An Energy Efficient & Hybrid Adaptive Intra Cluster Routing For Wireless Sensor Networks

Dissertation By:

Bashir Ahmed
(507-FBAS/MSCS/F08)

Supervised By:

Prof. Dr. Muhammad Sher

Co-Supervised By:

Mr. Zeeshan Shafi

Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University Islamabad

2011

International Islamic University, Islamabad

Dated: 22 August 2011

Final Approval

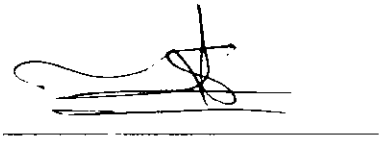
This is to certify that we have read the thesis submitted by **Bashir Ahmed, Reg # 507-FBAS/MSCS/F08**. It is our judgment that this project is of standard to warrant its acceptance by the International Islamic University, Islamabad, for the Degree of **MS in Computer Science**.

Project Evaluation Committee

External Examiner:
Dr. Hassan Mahmood
Assistant Professor
Department of Electronics
Quaid-e-Azam University
Islamabad

Internal Examiner:
Dr. Muhammad Zubair
Assistant Professor
Department of Computer Science
International Islamic University
Islamabad

Supervisor:
Prof. Dr. Muhammad Sher
Chairman
Department of Computer Science
International Islamic University
Islamabad



Co-Supervisor:
Mr. Zeeshan Shafi
Senior Researcher
Center of Excellence in Information Assurance (CoEIA)
King Saud University
Saudi Arabia

DEDICATION

I dedicate this research project to my beloved

PARENTS

A Dissertation submitted to

Department of Computer Science,

Faculty of Basic and Applied Sciences,

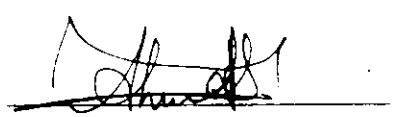
International Islamic University, Islamabad

As a partial Fulfillment of Requirements for the Award of the Degree of

MS in Computer Science

DECLARATION

I hereby declare that this thesis **"An Energy Efficient & Hybrid Adaptive Intra Cluster Routing for Wireless Sensor Networks"** neither as a whole nor as a part has been copied out from any source. It is further declared that I have done this research with the accompanied report entirely on the basis of my personal efforts, under the proficient guidance of my teachers and my friends especially my supervisor Prof. Dr. Muhammad Sher and co-supervisor Mr. Zeeshan Shafi. If any of the system is proved to be copied out of any source or found to be reproduction of any project from any of the training or educational institutions, I shall stand by the consequences.



Bashir Ahmed

Reg # 507-FBAS/MSCS/F08

ACKNOWLEDGEMENT

All praises to Almighty ALLAH PAK, the most Gracious and Beneficent, Whose copious blessings enable me to pursue and perceive higher ideals of life. Darood and salaam for His beloved Prophet MUHAMMAD (Sallalaho Alaehe wa Aahlehi wa Sallam) Who demonstrate the righteous path to whole mankind and drag it out from the nastiest depths of ignorance to the preeminent level of humanity.

I am proud to express my deep sense of obligation and special appreciation to my reverend supervisor honorable Prof. Dr. Muhammad Sher and co-supervisor Mr. Zeeshan Shafi for their dexterous guidance and kind behavior during the project. Their encouragement, moral support and motivation helped me a lot to get through any problem or difficulty during each step.

I am much grateful to all my respected teachers for their guidance and help throughout my life which results in letting me to reach at this stage. I am also thankful to whole administration and management team of International Islamic University especially of Computer Science Department for their managerial and administrative support at every step.

I would like to say thanks to Mr. Raja Adeel Akhtar and Bahria University Islamabad for helping and supporting me in implementation of my research idea.

Without thanking my all truly friends acknowledgement cannot be complete, who were always there to help me in every thick and thin and in completion of my degree. Especially Mr. Tanveer-ul-Haq, Mr. Abd-ur-Rehman, Mr. Muhammad Khalil, Mr. Umer Saleem, Mr. Zia-ur-Rehman, Mr. Tariq Niazi, Mr. Malik Khurram Awan, and Mr. Khurram Shafique.

Furthermore I must highlight that it was mainly due to my family's moral and financial support during my entire academic career which enabled me to complete my work dedicatedly. I owe all my achievement to my most loving and caring parents who mean most to me, are most precious than any treasure on earth. Their endless

efforts, care and prayers are always around me to keep me safe in every difficulty and help me in attaining every success.

I also would like to thank my brothers, sisters and life partner who encouraged me at those moments when I got exhausted and also thanks to my all supportive cousins for their always encouraging and guiding attitude.

Finally, I am whole heartedly thankful to all the fellows who have helped me during achieving this degree.

Bashir Ahmed

Reg # 507-FBAS/MSCS/F08

Project in Brief

Project Title:	An Energy Efficient & Hybrid Adaptive Intra Cluster Routing For Wireless Sensor Networks
Organization:	International Islamic University, Islamabad (IIUI)
Undertaken By:	Bashir Ahmed Awan (507-FBAS/MSCS/F08)
Supervised By:	Prof. Dr. Muhammad Sher Chairman, Department of Computer Science International Islamic University, Islamabad
Co- Supervised By:	Mr. Zeeshan Shafi
Start Date:	February 2010
Completion Date:	February 2011
Tools & Technologies:	TinyOS TOSSIM TinyViz MS Office
Operating System:	Windows XP Linux (cygwin)
System Used:	Intel(R) Pentium(R) Dual Processor 1.6 GHz RAM 1 GB 120 GB Hard Disk

ABSTRACT

Wireless Sensor Network (WSN) is one of the most emerging technologies that will change the world's future. Research work is in progress to enhance the productivity and effectiveness of this technology. As with the growth of this field, the challenges to sensor nodes are increasing with same pace. So for researchers there are many dimensions of research. WSN is the network of tiny independent sensor nodes, called motes, generally they are battery operated and due to their critical nature it not possible to change the batteries randomly, which is one of the main concerns of wireless sensor networks and is directly concerned to the network lifetime. For enhancement in the network lifetime any well-organized routing technique can play a vital role, as about 80% of energy is consumed while transmission of data from senor node to base station. To overcome this energy issue, this research "An Energy Efficient & Hybrid Adaptive Intra Cluster Routing" which is a blend of previously used techniques is proposed that provides an optimal solution. Quantitative analysis exposes its energy competence over the other work published.

TABLE OF CONTENTS

LIST OF FIGURES

Figures Page No.

LIST OF TABLES

Tables Page No.

LIST OF ABBREVIATIONS

Abbreviations	Explanation
ADC:	Analog to Digital Converter
AICR:	Adaptive Intra Cluster Routing
AM:	Active Messages
ASIC:	Application Specific Integrated Circuit
BS:	Base Station
CH:	Cluster Head
CIDRSN:	Cluster ID Based Routing in Sensor Networks
CLI:	Command Line Interface
CPU:	Central Processing Unit
CRC:	Cyclic Redundancy Check
DBG:	Debug
DPM:	Dynamic Power Management
DSP:	Digital Signal Processor
DVS:	Dynamic Voltage Scaling
EECR:	Energy Efficient Clustering Routing
EEPROM:	Electronically Erasable Programmable ROM
EYES:	Energy Efficient Networks
FPGA:	Field Programmable Gate Array
GHz:	Giga Hertz
GUI:	Graphical User Interface
IDE:	Integrated Development Environment
IEEE:	Institute of Electrical and Electronics Engineers
LAN:	Local Area Network
LASER:	Light Amplification by Stimulated Emission of Radiation
LEACH:	Low-Energy Adaptive Clustering Hierarchy
LED:	Light Emitting Diode
MAN:	Metropolitan Area Network

Abbreviations	Explanation
NesC:	Network Embedded System C
OS:	Operating System
QoS:	Quality of Service
RAM:	Random Access Memory
RFM:	Radio Frequency Module
ROM:	Read Only Memory
RSSI:	Received Signal Strength Indicator
SOS:	Sensor Network Operating System
TOSSIM:	TinyOS Simulator
WAN:	Wide Area Network
WLAN:	Wireless Local Area Network
WSN:	Wireless Sensor Network

1

INTRODUCTION

1. INTRODUCTION

The field of wireless communications is facing tremendous growth. Wireless technology is trying to cover every part of Earth. Millions of people from different communities and sectors communicate daily using multiple devices, so are multiple applications running on these devices. By the boom of this technology one is able to communicate anywhere on the earth.

Enhancements occurred in wireless networks is remarkable especially in mobile networks and Wireless LAN (WLAN). Beside these, Wireless Sensor Networks (WSNs) are the most rapidly growing technology and getting involved in almost all sectors of life. The small and low-cost computation and communication devices called sensors; are waiting for more research to maximize the capacity in terms of energy, data storage and processing.

