:

- Periodic monitoring: sensor can be made to periodically measure the values in the environment and send the measured values to the user.
- Event detection: Specific events can be detected by the sensor nodes. Such sensor are also applied to addresses the event of natural calamities. Some of the applications with event detection are grass fires, forest fires, volcanic eruptions etc. Classification of events is necessary in case of a sensor network used for more than one event detection.
- Location driven: Most of the applications work on the changes in the position of the sensors. Applications theft detection, inventory management, detection of behavior of humans is monitored based upon the changes in the position of the sensors.
- Tracking: Some applications sense the data even after continuous change in position. In cases of surveillance if any threat is on the move the sensor nodes communicate the source to the BS along with the speed and direction. This helps to estimate the present position of the targets.

Sensor Network Protocol Stack

The sensor network protocol stack is different from the standard TCP/IP and along with the layers used in traditional protocols it has additional planes to handle the issues in the sensor nodes.

The application layer is involved in usage and development of application software(s) based on the specified sensing tasks. The flow of data in sensor networks is managed by transport layer of the protocol stack. The transport layer provides the data to the network layer and it is the responsibility of the network layer to route the data as desired. Data link layer handles the noise and mobility of the nodes in the network and are power aware. Data link layer is also responsible for reducing the collision with the neighbors' broadcast. Transmission, modulation and receiving techniques are handled by physical layer of the protocol stack. The additional planes of power, mobility and task management are specially created to cater the needs of sensor node related to battery, movement and task distribution. Main objective of these planes is to carry out the collaborative sensing task as desired with least power consumption. The power management plane (PMP) has the responsibility to decide how a node will use its power like when it has to switch on their sensors and when they have to switch off them. Power management plane also takes care of when to transmit the information and when not to. Mobility management plane (MMP) is responsible to know the route back to the BS in case of movement of the sensor nodes from their initial positions and inform neighbors about the newly moved sensor node. With the updated knowledge of neighbors now they can contribute and use the power of the nodes efficiently. Task management plane (TMP) performs the balancing of sensing tasks amongst the sensor nodes. It is not necessary that all the sensor nodes in the vicinity should sense and transmit the information. Thus TMP takes such decisions as per the application used in WSNs. So the additional planes are very essential for the objective of WSNs as without them a node will have no capability and will act as an individual transmitting the unwanted information and wasting the constrained power of the sensor node. These planes are essential for simple computations at the sensor nodes as per the changes in power, mobility and required tasks efficiently by the sensor nodes.

Research Objective

Sensor networks can be employed to benefit a variety of applications in diverse environments. To benefit the applications, sensor nodes differ greatly in technical requirements for these various applications.

With the motivation explained in the previous section, the objective of my research work can be identified as:

- To find a better way to save energy during data transmission.
- To reduce the overhead on one node of transmitting the data to BS.
- Increasing the energy saving efficiency mechanism through serial communication.
- To overcome the constraints of cluster based architecture.
- More Cooperation between the neighboring nodes.
- To increase the reliability during data transmission.

I am proposing an enhancement over LEACH protocol. The protocol, called ADAPTIVE ENERGY-EFFICIENT PROTOCOL (AEFP). The protocol is a near optimal chain-based protocol for extending the lifetime of network.

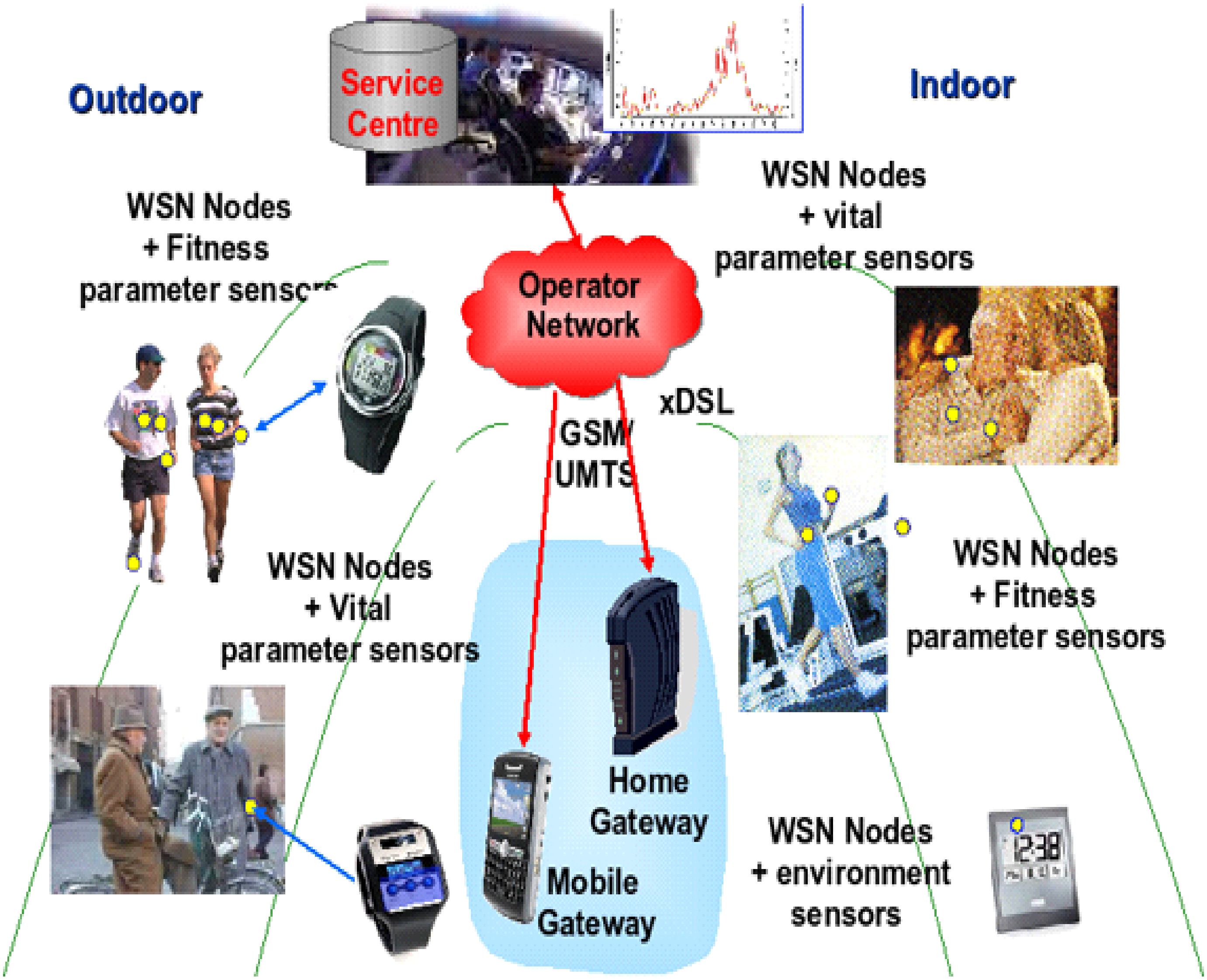


Fig 1.3 Research Objectives

Comparison with ad hoc

Wireless sensor networks mainly use broadcast communication while ad hoc networks use point-to-point communication.

Unlike ad hoc networks wireless sensor networks are limited by sensors limited power, energy and computational capability. Sensor nodes may not have global ID because of the large amount of overhead and large number of sensors.

A wireless ad hoc network (WANET**)** is a decentralized type of wireless network. The network is ad hoc because it does not rely on a pre existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity. In addition to the classic routing, ad hoc networks can use flooding for forwarding data.

An ad hoc network typically refers to any set of networks where all devices have equal status on a network and are free to associate with any other ad hoc network device in link range. Ad hoc network often refers to a mode of operation of IEEE 802.11 wireless networks.