1.1 Wireless Sensor Network

*"**Wireless Sensor Network** is a wireless network of very small autonomous wireless devices scattered in specific geographical area with one or more sensors attached to them to monitor changes in physical or environmental conditions."*

Field of Wireless Sensor Network is new-fangled and is briskly expanding, so is the inspiring research area which has attracted significant research attention in the near past. There is variety of research dimensions owing to the potential to transfigure many sectors of our daily life and economy. This research effort added a significant piece of knowledge in the vast field of WSN.

With the growth in WSN research, the challenges to sensor nodes are increasing with equal speed. Besides networking, energy consumption, data management and security are important challenges.

1.1.1 Past and Present

The idea of wireless sensor network was first provoked by military, but after seeing its marvelous results, now it is being tried in almost every field. If we look into history, Internet was invented for military purposes and later on it was used for

Industrial applications and later on common man benefited by it. WSN has same revolution history. it was introduced by the U.S Army to use in battlefield. but nowadays the use of wireless sensor network. is becoming part of daily life as Internet is. Few of the business oriented and daily life applications of wireless sensor network are as follows:

- Battlefield awareness (e.g., multi target tracking)[1]
- Modelling long-range animal migrations
- Observing inter-species predator-prey interactions
- Analyzing the impact of human development on animal behavior
- Machine surveillance (e.g., axels of train)
- Industrial sensing and diagnostic (e.g., appliances, factory, supply chains, etc.)
- Infrastructure protection (e.g., power grids, water distribution, etc.)
- Context aware computing (e.g., intelligent home, responsive environment)
- Environmental monitoring (e.g., traffic, habitat, security, pollution, etc.)
- Intruder detection
- Fire rescue and forest fire detection
- Condition based maintenance
- Air conditioning sensors, etc.

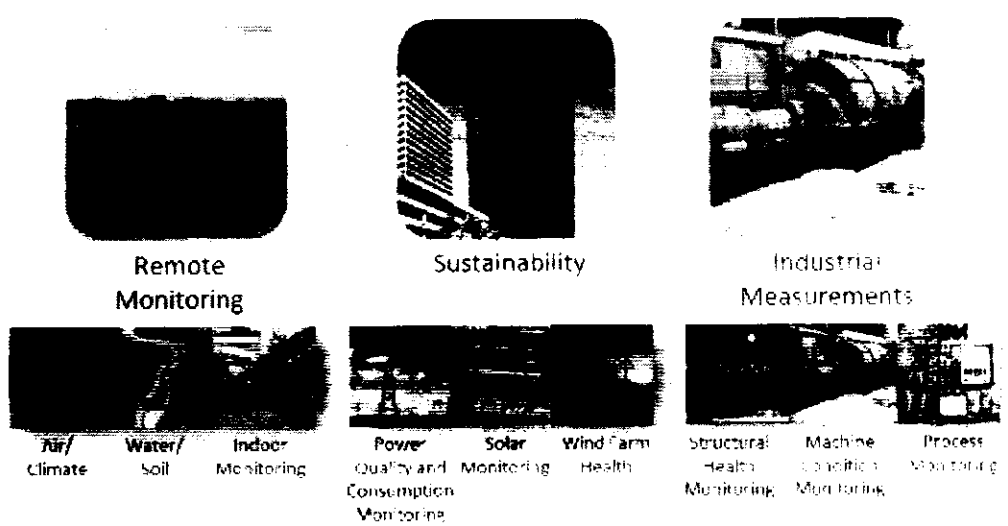


Figure 1-1: Applications of WSN [1]

1.2 Problem Identification

Wireless sensor network is a network of small tiny nodes called *motes*, the general working of motes in WSN is similar to working of a normal workstation in a LAN. These motes can perform limited processing, they can share (send and receive) sensed data with other or neighboring nodes connected to them or to other networks through cluster head. Motes are powered by batteries and battery replacement is not possible in most of the cases, therefore some significant actions are needed to be performed to reduce the usage of energy in the WSN.

Our research work is one in the chain of those significant actions to minimize the energy consumption. We have developed an energy efficient mechanism for routing in wireless sensor networks.

1.3 Contribution of our thesis in research

This new technique of routing "Hybrid Adaptive Intra Cluster Routing Mechanism" is now in the fleet of WSN, it has reduces the energy consumption which results in the enhancement in network lifetime by using the energy in quite efficient manner. This mechanism has been implemented and got mark able rise in results which is illustrated in chapter 6.

1.4 About this Thesis

This thesis is structured as per the rules of institution. A brief description of chapters is listed below.

Chapter 1 and 2: Describes Wireless Sensor Networks presenting general architecture, hardware used for WSN, software and routing models used in WSN.

Chapter 3: Literature survey. Study, summarize, and critically observed the publications of other researcher, who worked in the same domain.

Chapter 4: Complete details and core of the proposed technique.

Chapter 5: Simulation details: here we describe the details of TinyOS [2, 3], TOSSIM simulator [4, 5] and their working. Implementation details are also presented.

Chapter 6: Evaluation and comparison of proposed technique with previous techniques and protocols, results are represented in graphs.

At the end, in Chapter 7 conclusions and future work is presented.

2

WIRELESS SENSOR NETWORK

2. WIRELESS SENSOR NETWORK

Wireless Sensor Network is a new and fast growing technology. It has grabbed the attention and efforts of researchers towards itself and it is an innovative research domain in wireless communications.

As it is known that change is the only permanent thing in life, so it is the nature of humans to invent and discover more and more. Humans' mostly advances are in the area of science and technology to facilitate mankind.

2.1 Architecture of WSN

The inventions in WSN brought the combination of three main tasks, sensing computation and communication performed by a device sized about a coin, but further actions like computation, etc are not performed. As networks are combination of workstations and servers, same as Wireless Sensor Networks have two broad lines, sensing and computing. Motes are at sensing side. These are tiny, efficient with wireless capabilities, cheap and battery powered sensor units are known as sensor nodes. These are to sense the changes in the concerned area then perform basic and specific tasks on the required information and send to base station. This base station is at the computing side, which may be a laptop, workstation or heavy duty server depending upon the requirements and nature of usage.

Sensor nodes are to be installed in the specific region for specific tasks to get the required information that might be atmospheric changes or physical monitoring like spying, etc., and constantly send that information wirelessly to base station. At the base station information is received and further processed as per requirements in order to generate desired results. Figure 2-1 illustrates the WSN in a typical application.

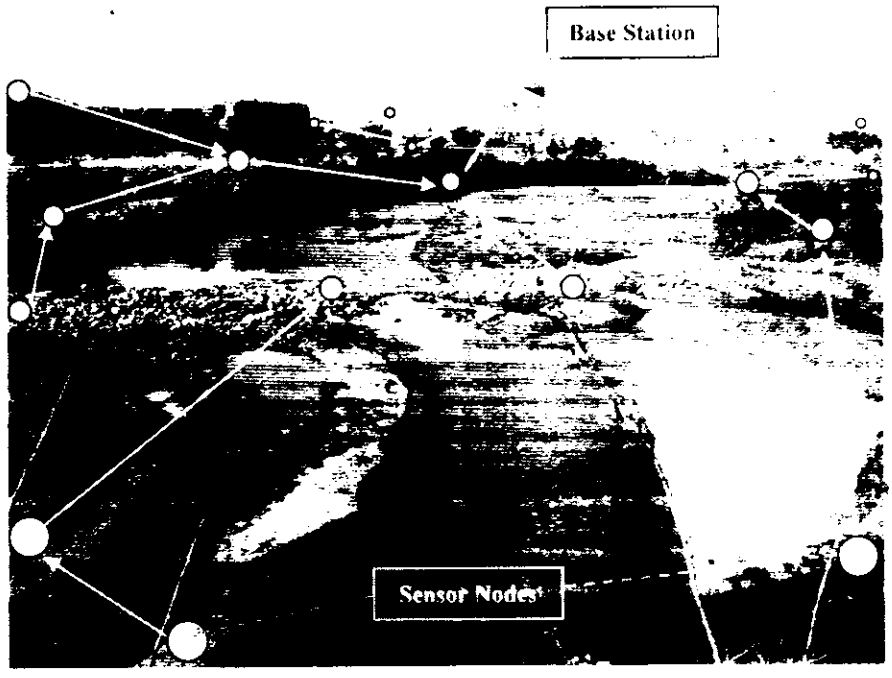


Figure 2-1: An Overview of a Wireless Sensor Network

2.2 Factors Affecting WSN

Here are few important factors which can affect the wireless sensor networks seriously, which include lifetime, coverage area, cost, ease of deployment, response time, temporal accuracy, security, and effective sample rate. The lifetime of a network is at the highest priority, therefore it is discussed in detail.

2.2.1 Lifetime

Lifetime of any network is the time from the start, of any network, till the death of its first node in that network. WSN nodes are mostly placed unattended in open area for years or even long. The major issue affecting the lifetime of a sensor network is the energy level. Each node must be designed in an organized way to properly utilize its local energy to maximize the network lifetime. It is noticed that 80% of energy is utilized during communication.

Therefore our research work focus on the same issue. We designed a routing mechanism to minimize the energy consumption during communication.

2.3 Sensing Node

There are many manufactures manufacturing the sensor nodes' hardware called mote. Few are very specific according to special environmental conditions, e.g. motes to check the affect of chemical reactions are very specific in critical chemical environments. Generally a sensor node has five main components named as: Microcontroller, Sensors, Memory, Power Supply and Radio Transceiver. All components are smart in size and energy consumption. Components and architecture of sensor node are well illustrated in **Figure 2-2**.

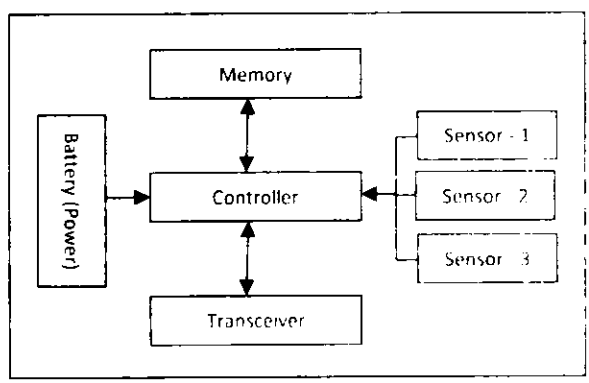


Figure 2-2: Components and Architecture of Sensor Node

2.3.1 Components of Sensor Node

Here is the detail of the components listed above.

2.3.1.1 Microcontroller

Microcontroller is the most important component of a sensor node. It processes data and controls the functionality of other components.

Easy to program, energy efficient (as consumes less power), and flexibility in establishing connection with other devices are the key features of microcontroller which let it the most suitable and appropriate for a sensor node and mostly the first priority for other embedded systems as well.

Other substitutes of microcontroller are General Purpose Microprocessors, Field Programmable Gate Array (FPGA), Digital Signal Processors (DSP) and Application Specific Integrated Circuit (ASIC). But they have much more drawbacks, so can't be used in sensor nodes.

2.3.1.2 Sensors

Sensors are used to sense the data. There are two main types of sensors, *Analog* and *digital* sensors. Analog signal is concerned with the physical changes in environment. These signals are firstly converted to digital and then sent to microcontroller for processing. Sensor nodes are typically disposable in nature as have limited energy to be consumed. Users have to be ultra careful while deploying the sensor nodes to get the optimum results.

2.3.1.3 Memory

Sensed data stays in a local sensor for a short period of time and is then transmitted, normally motes require small amounts of storage and memory, but still is one of the important components and cannot be neglected. If we need to store data for long period of time, it is more resourceful to use flash (on-chip memory) instead of SRAM (Static Random Access Memory) because flash memory does not require energy to maintain data and from storage point of view, flash has higher density than Off-Chip SRAM. Flash takes less processing time than SRAM.

2.3.1.4 Power

Power source is the most important component of a sensor node as life of a sensor node depends on it. Communication is most important task which is performed by the transceiver and process the majority of the energy available. Power can be stored in capacitors or in batteries, but primary and frequent source of power supply is battery. Few common battery technologies are Lithium, Alkaline, and Nickel Metal. Due to the current development batteries are able to be recharged by vibration or solar energy.

2.3.1.5 Radio Transceiver

In wireless communication radio transceiver has a key role. It is the component, which receives and send signal. Radio Frequency, Bluetooth, Infrared and Optical Communication are suitable options, but Radio Frequency (RF) is the most suitable option for Wireless sensor nodes because other two has issues with sensor nodes i.e. Infrared and Bluetooth has limited range for communication. Optical Communication (LASER) are sensitive and requires line of sight for communication.

Among the four modes of transceiver, "Idle" should not be utilized because of its high consumption of energy as in "Transmit" or "Receive" and there should be some intelligent rule for transceiver to move to "Sleep" mode and vice versa as plenty of energy is required in this process.

2.4 Sensor Hardware

Sensor nodes are minute sized, less energy consumers and mostly disposable. Few common brands of motes are Crossbow Berkeley Motes [7], Dust Networks [8] and Telos [9] etc.

There are multiple models of sensor nodes required for different types of tasks to be performed. Names of few common motes are mica, mica2 shown in Figure 2-3, mica2dot shown in Figure 2-4, Eyes, KMote IMote, Rene and micaz, etc.

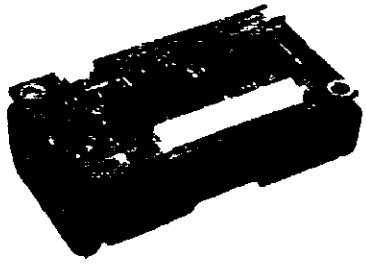


Figure 2-3: Mica2 mote. [7]

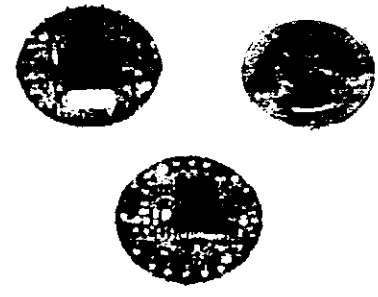


Figure 2-4: The MICA2DOT mote [7]

Complete specifications of mote models can be found on internet.

2.4.1 Future Motes

Size of electronic devices is reducing day by day and efficiency increasing simultaneously. New 5 square millimeter sized chip named as **"Spec"** has all the components found in a normal mote: CPU, memory, sensed data reader and a transmitter. To make it a complete functionally active mote only sensor(s), an antenna and battery is required. Shown in Figure 2-5.

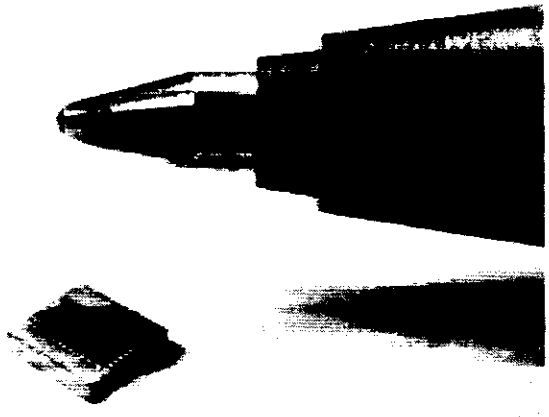


Figure 2-5: "Spec" mote compared with the tip of a ballpoint [10]

2.5 Sensor Software

Design of software architecture in wireless sensor networks has a vital role to bridge the gap between hardware and complete system. Avery sophisticated software is required to meet the requirements of complex applications. It should be competent enough that takes less energy to process the tasks, requires minimum memory and must be able to allow different applications to use system simultaneously. Software for wireless sensor networks must be efficient in:

- Real time on-sensor processing
- Time synchronization
- Managing data and events

2.5.1 Operating System

Operating system is a software which enables the hardware to run for the user, but for WSN a less complex than general-purpose operating system is required; it is because of energy constraints and applications of WSN. To facilitate wireless sensor networks there are multiple options of operating system to develop and implement algorithms. TinyOS by Hill et al [2] is designed explicitly for WSN; we use it for our research work. Unlike multithreading used by general operating systems *TinyOS uses event-driven programming model* and possess *event handler to handle the events*. TinyOS is open source software freely available on Internet for application development for wireless sensor networks.

Code of TinyOS and programs written for TinyOS are written in a special extended C-programming language named as "nesC" [11] specially designed for WSN.

There are few other operating systems for WSN.

- *"Energy Efficient Networks (EYES)"*. This operating system was proposed by Dulman and Havinga [12] for a European project. This project had characteristics of distribution, small size, power conscious and reconfiguration.
- *"Contiki"* by Dunkels et al. [13]. Contiki is an open source operating system for embedded systems, implemented in C language, portable and supports multi-tasking. Authors have also developed a simulator for it named as *Contiki network simulator*.
- *Sensor Network Operating System (SOS)* by Han et al [15] in 2005.

2.5.1.1 TinyOS

We have used TinyOS operating system for our research work because of its flexibility. It is open source and freely available on Internet. TinyOS is widely used as compared to other operating systems. Lots of improvements have been made in latest version of TinyOS.

2.6 Routing.

Routing is evolved from route which means path, way, course etc. Routing is selecting paths between one node to another in a network on which information is to be sent. Routing has a key role in all kinds of networks, but each network has its own requirement and priorities for routing. LAN, MAN, WAN are under the umbrella of electronic data networks that use packet switching, while telephone networks use circuit switching. In packet switching networks, data is sent in the form of logically addressed packets from source toward their ultimate destination either direct or through intermediate nodes. The routing process is based on the routing tables which maintain a record of the routes to a variety of destinations in the network.