It also refers to a network device's ability to maintain link status information for any number of devices in a 1-link (aka "hop") range, and thus, this is most often a Layer 2 activity. Because this is only a Layer 2 activity, ad hoc networks alone may not support a routable IP network environment without additional Layer 2 or Layer 3 capabilities.

The earliest wireless ad hoc networks are the "packet radio" (PRNETs) from the 1970s, sponsored by DARPA after the ALOHA net project.

Chapter 2. Related Work

In a hierarchical architecture, sensors organize themselves into clusters and each cluster has a cluster head, i.e. sensor nodes form clusters where the low energy nodes are used to perform the sensing in the proximity of the phenomenon. The less energy constrained nodes play the role of cluster-heads and process, aggregate and forward the information to a potential layer of clusters among themselves toward the base station.

2.1 LEACH

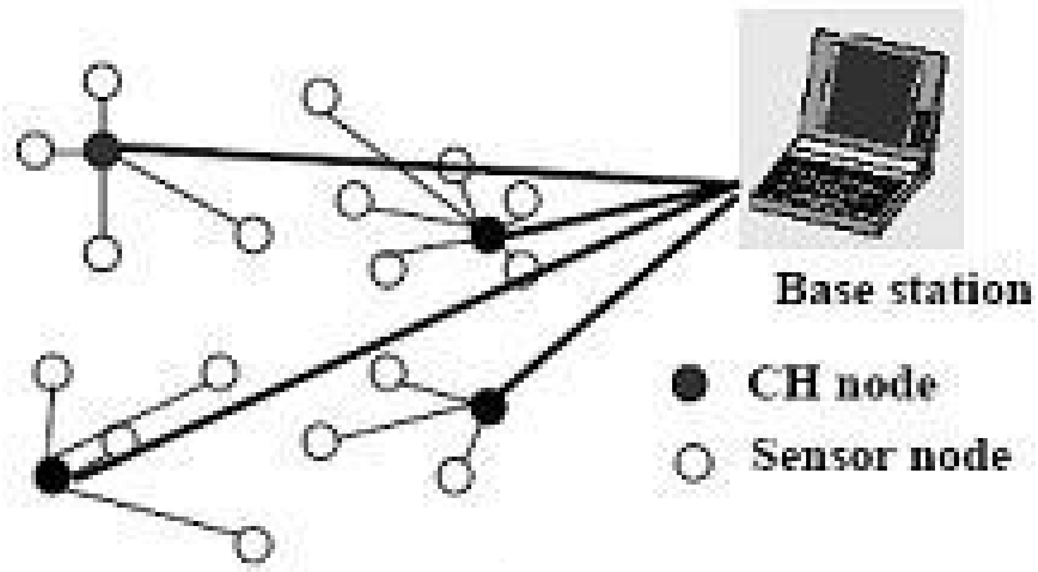


Fig.2 LEACH

Fig. 2 shows the communications in LEACH protocol.

Heinzelman introduced a hierarchical clustering algorithm for sensor networks, called Low Energy Adaptive Cluster Hierarchy – based protocol (LEACH).
In LEACH the operation is divided into rounds.

2.2 Protocol explanation

LEACH is a hierarchical protocol in which most nodes transmit to cluster heads, and the cluster heads aggregate and compress the data and forward it to the base station (sink). Each node uses a stochastic algorithm at each round to determine whether it will become a cluster head in this round. LEACH assumes that each node has a radio powerful enough to directly reach the base station or the nearest cluster head, but that using this radio at full power all the time would waste energy.

Nodes that have been cluster heads cannot become cluster heads again for P rounds, where P is the desired percentage of cluster heads. Thereafter, each node has a 1/P probability of becoming a cluster head in each round. At the end of each round, each node that is not a cluster head selects the closest cluster head and joins that cluster. The cluster head then creates a schedule for each node in its cluster to transmit its data.

All nodes that are not cluster heads only communicate with the cluster head in a TDMA fashion, according to the schedule created by the cluster head. They do so using the minimum energy needed to reach the cluster head, and only need to keep their radios on during their time slot.

LEACH also uses CDMA so that each cluster uses a different set of CDMA codes, to minimize interference between clusters.

Fig. 2.1 illustrates an example of fixed zoning and the resulting virtual grid architecture (VGA)

VGA used to perform two level data aggregation. Note that the location of the base station is not necessarily at the extreme corner of the grid; rather it can be located at any arbitrary place.

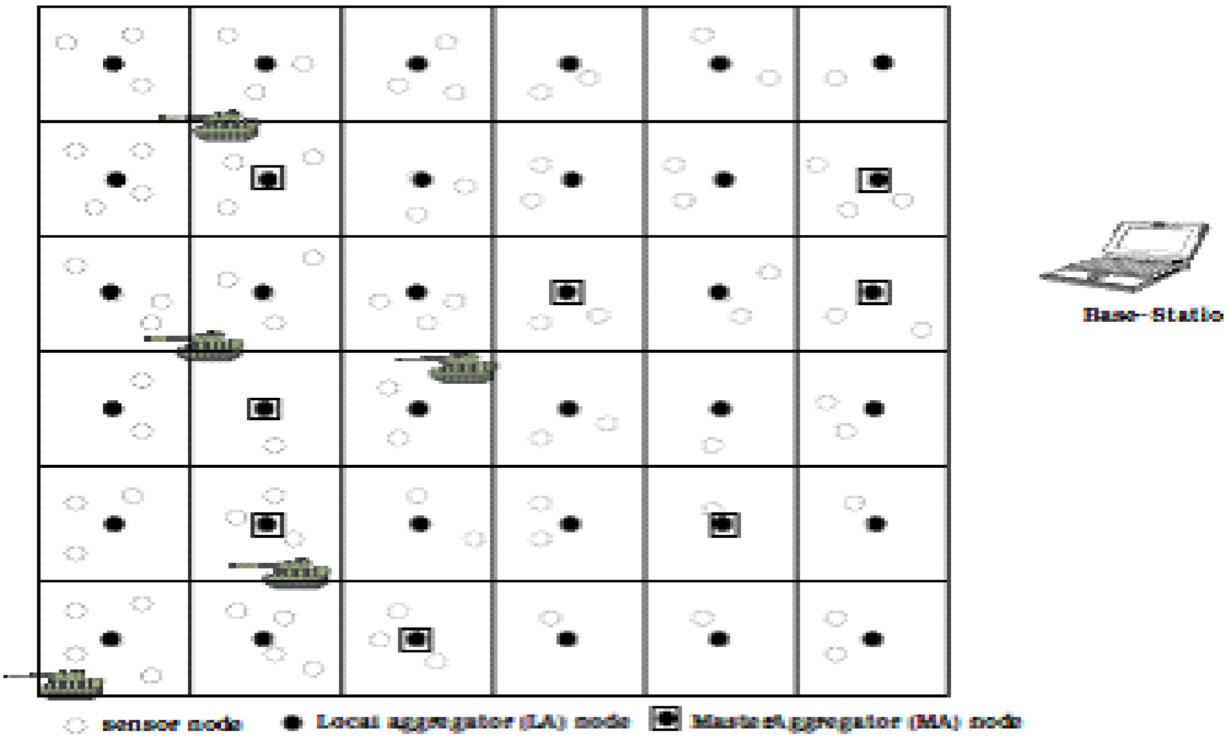


Fig.2.1 Regular shape tessellation applied to the network area

2.3 Properties

Properties of this algorithm include:

Cluster based
Random cluster head selection each round with rotation. Or cluster head selection based on sensor having highest energy
Cluster membership adaptive
Data aggregation at cluster head
Cluster head communicate directly with sink or user
Communication done with cluster head via TDMA

2.4 Functionality

- In each round a different set of nodes are cluster-heads (CH).
- Nodes that have been cluster heads cannot become cluster heads again for P rounds.(each node has a 1/p probability of becoming a cluster head in each round. At the end of each round)
- Each node that is not a cluster head selects the closest cluster head and joins that cluster to transmit data.
- The cluster heads aggregate and compresses the data and forward it to the base station, thus it extends the lifetime of major nodes.
- In this algorithm, the energy consumption will distribute almost uniformly among all nodes and the non-head nodes are turning off as much as possible.