Router's memory has vital role in functionality of Routers. Constructing of routing tables is performed in it, by which efficient routing is desired, normally most of the

routing algorithms utilize only one network path at a time, but by the help of multipath routing techniques multiple alternative paths can be used for routing.

2.7 Routing in WSN

Communication has a vital role in WSN. It plays a great role in deciding the routing technique to be used. Conventional routing schemes have several limitations and are not affective anymore when being used in sensor networks since the energy considerations stipulate the only essential and minimal routing to be done. Researchers have put a lot of efforts to develop a best suitable and acceptable protocol of routing in wireless sensor networks to enhance the life of network.

There are two main routing protocols for routing in WSN that are direct and indirect. Direct is also known as One-Hop/Single-Hop, which means there are no intermediate nodes between a node and base Station/cluster head. In indirect or Multi-Hop routing data is sent to its next hop and so on until it reaches to the base station. Precise description of above mentioned routing models are discussed in literature review.

3

LITERATURE SURVEY

3. LITERATURE SURVEY

Wireless Sensor Network is a key research area since a decade. a lot of research has been made and still the research is going on to make this field more and more effective. Still there are many un-discovered areas in this field to be revealed. We have studied some of latest work done published in this area. which is for knowledge. to know about the updated research work and better understanding in this specific area.

3.1 Already Published Research Work

Researchers have contributed a lot in designing multiple routing protocols for above illustrated models but still more improvement needed; critics are discussed as well at the end of each paper.

LEACH (Low-Energy Adaptive Clustering Hierarchy). a well-known routing protocol, it is developed by Heinzelman et al [16]. It's a very classic protocol. Author discussed and analyzed different routing protocols and their energy. Later on. proposed a new idea of segmentation of sensor nodes into clusters. One node among the nodes will be selected as Cluster Head. This CH role is not specific to one node and can be transferred to other nodes depending upon the energy level. Observing analytically it is proved that by this rotation of CH role energy utilization can be reduced; in result network lifetime will be increased eventually. By simulation it is calculated that LEACH can have factor of 8 in reductions in energy consumption as compared to other protocols.

The author focuses the selection of cluster head and energy is controlled by selection of cluster. This paper does not focus on the routing techniques used by the sensor nodes other than the cluster head. Moving the cluster head from one node to another again involves execution of cluster head selection algorithm, which in turn consumes energy, consequently reduces the network lifetime.

Bandyopadhyay [17] presented another technique for cluster creation. in this paper the author has presented an algorithm by which sensor nodes are organized into clusters

randomly. Energy has been saved by limiting the CH advertisement to specific number of hops.

The model proposed by the author [17] reduces the use of energy and proposed the idea of clusters but here author again mentioned that within the cluster, either Direct or MultiHop routing will be used. So the mixture of the two routing techniques is out of scope of this paper which results in reducing the network lifetime.

Researchers in [18] have discussed the two major types of routing models in WSN, Single Hop Routing Model (Direct Routing) and Multi-Hop Routing Model (Indirect Routing). Precise description of these models is as follows:

One/Single Hop Routing Model (Direct Routing)

A common and simplest most routing model uses one to one sending. Nodes are totally isolated; they are directly concerned with the base station in sending or receiving data and have no relationship with the other nodes. It is similar to STAR Topology architecture. Like in star topology all workstations are directly connected to a central switch. Figure 3-1 is well illustrating the direct routing.

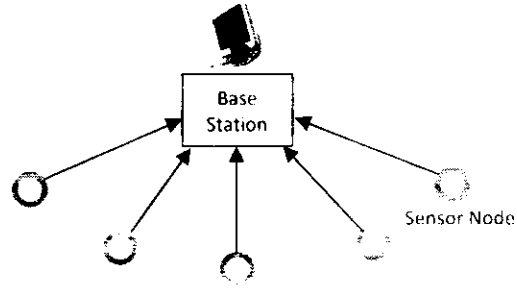


Figure 3-1: One hop/Single hop Routing Model

No dependency to other nodes. No sharing, so no security hazard, but more energy consumption. It is best suitable for small networks.

Multi-Hop Routing Model (Indirect Routing)

This is almost opposite to the Single Hop model. In this model nodes are too social. Except few which are close to base station, all nodes rely on other nodes to send data

to the base station. This model consumes less energy and is suitable for bigger networks as compared to direct routing. Scenario in quite clear in Figure: 3-2.

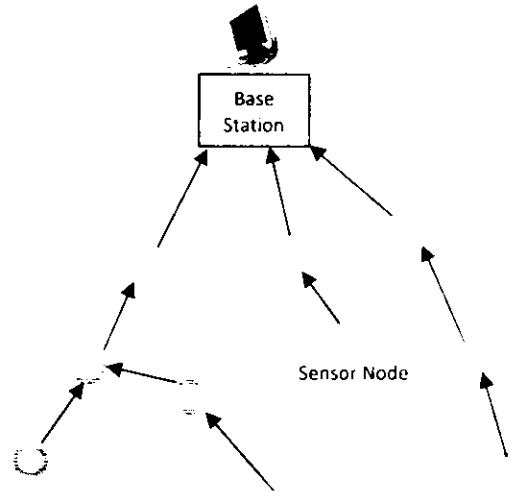


Figure: 3-2 Multi-hop model

Less energy consumption, but security hazard, data can be sniffed at the intermediate nodes. More burden on the nodes near to base station.

Researchers in [18] also presented an advanced routing model named as Cluster-Based model which later on became the core routing model in Wireless Sensor Networks. In this model whole network is segmented into clusters on the basis of distance between nodes. Few nodes form a cluster and one of them is a cluster head. Cluster head gets data from the nodes of cluster and perform few basic calculations, if required; then send the data to base station or to any other cluster head, where needed. Figure 3-3 is showing level 1 and level clusters and cluster heads.

More efficient than the direct and multihop, less energy consumption, comparatively more secure than Multi-Hop.

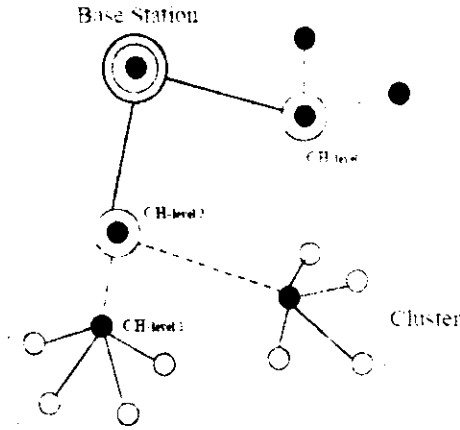


Figure 3-3: Cluster based model [18]

Ming Ma et al in [19] divide the network into two types of nodes, basic static nodes and mobile Cluster Head, on this basis he named the network as hybrid. Cluster Heads are considered as upper layer while basic static nodes are considered at lower layer. Static nodes of lower layer sense the data and send it to mobile Cluster Heads. A technique to position the Cluster Head dynamically is proposed here. Results show that by updating the Cluster Heads to better location, lifetime of network is enhanced and a balanced network can be achieved. Performance of this hybrid network is quite better than homogenous network for energy utilization. An overview of this two layered hybrid sensor networks is quite clear in Figure 3-4.

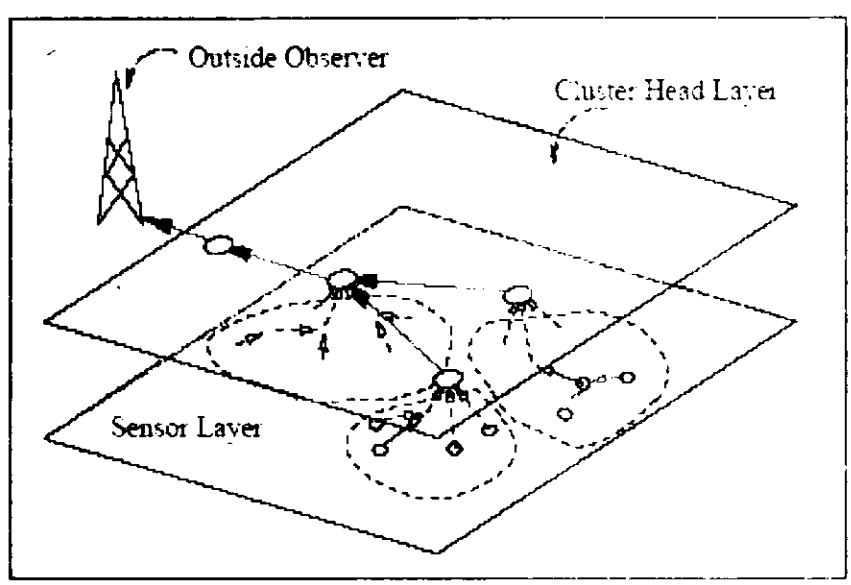


Figure 3-4: A two Layered Hybrid Sensor Network [19]

In the above mentioned paper, author achieves the energy efficiency by moving the cluster head with passage of time. This paper again uses the static routing technique. No doubt, energy efficiency can be achieved through moving the position of the cluster head, but if some efficient and adaptive routing is used along with this model it can give far better results.