2.5 Limitations

- LEACH assumes that all nodes are in wireless transmission range of the base station which is not the case in many sensor deployments.
- In each round, LEACH has cluster heads comprising 5% of total nodes.
- It uses Time Division Multiple Access (TDMA) as a scheduling mechanism which makes it prone to long delays when applied to large sensor networks.

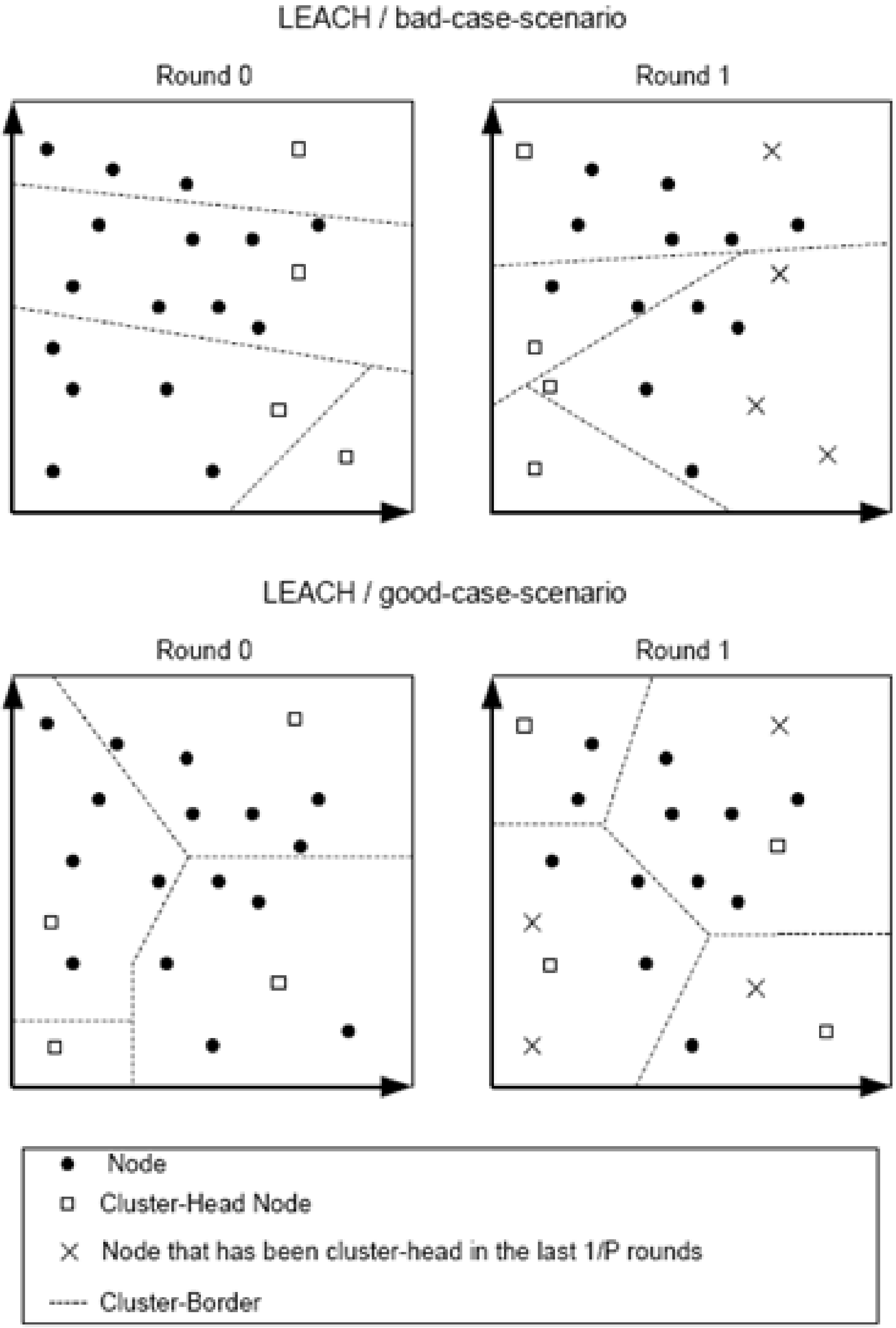


Fig 2.2 LEACH Scenarios

3. AEFP Functionality

In AEFP, each node communicates only with the closest neighbor by adjusting its power signal to be only heard by this closest neighbor.

Each node is aware of its closest neighbor which helps him to propagate the data.

After chain Formation AEFP elects a leader from the chain in terms of residual energy every round to be the one who collects data from the neighbors to be transmitted to the base station. As a result, the average energy spent by each node per round is reduced.

Step 1: Initializing WSN parameter
Step 2: Deploying No. of Nodes
Step 3: Implementing AEFP Protocol (Chain formation based on nearest neighbor)
Step 4: Initialize parameter for routing
Step 5: Defining Energy rate
Step 6: Routing update with each round
Step 7: Checking Lifetime and Data Consumption

The main idea in AEFP is for each node to receive from and transmit to close neighbors and take turns being the leader for transmission to the BS. This approach will distribute the energy load evenly among the sensor nodes in the network. I initially place the nodes randomly in the play field, and therefore, the i-th node is at a random location. I used random 100-node networks for our simulations with similar parameters used in. I propose to place the BS at a far distance from all other nodes. For a 50m x 50m plot, our BS is located at (25, 150) so that the BS is at least 100m from the closest sensor node.

For constructing the chain, I assume that all nodes have global knowledge of the network .Chain is formed based on the neighboring node. Each node fuses its data and sends the packet to next nearest node.

To construct the chain, I will propose to start with the furthest node from the BS. I propose to begin with this node in order to make sure that nodes farther from the BS have close neighbors.

Fig3 shows node X connecting to node X+1, node X+1 connecting to node X+2, and node X+2 connecting to node X+3 in that order(In a sequential order). When a node dies, the chain will be reconstructed in the same manner to bypass the dead node.

3.1 Efficiency

Unlike LEACH, AEFP avoids cluster formation and uses only one node in a chain to transmit to the Base station instead of multiple nodes. This approach reduces the overhead and lowers the bandwidth requirements from the BS. Fig. 3 shows that only one cluster head leader node forward the data to the BS.