Goyeneche et al proposed a distributed data gathering algorithm for wireless networks [20]. In which each node broadcast its data to its neighbors and so on until data reaches to that specific node that has already sent its data to the BS. This node will be elected as Cluster Head for the sending nodes.

+/-: ***Mechanism for selection of cluster head. Although less energy is consumed while selecting the cluster head, but this research work is not concerned with the routing techniques.***

Nauman Israr and Irfan Awan in [21] proposed a cluster based routing algorithm to achieve the less energy utilization in wireless sensor network. In this algorithm the node whose centre is in the diameter range of transmitter of neighbor nodes becomes the temporary Cluster Head of all those nodes. These temporary cluster heads are considered at top layer and all nodes are at bottom layer. Nodes send data to respective temporary cluster head at the bottom layer and at the top layer temporary cluster heads deliver data to the base station using multihop routing. By performance analyses by the author it is shown that the suggested solution is more energy aware and performs the balancing in network. Figure 3-5 shows the process of formation of temporary cluster head on the basis of coverage area and Figure 3-6 shows the general operation of algorithm.

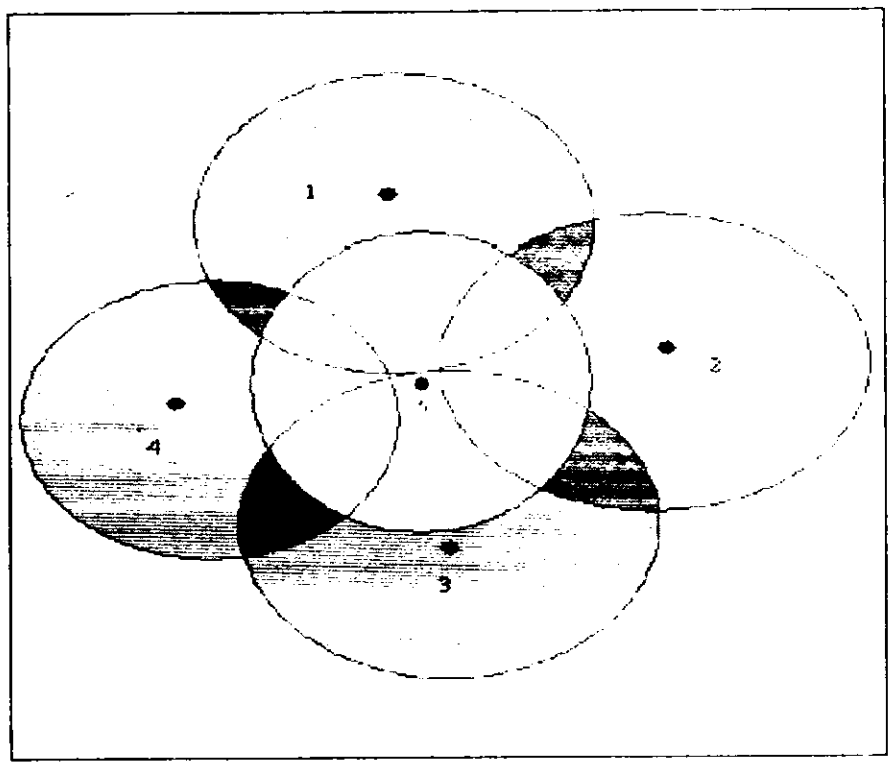


Figure 3-5: Formation of Temporary Cluster Head [21]

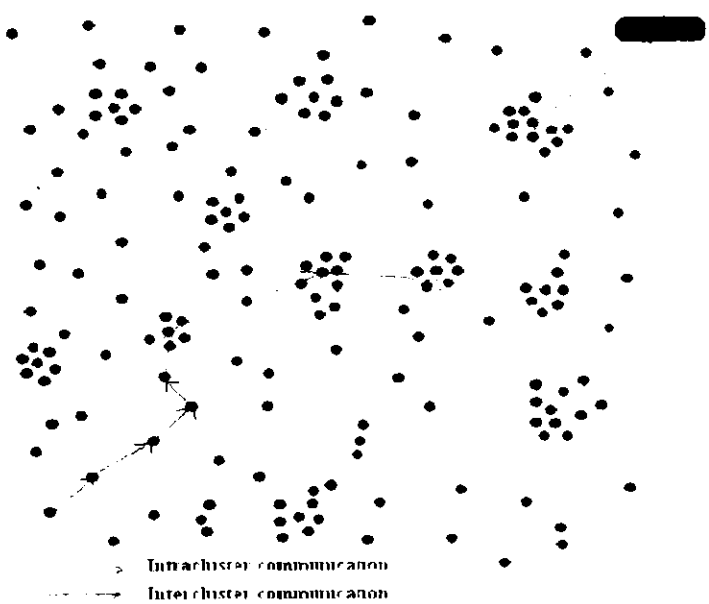


Figure 3-6: General Operation of Algorithm [21]

In this paper [21] author proposed the idea of temporary cluster head. Temporary cluster head technique can be called MultiHop routing, because the basic phenomenon is same. However selecting a temporary cluster head again requires execution of some cluster head selection algorithms, which require energy. So the energy consumed by the algorithms reduces the network lifetime.

Hoang at el. [22] presented one more energy conservation model. In this paper the broadcast nature of sensor nodes is utilized. Each time the data is transmitted in the coverage area, each node hears it even if that data is not for it. In this research work, a technique is used in which a node compresses its own data by overhearing the other communication. The energy utilization can be achieved. Collaborative broadcasting and compression used this technique for WSN is shown in **Figure 3-7**.

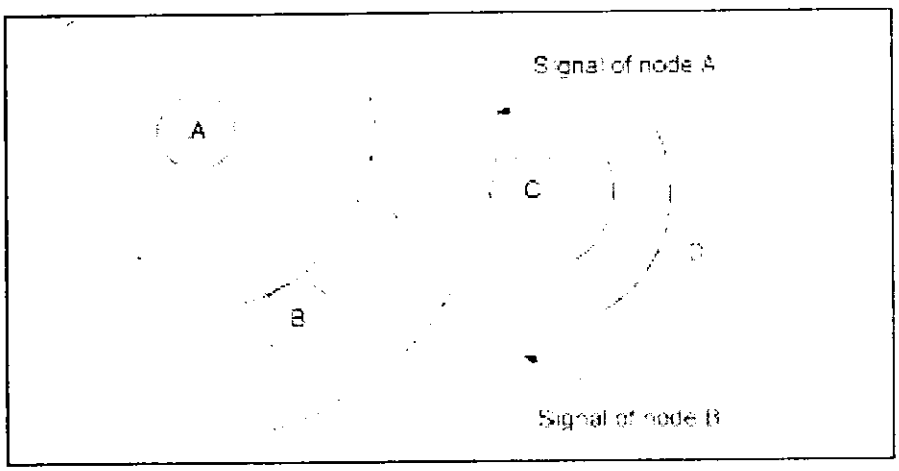


Figure 3-7: Collaborative Broadcasting and Compression in WSN

The focus of this paper is to save the energy by minimizing the receiving traffic of other nodes that is caused by the broadcast nature of the routing algorithms. But here the author again describes that either to use Direct or MultiHop routing, but not both at a time.

In [23] LiLi et al proposes an algorithm for energy efficient clustering and routing in WSN. In this algorithm cluster creation and routing in clusters is implemented. Algorithm first forms the clusters then selection of cluster heads takes place on account of weight values of nodes. A balanced network can be achieved and less amount of energy consumed during routing which results in the increase in network's lifetime.

No routing issue is discussed. Even then in routing in clusters is only by multihop or direct.

Wu et al [24] have presented a mechanism of Self-Reorganizing Slot Allocation (SRSA). In this research work inter cluster TDMA interference is reduced. Also presented a mechanism of reorganization of slot allocation on MAC layer by which inter cluster interference can be reduced as well. This mechanism is adaptive and

feedback based. Comparison of TDMA with CDMA, with randomly slot allocation and with CSMA MAC protocols took place by the simulation.

No routing issue is described in this paper. This work correlated to the MAC layer. Energy efficiency is achieved through SRSA.

In [25] Irfan Ahmed et al presented a routing protocol named as (Cluster ID based Routing in Sensor Networks). In this algorithm Cluster ID based routing is adopted and cluster size adaptation is proposed. Here Cluster ID is used as next hop instead of CH-ID in routing table. This eliminates the cluster formation process in each round of inter cluster communication. Cluster formation is only carried out in start and reformation of cluster head doesn't affect on inter-cluster communication. By this, energy consumption reduces which results in increase of network lifetime to about 16%. The results are compared with traditional hierarchical routing protocols and got the better results.

+/-: ***This work is related to inter cluster routing, energy consumption is reduced while communicating from one cluster to another. No routing issue is discussed here in this paper.***

A. Akhtar [26] presented a technique, Adaptive Intra Cluster Routing (AICR). Results show that this solution of routing minimizes the energy consumption. In which nodes are categorized on the basis of distance of nodes from cluster head. Close regions are formed by the nodes near to Cluster head. Only Direct routing is adopted in close regions and MultiHop routing outside the close region, shown in **Figure 3-8**. This utilizes the energy of network efficiently and improved the lifetime of the network.

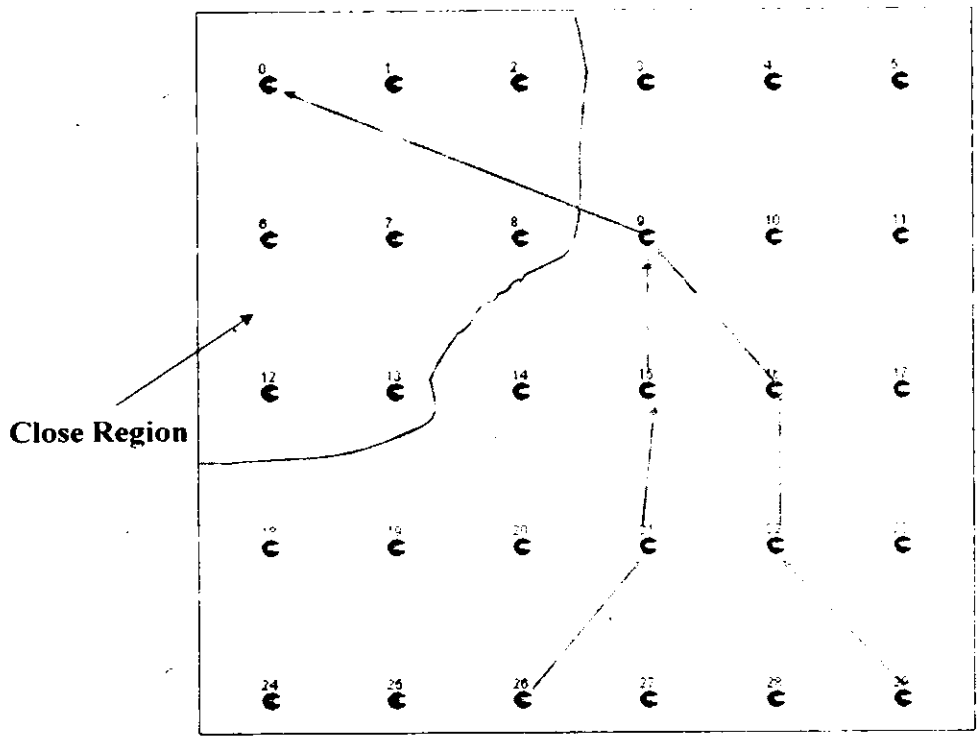


Figure 3-8: Adaptive Intra Cluster Routing technique

Energy efficiency is achieved by the blend of both the routing techniques, direct and multihop. But still they are not effectively used; full benefits of the blend are not availed, so is not fully adaptive. Selection of nodes in close region is static. Nodes using direct or multihop routing will be stick to it till their death. Proper usage of this blend can enhance the network life.

Chiu-Kuo Liang, Jian-Da Lin and Chih-Shiuan Li [27] in this paper suggested that nodes in grid can save energy in routing the data. In this scheme named Steiner Points Grid Routing (SPGR) protocol, the nodes only in grid can communicate to each other, rest of the nodes will not be eligible. Authors have improved the deficiencies in already present grid routing protocols.

Yan Sun et al in [28] claim the reduction in energy consumption and quality of delivered data by using their technique. In this solution authors no activity is there when there is no event to perform. Positive feed-back scheme enables the nodes to work efficiently on occurrence of any event. Only negative-acknowledgment of packets is entertained to achieve reduction in consumption of energy and bandwidth.

Shancang Li: Xinheng Wang and Zongbin Li in [29] Authors proposed a scheme for scalability in WSN. This scheme is based on hierarchical structure. Nodes are segmented into clusters in hierarchical manners. After organizing the nodes and clusters, any intra or inter-cluster routing is used to make the network scalable. Authors compared the proposed scheme with already present schemes and found it better than others.

Guowei Wu et al in [30] designed an algorithm which collects the data from source nodes in an efficient way. Nodes are grouped in a group and a mobile agent is allotted to each group. This mobile agent is sent in the network to find the next hop, in case of unavailability of next hop, mobile agent goes back to current node. Current sensor node then sends the mobile agent to the node next to the failure one. By this data aggregated properly. This technique uses less number of mobile agents and so reduces the overall energy consumption.

3.2 Limitations in Literature Survey

We keenly study the research work of researchers related to energy efficiency and found that there should be something new and unique from conventional direct and multihop routings. As published work is either on multihop routing or direct routing, in few cases they might work fine but not for most of the cases. For large networks these techniques are not suitable as network lifetime will be minimized and results in poor utilization of energy.

3.2.1 Limitations of Direct Routing

In case of large network size, if direct routing is selected as mode of routing which means all nodes will send their data directly to Cluster Head. The node at less distance will have more than enough energy as compared to the node at maximum distance from Cluster Head. The most distant node will have to invest more energy to send its data to Cluster Head. So after sending very small amount of data packets it will be dead which leads to small network lifetime. So using direct routing in large network size is not feasible.

3.2.2 Limitations of MultiHop Routing

If multihop is used for routing in a large network even than short network lifetime will be the fate of WSN, because every node sends its data next to it, and the node who received the data send its own data as well as received data to its immediate node towards cluster head, so it has to invest the double of its energy. In same case the nodes closest to the CH will get maximum burden to be sent to CH and they become dead in no time as they have to transmit huge number of data packets which swallows its maximum energy.

3.2.3 Limitations of Adaptive Intra Cluster Routing

Nodes are grouped into two groups in Adaptive Intra Cluster Routing. The group of nodes close to the cluster head is termed as Close Region will always communicate to cluster head using Direct routing and the nodes in 2nd group, which are far away from cluster head will always communicate using Multihop routing. By this, nodes near to Close Region become overloaded and exhausted early and cause the end of network lifetime. However few nodes remained with excessive energy with them, so energy is not fully utilized or managed properly. Above mentioned grouping took place in the very beginning of communication and remains constant till last. So this solution is completely static in nature, besides the grouping of nodes, routing in the groups will remain constant as well. Solution is less scalable as well; it doesn't have ability of adding or removing any node at runtime.

3.3 Problem Statement

After going through all of the above research works, few issues and deficiencies are explored which are essential to be addressed: Efficient use of energy, lifetime, scalability, management and maintenance of Wireless Sensor Networks.

Most important of all above is the "Energy" issue which is the core concern as has direct influence on the network lifetime. So an energy efficient routing protocol is mandatory as energy consumption is the focal issue in WSN.

At present two main routing techniques are widely used at commercial level, direct and multihop routing. Normally for small networks Direct routing is used and Multihop for large networks to optimum energy consumption. But there occurred some problems in large network upon using multihop routing i.e. Far most nodes send data to its immediate neighbor towards cluster head. Now this neighbor has to send its own data as well as the data of previous nodes to next hop. By this, nodes near to cluster head have to send their own data as well as data of nodes behind them to transmit towards the cluster head. The term Network lifetime or life of a network is defined as: Time from the start of network till the death of first node in a network. So in multihop closest nodes can't survive longer due to heavy load of transmit. Such mishap can also occur in direct routing. Nodes at the borderline of network need more energy to transmit, as energy is directly proportional to distance. To overcome this problem *Hybrid Adaptive Intra Cluster Routing* was designed, that improved the problem till some extend by reduction in energy consumption. In this protocol decision of making "close region" is taken in the start and they remain constant till the network life ends. In the light of above mentioned issues a more scalable solution is needed to resolve the energy issue in wireless sensor network and prolongs its lifetime. Problem statement is quite clear in the figure 3-9.