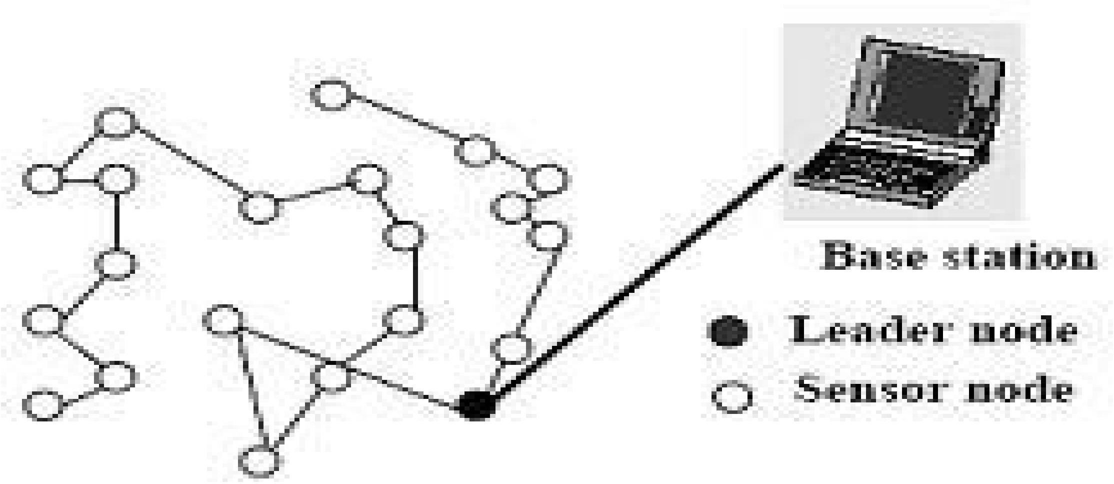


Fig.3 (AEFP)

A cluster-free approach is used to transmit the data. Data aggregation is performed at two levels: local and then global. The data is sent via neighboring nodes. The node sends data via its neighbor node. The overhead of Cluster head sending data to base station is ignored. This saves energy of each node and also makes data transfer more reliable.

Routing Protocol	**Classification**	**Power Usage**	**Data Aggregation**	**Scalability**	**Query Based**	**Overhead**
LEACH	Hierarchical / Node-centric	High	Yes	Good	No	High
AEFP	Hierarchical	Low	Yes	Good	No	Low

Fig.3.1 AEFP/LEACH Comparison Table

3.2 Network Specification

To evaluate the performance of the hierarchal routing protocols, the simulation consists of 100 homogeneous nodes with initial energy of 0.5 Joule, scattered randomly within a 40x40 m sensor field. The Base Station is (BS) located at (25,150) m, so it is at least 110 m far from the closest packets and Data packet length 2000. The energy consumption due to communication will be calculated using the radio energy model. I assume that each sensor node generates one data packet per time unit to be transmitted to the BS. For simplicity, I refer to each time unit as a round. During the simulation process, only a set of sources will be selected randomly at each round to send their data to the BS. Transmitting energy & Receiving Energy is 50*0.000000001

3.3 Proposed Algorithm

```
The sensor node SN(i,j) is assigned as the initiator in the jth round;
J=1;
SN(i,j) generates 2 packets and sends them to SN(i,1) and SN(i,n) respectively;
let x=1 and y=n;
repeat
if ( x<j && distance(SN(i,j),SN(i,x)) <  distance (SN(i,j),SN(i,y)) ) // data transmission in the left side of S(i,j)
{
    SN(i,x) fuses received data(i,x-1) (x>1) and its own data(i,x);
    SN(i,x) transmits the fused data and the token to its neighboring
    SN(i,x+1);
     x=x+1;
}
Else if (y>j  &&  distance(SN(i,j),SN(i,x)) >  distance(SN(i,j),SN(i,y))  ) // parallel data transmission in the right side of S(i,j) if left node distance is greater than right node
{
    SN(i,y) fuses received data(i,y+1)(y<n) and its own data(i,y);
    SN(i,y) transmits the fused data and the token to its neighboring
    SN(i,y-1);
    y=y-1;
 }
```

```
until (x=j) and (y=j)

J=j+1;
```

3.4 Energy Model:-

In this work, the energy model adopted is as follows

Radio mode SIMULATION METRICS FOR WIRELESS SENSOR NETWORKS (WSN)

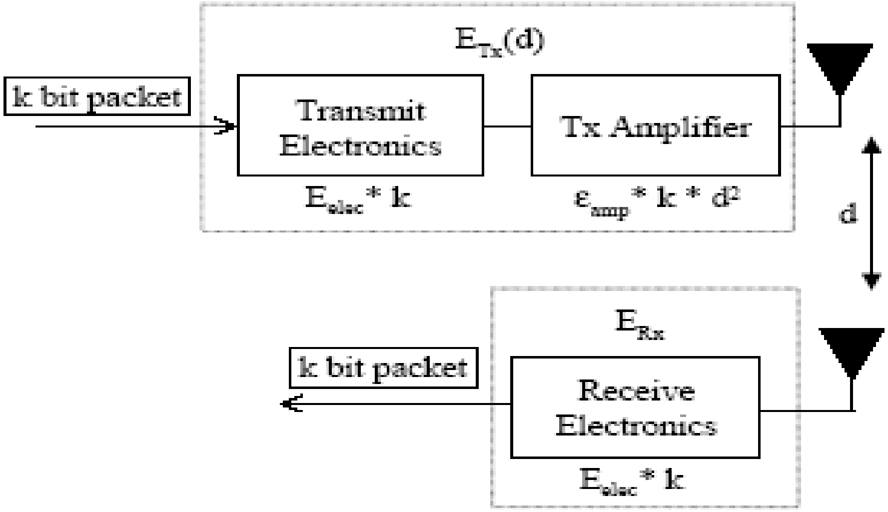


Fig 3.2 Energy Model

```
%Energy Model (all values in Joules)


Eo = 0.5;              %Initial Energy


ETX=50*0.000000001;     %Eelec=Etx=Erx


ERX=50*0.000000001;


%Transmit Amplifier types
```

```
Efs=100*0.000000000001;

Emp=0.0013*0.000000000001;

%Data Aggregation Energy

EDA=2*0.000000001;

data_pkt_length=k;

Eelc= ERX+ETX; (Eelc is the energy dissipated per bit to run the  transmitter or the receiver circuit)

Node.send_dis(i)=d;

Energy_Cost_per_round=Energy_Cost_per_round+(ERX+ETX)*data_pkt_length+Efs*data_pkt_length*(d.^2);

Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Emp*(d.^4)*data_pkt_length);
```

1. Physical parameters (Referred from LEACH)

Physical parameters are the same of all three protocols as follows

- Transmission speed 100 bit/s

- Network bandwidth 5000 bit/s

- Data packet processing delay 0.1 ms

2. Protocol parameters (Referred from LEACH)

LEACH parameters:

Cluster type Dynamic

CHs percentage (%) 5.0

CHs selection cycle 1.00 s

AEFP parameters:

No special parameters for AEFP

Simulation parameters (Referred from LEACH)

Simulation criteria (Referred from LEACH)

Two scenarios are used to measure the performance, as follows:

A) When the value of transmission range is 15 m

B) When the value of transmission range is 70 m

These two scenarios are used according to the Loss of network connectivity. Sensorial will report the network life time at the round when a sensor node is isolated (all its neighbors run out of energy), i.e. the network is not fully connected.

3.5 Result:-

It is considered to be an extension of LEACH protocol.

In this protocol the concept of chaining comes into picture. The sensor nodes which are closest to each other will be considered to form the chain and this chain is responsible for communicating with the base station. Only one node will be considered from this chain to transmit to the base station instead of multiple nodes.

Each node will fuse its own information with the information of the Neighboring node and this will form a single packet. This single packet will be of the same length and transmit the fused information to the next sensor node.

A new chain is constructed using the same process when a sensor node in the chain dies due to limited battery power. The problem with the chaining approach of AEFP protocol is that whenever a single node dies the whole chain has to be constructed as it becomes non functional.

It is possible that some nodes may have relatively distant neighbor nodes along the chain in AEFP. On one hand, nodes already on the chain cannot be revisited. On the other hand, when a node dies, the chain is reconstructed in the same manner to bypass the dead node.

A Chain Construction Phase The algorithm uses the following steps to form a chain: a) Initialize the network parameters. Determine the number of nodes, initial energy, BS location information et al. Then chain construction starts. b) Transmit data to next nearest neighbor node. c) Neighbor node fuses its data if it wants to send and then send to next neighbor node. d) End node of the chain obtains the information of distance between itself and other nodes which have not joined the chain yet, finds the nearest node: i represent the i-th node joined. The chain-building methods in exiting protocol and proposed protocol are respectively used to the same network of 100 nodes randomly arranged.

Parameter	**Value**
No of Nodes	100
Data Packet Length	2000
`Efs`	`100*0.000000000001`
Transmission Energy	50*0.000000001
Receiving Energy	50*0.000000001
`Emp`	`0.0013*0.000000000001`
Topology	Random

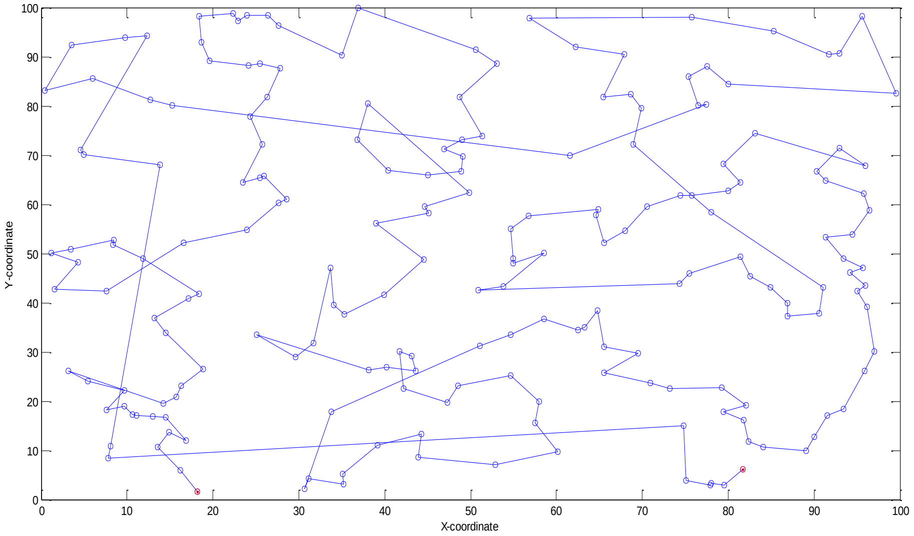


Fig.4 AEFP Result Topology

Energy cost per round= 0.0486

Parameter	**Value**
No of Nodes	400
Data Packet Length	2000
`Efs`	`100*0.000000000001`
Transmission Energy	50*0.000000001
Receiving Energy	50*0.000000001
`Emp`	`0.0013*0.000000000001`
Topology	Random

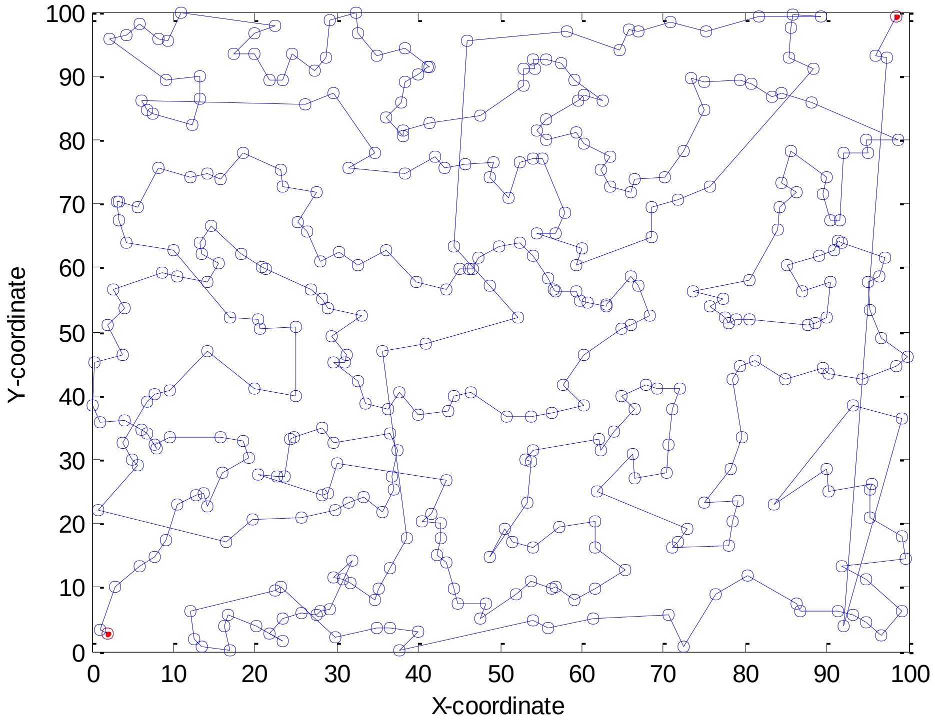


Fig.4.1 AEFP Result Topology

Energy cost per round= 0.09

Parameter	**Value**
No of Nodes	400
Data Packet Length	200
`Efs`	`100*0.000000000001`
Transmission Energy	50*0.000000001
Receiving Energy	50*0.000000001
`Emp`	`0.0013*0.000000000001`
Topology	Random

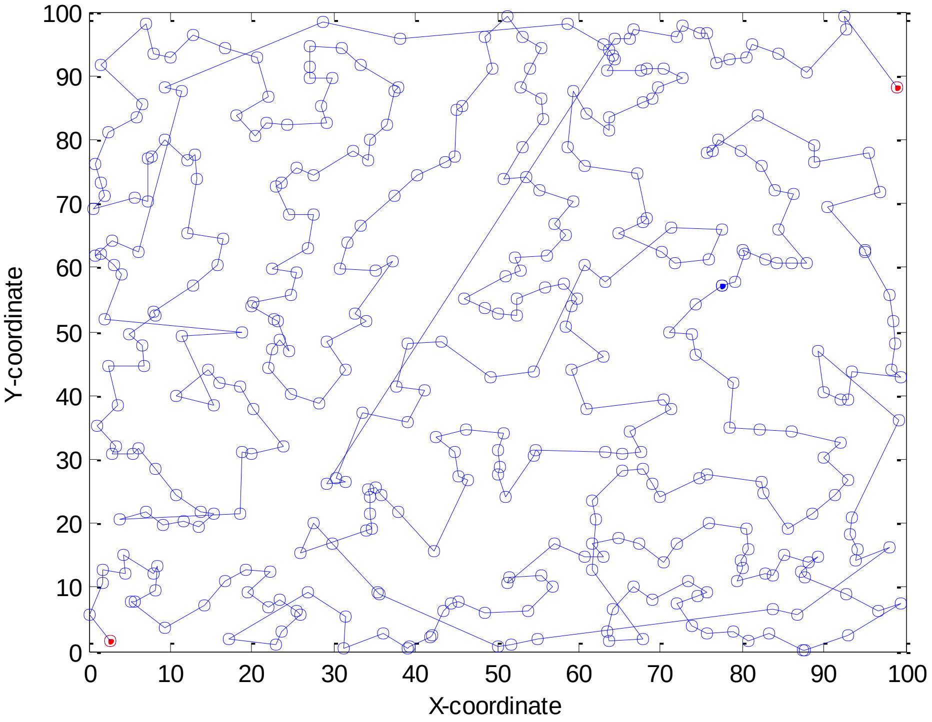


Fig.4.2 AEFP Result Topology

Energy cost per round= 0.0133

Parameter	**Value**
No of Nodes	300
Data Packet Length	200
Efs	100*0.000000000001
Transmission Energy	50*0.000000001
Receiving Energy	50*0.000000001
Emp	0.0013*0.000000000001
Topology	Random