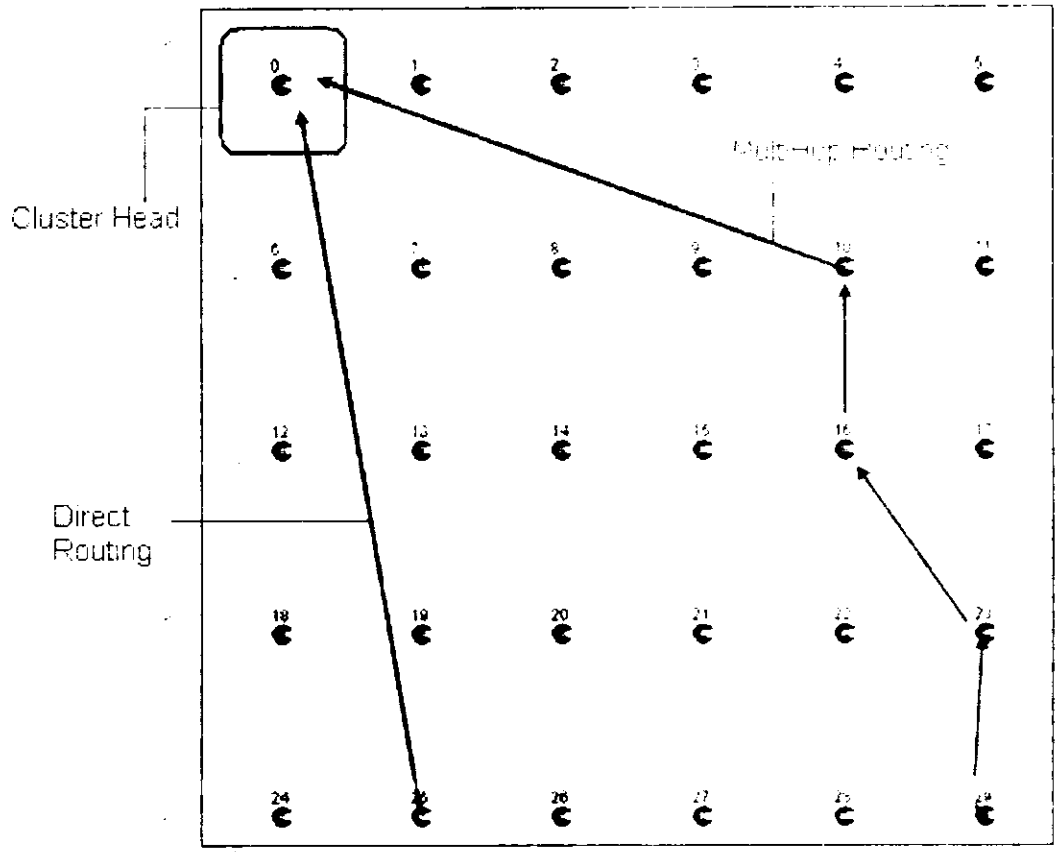


Figure 3-9: Problem Statement

3.4 Research Objectives

There is a quite enough room to work on adaptive routing. So a Hybrid Adaptive Routing Mechanism is necessarily required to fill the existing gap in this field. The major objectives for the development of our new *Hybrid Adaptive Routing Technique* are:

- Should have support for large no of nodes.
- Take the benefits of both Direct and MultiHop routing.
- Should be better than traditional routing techniques.
- Should be energy efficient.
- Increased Network lifetime.
- Should have capacity of adding and removing nodes into cluster.

4

PROPOSED ROUTING MECHANISM

4. PROPOSED ROUTING MECHANISM

ENERGY EFFICIENT & HYBRID ADAPTIVE INTRA CLUSTER ROUTING

This proposed routing solution is a complete in its nature for the above discussed problem in WSN. As it covers every aspect of present problems related to energy consumption. Now we will discuss the design features and proposed technique.

4.1 Design Features

Our designed protocol "Energy Efficient & Hybrid Adaptive Intra Cluster Routing" possesses following features.

- Should support large no of nodes.
- To take the benefits of both Direct and MultiHop routing.
- Should be better than traditional routing techniques.
- Increased Network lifetime.
- Should be energy efficient.
- Should have capacity of adding and removing nodes into cluster.

4.2 Energy Efficient & Hybrid Adaptive Intra Cluster Routing

Out of multiple domains for research in Wireless Sensor Network routing is ranked at the top. As existence is a key point, if a node became dead because of low energy, it has no more existence, its whole hardware / software is of no use unless its battery is charged or replaced. So to make a node alive for a longer period of time is the most critical issue and it can be solved by an appropriate routing protocol.

Main purpose of this research work is to increase the network lifetime of WSN, which is a major challenging issue. In this algorithm each node decides during run time to select the mode of routing depending upon its energy level that when to route the data using direct routing and when by multihop routing. Finally the energy consumption is reduced which results in the extension in network lifetime also increase in number of packet sent in network.

There are three main phases in the whole process that are:

1. Cluster Head selection
2. Cluster formation
3. Intra Cluster Routing (Inter Nodes Routing)

We worked on the Intra Cluster Routing only, as is the scope of our work. For first two phases any energy aware technique can be exercised.

4.2.1 Intra Cluster Routing (Inter Nodes Routing)

With the help of RSSI value (Received Signal Strength Indicator) distance of each node can be calculated. Value of RSSI is inversely proportional to the distance, greater the RSSI value smaller the distance or vice versa.

Each node checks its energy level, then on its energy level, it will be decided that which routing technique is to be used? If the node has energy level greater than a certain threshold, it will use Direct routing, in case of vice versa Mulihop routing will be best option.

In start all nodes have maximum energy; ultimately Direct routing will be adopted. With each packet send, energy consumes depending upon the distance. After sending each packet energy level is checked. Nodes have powerful dynamic transceivers, which consumes energy according to the power by which data is sending. More distant nodes need more energy to send data to its parent node, which may be the cluster head, base station or any other node.

Direct routing has no issue of parent selection but in multihop it is a tough job. We have used "Shortest Path Selection" in our work. For this each node has to maintain its neighbor table, by which the shortest path towards cluster head is calculated.

While routing when energy level of any node reaches to an end, that node will stop sending data and node will be considered as dead. At this time our algorithm calculates the lifetime of network, energy consumed and number of packets sent to the cluster head.

By implementation and simulation of our technique, it is calculated that overall network lifetime is increased, energy consumption of each packet is decreased and number of packets sent in the network increased remarkably.

The flow diagram of the proposed idea is illustrated in **Figure 4-1**.

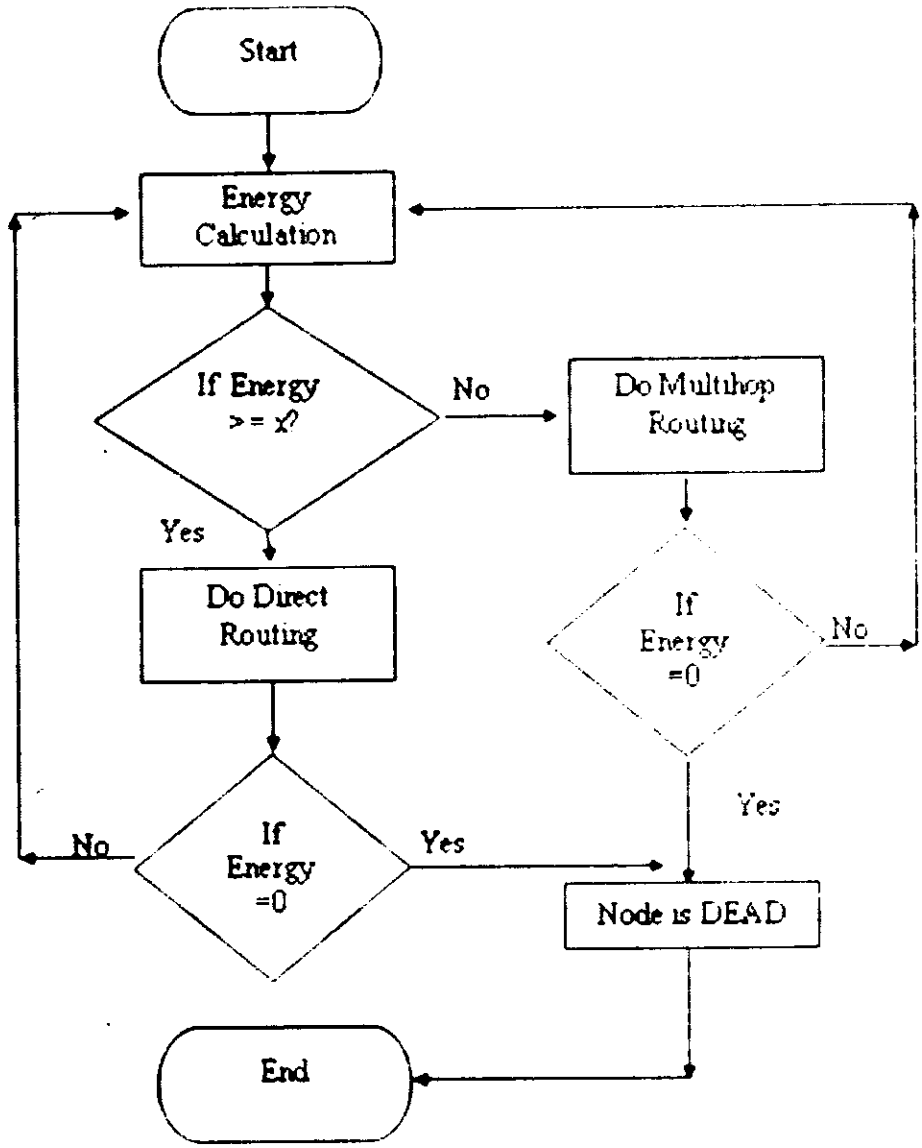


Figure 4-1: Flow Chart - Decision to select Routing (Direct Multihop)