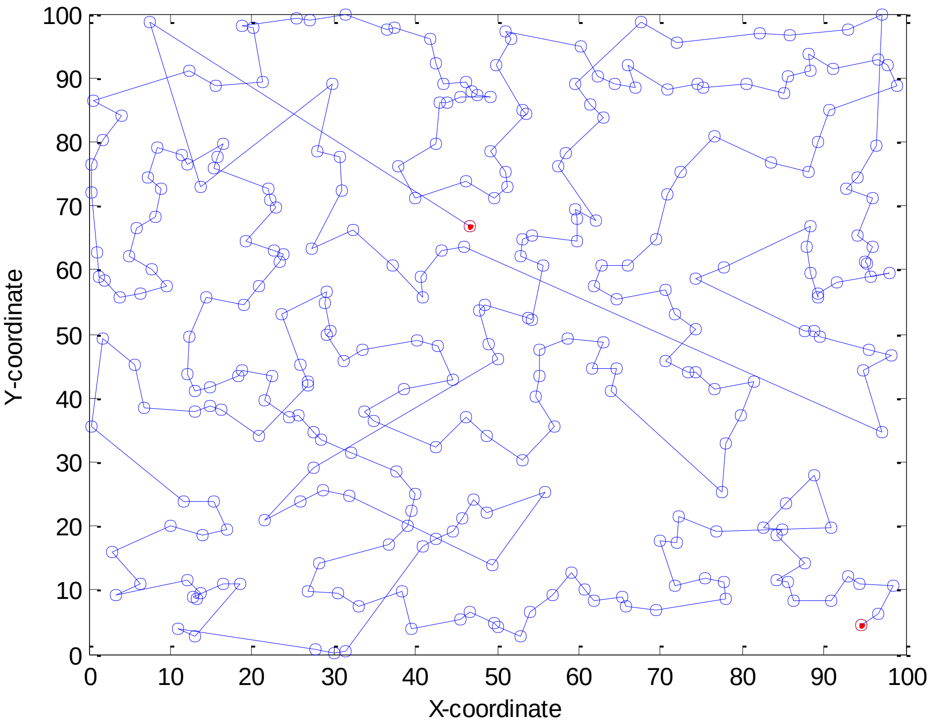


Fig.4.3 AEFP Result Topology

Energy cost per round= 0.0102

Parameter	Value
No of Nodes	200
Data Packet Length	200
Efs	100*0.000000000001
Transmission Energy	50*0.000000001
Receiving Energy	50*0.000000001
Emp	0.0013*0.000000000001
Topology	Random

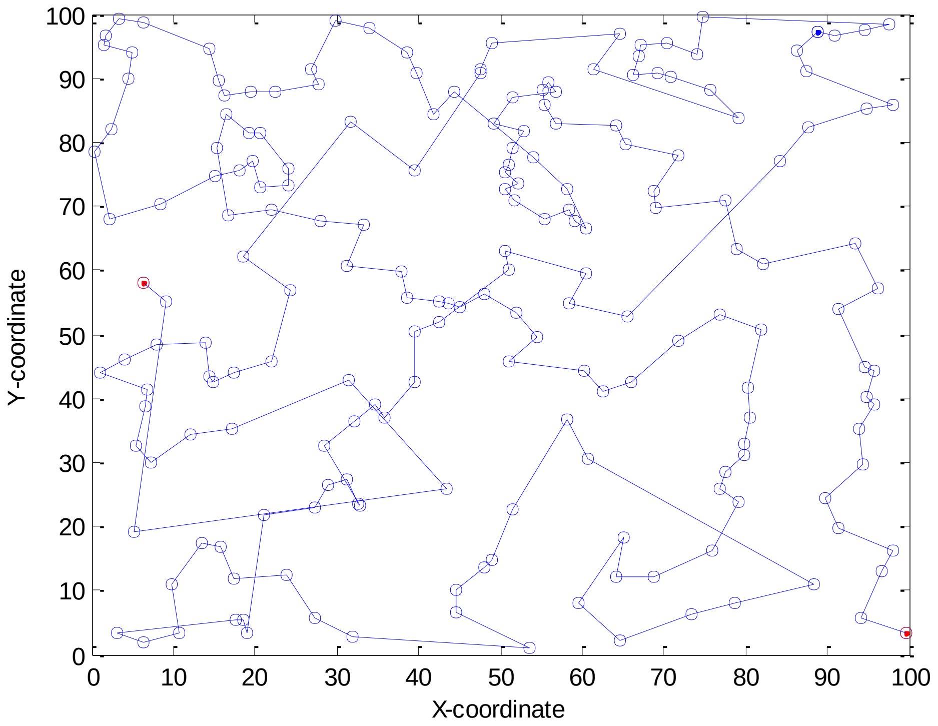
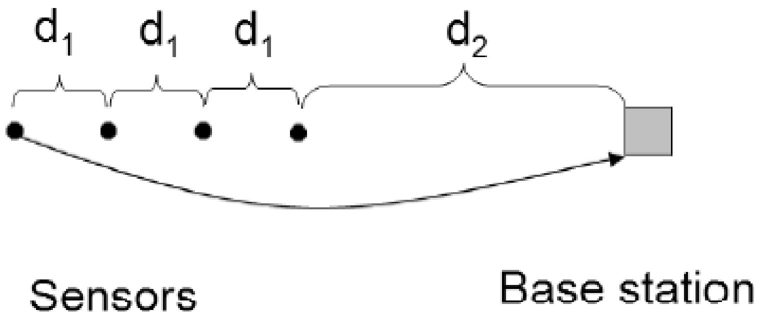
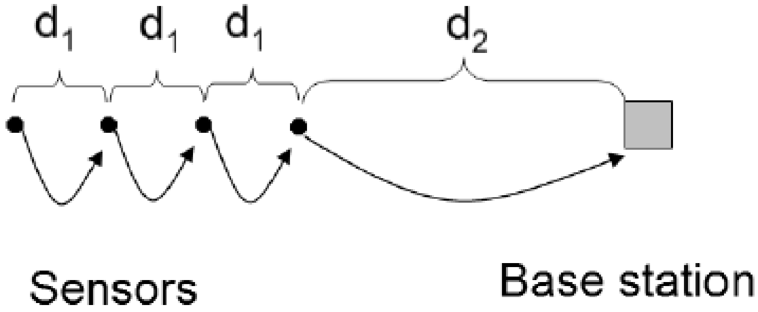


Fig.4.4 AEFP Result Topology

Energy cost per round= 0.0069



(a) Direct transmission



(b) Minimum transmission energy

Fig 5 Energy Transmission

Conclusion:-

In this report, I describe AEFP; it is chain based protocol that is near optimal for a data-gathering problem in sensor networks. AEFP outperforms LEACH by eliminating the overhead of dynamic cluster formation, minimizing the distance non leader-nodes must transmit, limiting the number of transmissions and receives among all nodes, and using only one transmission to the BS per round.

The proposed algorithm will improve the existing LEACH protocol. The proposed work is implemented on Wireless Sensor network to improve the network life in case of chain based protocol. The main problem with cluster network is to find the next neighbor for communication. Here the improvement is done for existing LEACH protocol. In this work I have include one parameter, which is distance from neighbor to select the next neighbor. The work is about to identify an energy efficient aggregative path to communicate over the network.

Future Work

Distribution of the energy load among the nodes in the network increases the life-time of the network. The energy load distribution among the nodes also decides the quality of the network.

In this case AEFP is better than LEACH. When the size increases AEFP shows better performance that LEACH. There is still scope of improving the node distribution in the network.

In Mobile Sensor network, the size of the cell should be modified according the node distribution. In the network nodes take turns to transmit the encrypted data to the base station to balance the energy usage in the Network. It increases the robustness of the network sensor nodes.

There is a scope in improving the energy distribution with adjacent nodes in terms of node deployment.

Reference

- An Energy Efficient Algorithm Based on LEACH Protocol [Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference]- {Adopted: LEACH Protocol}

- Energy Efficient Routing in Wireless Sensor Networks Through Balanced Clustering. [Stefanos A. Nikolidakis 1,*, Dionisis Kandris 2, Dimitrios D. Vergados 1 and Christos Douligeris 1] –
{Adopted: Energy Distribution Algorithm}

- Comparative Study of Radio Models for data Gathering in Wireless Sensor Network International Journal of Computer Applications (0975–8887) Volume 27–No.4, August 2011 {Adopted: Radio Model for Energy Algorithm}

- J. Agre and L. Clare, "An integrated architecture for cooperative sensing networks," Computer, vol. 33, no. 5, pp. 106-108, 2000.
- S. Zhang and H. Zhang, "A review of wireless sensor networks and its applications;' in Automation and Logistics (ICAL), 2012 IEEE International Conference on. IEEE, 2012, pp. 386-389.
- S. Athmani, D. E. Boubiche, and A. Bilami, "Hierarchical energy efficient intrusion detection system for black hole attacks in wsns," in Computer and Information Technology (WCCIT), 2013 World Congress on. IEEE, 2013, pp. 1-5.
- M. C. M. Thein and T. Thein, "An energy efficient cluster-head selection for wireless sensor networks," in Intelligent systems, modelling and simulation (ISMS), 2010 international conference on. IEEE, 2010, pp. 287-291.
- F. Zhang and Q. Wang, "A probability optimize clustering routing algorithm for wsns," in Electronics Computer Technology (ICECT). 2011 3rd International Conference on, vol. 4. IEEE, 2011, pp. 348-352.
- B. Wang, C. Shen, and J. Li, "Study and improvement on LEACH protocol in wsns," 2012.

Appendix:-

```
//Variable Declaration


clear;
clc;
clf;

NodeNums=400;

AreaR=200;

Bx=50;
By=175;
Tr=100;
%for NodeNums=100:20:400

num=1;
num_plot=1;
En=0.25;
send_to_sink=0;
ctl_pkt_leng=100;
data_pkt_length=2000;
die_node_num=0;
die_node_num_pri=0;
```

```
run_per_round=0;

Transmit_Pack=0;

Energy_Cost=0;

Pre_Energy_Cost=0;

Engery_Inter_Cost=0;

Send_Begin=0;

Per_Round_Energy_Cost=0;


alive=1;

dead=0;



//Node Selection

Node.x=AreaR*rand(1,NodeNums); % the position of node

Node.y=AreaR*rand(1,NodeNums);

%Node.x(100)=AreaR/2;

%Node.y(100)=AreaR/2;

Node.pri=linspace(0,0,NodeNums);

Node.already=linspace(0,0,NodeNums);

Node.to_nbr_dis=zeros(NodeNums);

Node.to_pri_dis=linspace(0,0,NodeNums);

Node.send_dis=linspace(0,0,NodeNums);

Node.E=linspace(En,En,NodeNums);

Node.status=linspace(alive,alive,NodeNums);

Node.E_dis=linspace(0,0,NodeNums);
```

```
%Eelec=Etx=Erx
ETX=50*0.000000001;
ERX=50*0.000000001;
%Transmit Amplifier types
Efs=100*0.000000000001;
Emp=0.0013*0.000000000001;
%Data Aggregation Energy
EDA=5*0.000000001;
do=sqrt(Efs/Emp);

for i=1:1:NodeNums
To_sink_dist(i)=sqrt((Node.x(i)-Bx)^2+(Node.y(i)-By)^2);
end

%[Max_Dis, Max_num]=max(to_sink_dist);
%now_node=Max_num;
%Node.already(now_node)=1;
%Max_num
%now_node
%node_head=Max_num;
%send_to_sink=now_node;
%node_to_send=now_node;
%start_node=now_node;
```

```
%for nodes=1:1:100

%Node.pri=linspace(0,0,NodeNums);
%Node.already=linspace(0,0,NodeNums);
%Node.to_nbr_dis=zeros(NodeNums);
%Node.to_pri_dis=linspace(0,0,NodeNums);
%Node.send_dis=linspace(0,0,NodeNums);
%Node.E=linspace(En,En,NodeNums);
%Node.status=linspace(alive,alive,NodeNums);
%Node.E_dis=linspace(0,0,NodeNums);
%run_per_round=0;
%send_to_sink=0;
%die_node_num=0;
%die_node_num_pri=0;
%Send_Begin=0;

while run_per_round<1 %die_node_num<1
```

```
%                 num_plot=1;

%                             die_node_num_pri=die_node_num;

Pre_Energy_Cost=Energy_Cost;

%                 run_per_round

%                 die_node_num

%                 Energy_Cost


for i=1:1:NodeNums

if Node.status(i)==alive

%                                              Node.E_dis(i)=Node.E(i)/(ETX+Efs*(To_sink_dist(i)^2));

Node.E_dis(i)=Node.E(i);

else

Node.E_dis(i)=0;

end

end


if run_per_round==0||die_node_num_pri~=die_node_num


Node.pri=linspace(0,0,NodeNums);

num_plot=1;

Node.already=linspace(0,0,NodeNums);

%                  run_per_round

for i=1:1:NodeNums

if Node.status(i)==alive

to_sink_dist(i)=sqrt((Node.x(i)-Bx)^2+(Node.y(i)-By)^2);
```

```
else
to_sink_dist(i)=0;
end
end

[Max_Dis, Max_num]=max(to_sink_dist);
now_node=Max_num;
Node.already(now_node)=1;
node_head=Max_num;
%                 send_to_sink=now_node;
node_to_send=now_node;
start_node=now_node;
%                 now_node

while num_plot~=NodeNums-die_node_num
if num_plot==NodeNums-die_node_num
Node.pri(now_node)=0;
node_tail=now_node;
end

for j=1:1:NodeNums
if Node.already(j)==0&&Node.status(j)==alive
Node.to_nbr_dis(now_node,j)=sqrt((Node.x(now_node)-Node.x(j))^2+(Node.y(now_node)-Node.y(j))^2);
else
```

```
Node.to_nbr_dis(now_node,j)=0;
end
end

j=1;
while Node.to_nbr_dis(now_node,j)==0
j=j+1;
end

min_dis=Node.to_nbr_dis(now_node,j);
min_num=j;

for j=1:1:NodeNums
if Node.to_nbr_dis(now_node,j)~=0&&Node.to_nbr_dis(now_node,j)<min_dis
min_dis=Node.to_nbr_dis(now_node,j);
min_num=j;
end
end
Node.to_pri_dis(now_node)=min_dis;
Node.pri(now_node)=min_num;

Node.E(now_node)=Node.E(now_node)-(ETX*ctl_pkt_leng+Efs*2*(min_dis*
min_dis)*ctl_pkt_leng);
Node.E(Node.pri(now_node))=Node.E(Node.pri(now_node))-ETX*ctl_pkt_leng;
```

```
if now_node==Max_num

Node.to_pri_dis(now_node);

end

now_node=Node.pri(now_node);



%                                   min_num

Node.already(now_node)=1;

%                                   Node.to_pri_dis(now_node)=min_dis;

%                                   min_dis

num_plot=num_plot+1;

end



node_tail=now_node;







I_want=1;

if I_want==1

Fig.(1);

if run_per_round~=1

clf;

end

%             plot(Bx,By,'+');

%             hold on;

for i=1:1:NodeNums

if Node.status(i)==alive
```

```
plot(Node.x(i), Node.y(i), 'o','markersize',5);

hold on;

end

end


plot(Node.x(node_tail), Node.y(node_tail), 'o-r','markersize',5);

plot(Node.x(node_head), Node.y(node_head), 'o-r','markersize',5);


for i=1:1:NodeNums

if Node.status(i)==alive

if Node.pri(i)~=0

plot([Node.x(i);Node.x(Node.pri(i))],[Node.y(i);Node.y(Node.pri(i))]);

hold on;

end

else

plot(Node.x(i), Node.y(i), 'o-r');

end

end

end

xlabel('X-coordinate');

ylabel('Y-coordinate');


%
                plot([Node.x(send_to_sink);Bx],[Node.y(send_to_sink);By]);

%                                           hold on;
```

```
%                     node_to_send=Max_num;
%                     start_node=Max_num;

end

%                   send_to_sink=node_head;
%                   next_next=mod(run_per_round,NodeNums-die_node_num);
%                   while next_next~=0
%                                              send_to_sink=Node.pri(send_to_sink);
%                                              next_next=next_next-1;
%                   end
%                     [max_node_En,send_to_sink]=max(Node.E_dis);

send_to_sink=mod(Send_Begin,NodeNums)+1;
run_per_round=run_per_round+1;
send=send_to_sink;
while Node.status(send_to_sink)~=alive
send_to_sink=mod(send_to_sink,NodeNums)+1;
end
Send_Begin=send_to_sink+1;
Transmit_Pack=Transmit_Pack+NodeNums-die_node_num;

die_node_num_pri=die_node_num;
%                   for i=1:1:NodeNums
%                                  if Node.E(i)<0.01*En
```

```
%                                               die_node_num_pri=die_node_num_pri+1;
%                            end
%                end

num=1;
right_send_num=1;
node_to_send=node_head;
start_node=node_head;

while node_to_send~=send_to_sink
right_send_num=right_send_num+1;
node_to_send=Node.pri(node_to_send);
end
%                  right_send_num
Node.send_dis=linspace(0,0,NodeNums);
while num~=NodeNums+1-die_node_num
if num<right_send_num
Node.send_dis(start_node)=Node.to_pri_dis(start_node);
start_node=Node.pri(start_node);
end
if num==right_send_num
Node.send_dis(start_node)=to_sink_dist(start_node);
start_node_1=start_node;
start_node=Node.pri(start_node);
end
```

```
if num>right_send_num
Node.send_dis(start_node)=Node.to_pri_dis(start_node_1);
start_node_1=start_node;
start_node=Node.pri(start_node);
end
num=num+1;
end

Engery_Inter_Cost=0;
for i=1:1:NodeNums
if i~=send_to_sink && Node.status(i)==alive
Engery_Inter_Cost=Engery_Inter_Cost+ETX+ERX+Efs*Node.send_dis(i).^2;
end
end
%                    Engery_Inter_Cost/(NodeNums-die_node_num)

if send_to_sink==node_head
for i=1:1:NodeNums
if Node.status(i)==alive
if i==send_to_sink
if Node.send_dis(i)>do
%                                                  Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);
Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
```

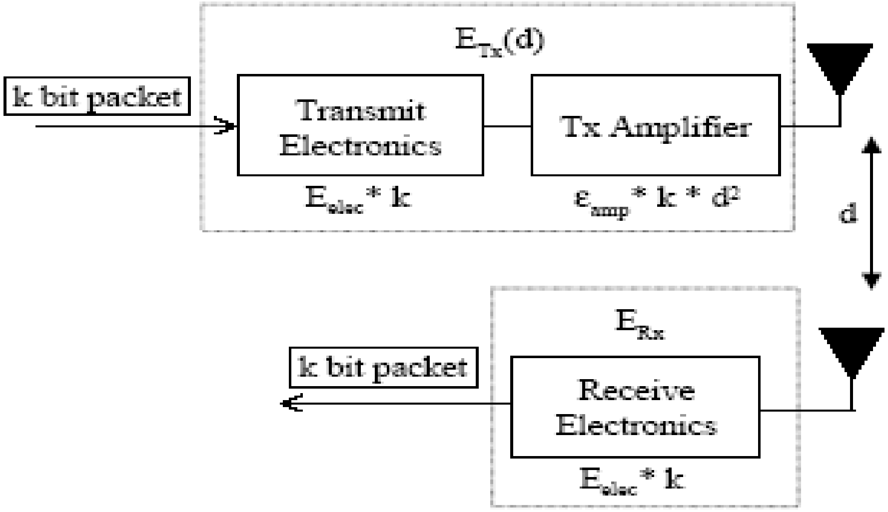
```
else
Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
end
end
if i==node_tail
if Node.send_dis(i)>do
% Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);
Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
else
Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
end
end
if i~=send_to_sink&&i~=node_head&&i~=node_tail
if Node.send_dis(i)>do
% Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);
Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
else
Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
end
```

```
end
end
end
end

if send_to_sink==node_tail
for i=1:1:NodeNums
if Node.status(i)==alive
if i==send_to_sink
if Node.send_dis(i)>do
% Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);
Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
else
Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
end
end
if i==node_head
if Node.send_dis(i)>do
% Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);
Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);
else
```

```
Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

end

end

if i~=send_to_sink&&i~=node_head&&i~=node_tail

if Node.send_dis(i)>do

% Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);

Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

else

Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

end

end

end

end

end




if send_to_sink~=node_tail&&send_to_sink~=node_head

for i=1:1:NodeNums

if Node.status(i)==alive

if i==send_to_sink
```

```
if Node.send_dis(i)>do

% Node.E(i)=Node.E(i)-(2*ERX*data_pkt_length+EDA*3*data_pkt_length+ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);

Node.E(i)=Node.E(i)-(2*ERX*data_pkt_length+EDA*3*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

else

Node.E(i)=Node.E(i)-(2*ERX*data_pkt_length+EDA*3*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

end

end

if i==node_head||i==node_tail

if Node.send_dis(i)>do

% Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);

Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

else

Node.E(i)=Node.E(i)-(ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

end

end

if i~=send_to_sink&&i~=node_head&&i~=node_tail

if Node.send_dis(i)>do

% Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Emp*(Node.send_dis(i).^4)*data_pkt_length);
```

```
Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

else

Node.E(i)=Node.E(i)-(ERX*data_pkt_length+EDA*2*data_pkt_length+ETX*data_pkt_length+Efs*(Node.send_dis(i).^2)*data_pkt_length);

end

end

end

end

end


%                       for k=1:1:NodeNums

%                                               if Node.pri(k)~=0

%
```



```
%%%%%%%%%%%%%ENERGY CALCULATION%%%%%%%%%%%%%
Per_Round_Energy_Cost=Per_Round_Energy_Cost+(ERX+ETX)*data_pkt_length+Efs*data_pkt_length*(Node.send_dis(k).^2);
%                                    else
%
Per_Round_Energy_Cost=Per_Round_Energy_Cost+(ERX+ETX)*data_pkt_length+Efs*data_pkt_length*(Node.send_dis(k).^2);
%                                    end
%                  end
Per_Round_Energy_Cost=0.25*100-sum(Node.E);
Energy_Cost=En*NodeNums-sum(Node.E);
round_cost=Energy_Cost-Pre_Energy_Cost;
%                 round_cost
%                 send_to_sink
%                 Node.E
%                 if Node.pri(send_to_sink)==0
%                           send_to_sink=node_head;
%                 else
%                           send_to_sink=Node.pri(send_to_sink);
%                 end

die_node_num=0;
for i=1:1:NodeNums
if Node.E(i)<0.001*En
die_node_num=die_node_num+1;
```

```
Node.status(i)=dead;
end
end
end
run_per_round
%plot(NodeNums,run_per_round,'o-r');
%hold on;
%end
%plot(run_per_round,NodeNums-nodes,'o-r');
%hold on;
%end
```