4.2.2 Pseudo Code for Intra Cluster Routing

START

```
**********************************
Destination ID = D-ID
Destination Address = DA
Source ID = S-ID
Parent ID = P-ID
```

```
Cluster Head = C.H
Base Station = B.S

**********************************

Initialization
          S-ID= Source Address
          D-ID= DA
          P-ID= Parent Address
          Data, Energy Level
// End of Initialization

     Get Sensed Info
     Data= Info
     Select C.H
     IF (Energy > x) THEN
          Call DIRECT-ROUTING
     ELSE
          Call MULTIHOP-ROUTING
```

DIRECT-ROUTING

```
     Sense required info
     Data = Info
     D-ID = C.H

     WHILE (Energy Level ≠ 0)
          Send Data to D-ID
     END WHILE
Broadcast "Node is dead"
```

MULTIHOP - ROUTING

```
     Sense required Info
     Data = Info
     Calculate P-ID for data Delivery
```

```
      D-ID = P-ID

WHILE (Energy Level ≠ 0)
          Send Data to D-ID
END WHILE
      Broadcast "Node is dead..."
```

END

4.3 Step by Step Visual Explanation

The algorithm first calculates the energy of each node. After that it checks if all nodes have enough energy to send the data using Direct routing. If test is true the data will be sent by using Direct routing, as in the start every node will have full energy so Direct routing will be adopted. Whole process is shown in the Figure 4-1. **Figure 4-2** is showing it is the start, i.e. t0, every node is communicating directly to the base station.

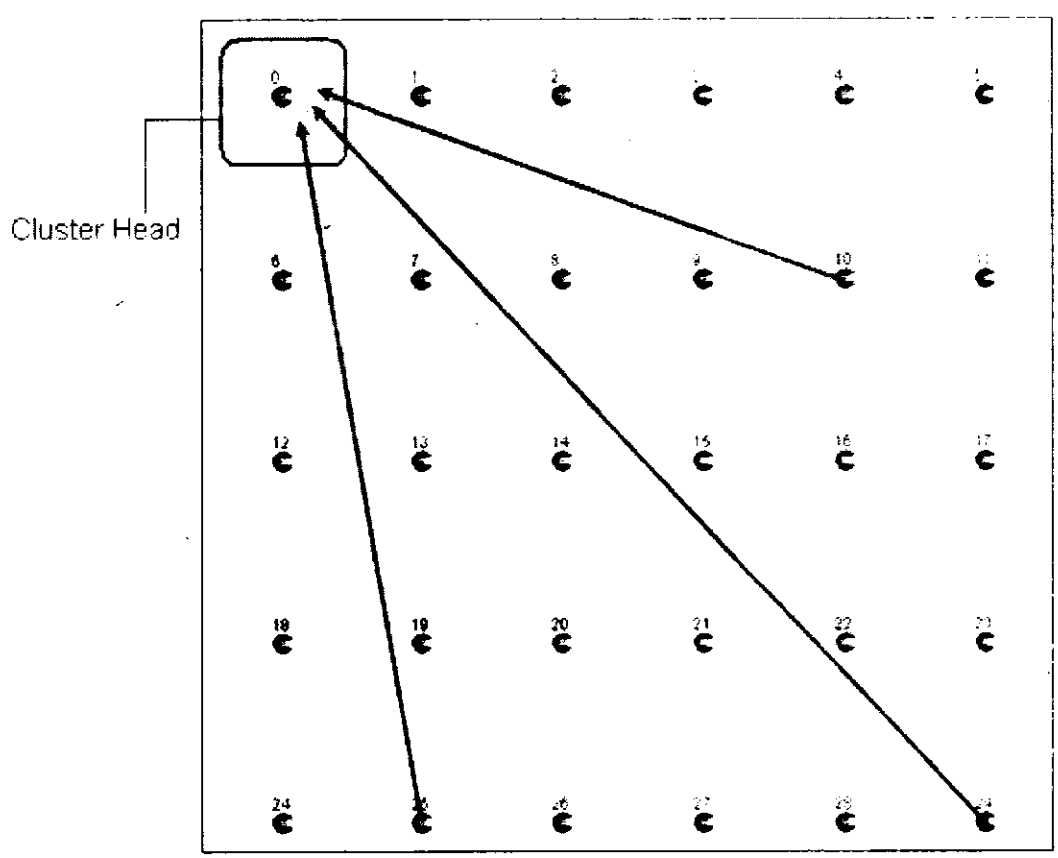


Figure 4-2: Situation at time t_0

After some interval at time t_1 the nodes which are at the long distance from cluster head will face shortage of Energy, then they forms a "Low Energy Region" and start MultiHop routing in Low Energy Region, shown in **Figure 4-3**.

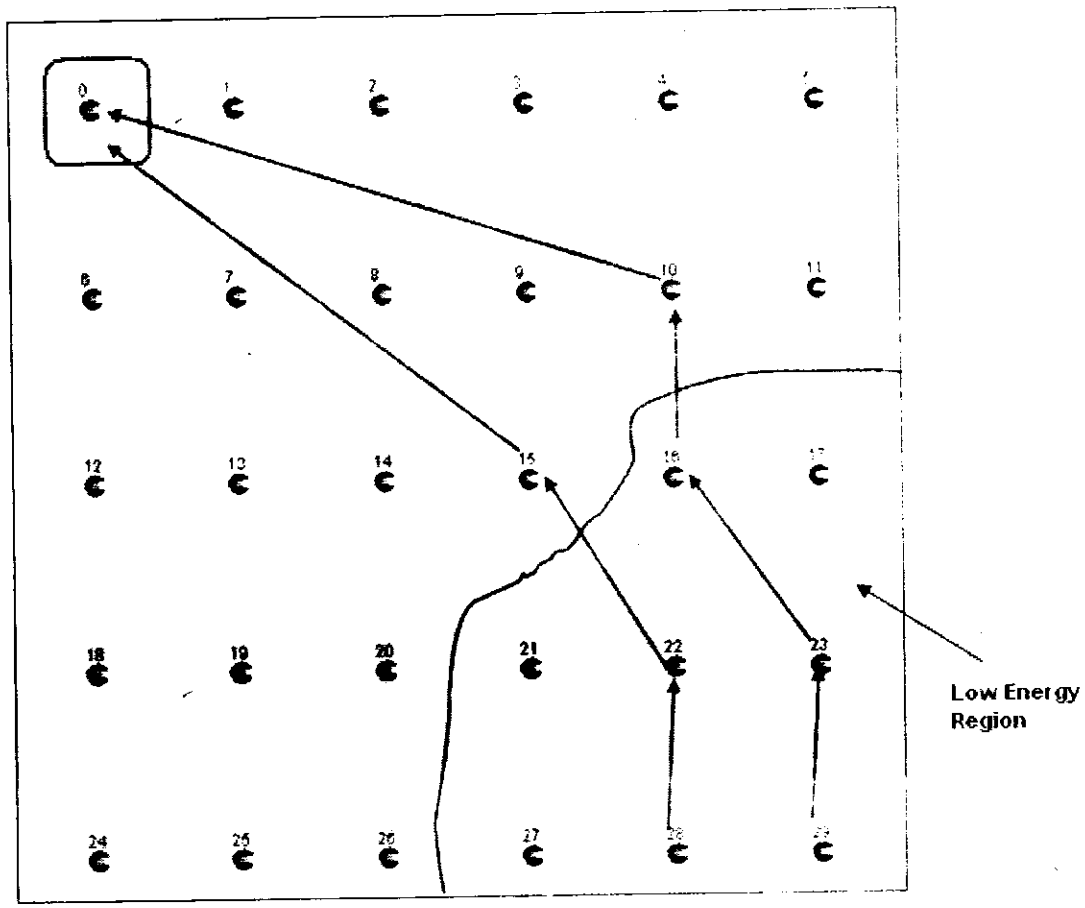


Figure 4-3: Situation at time t_1

The reason to use MultiHop routing is that at time t_1 energy level of the nodes of the "Low Energy Region" reduces to a specific threshold.

At time t_2 nodes which are at most distance from Cluster Head (excluding the nodes of "Low Energy Region") also becomes the part of the Low Energy Region and the Low Energy Region will be expanded. Shown in Figure 4-4.

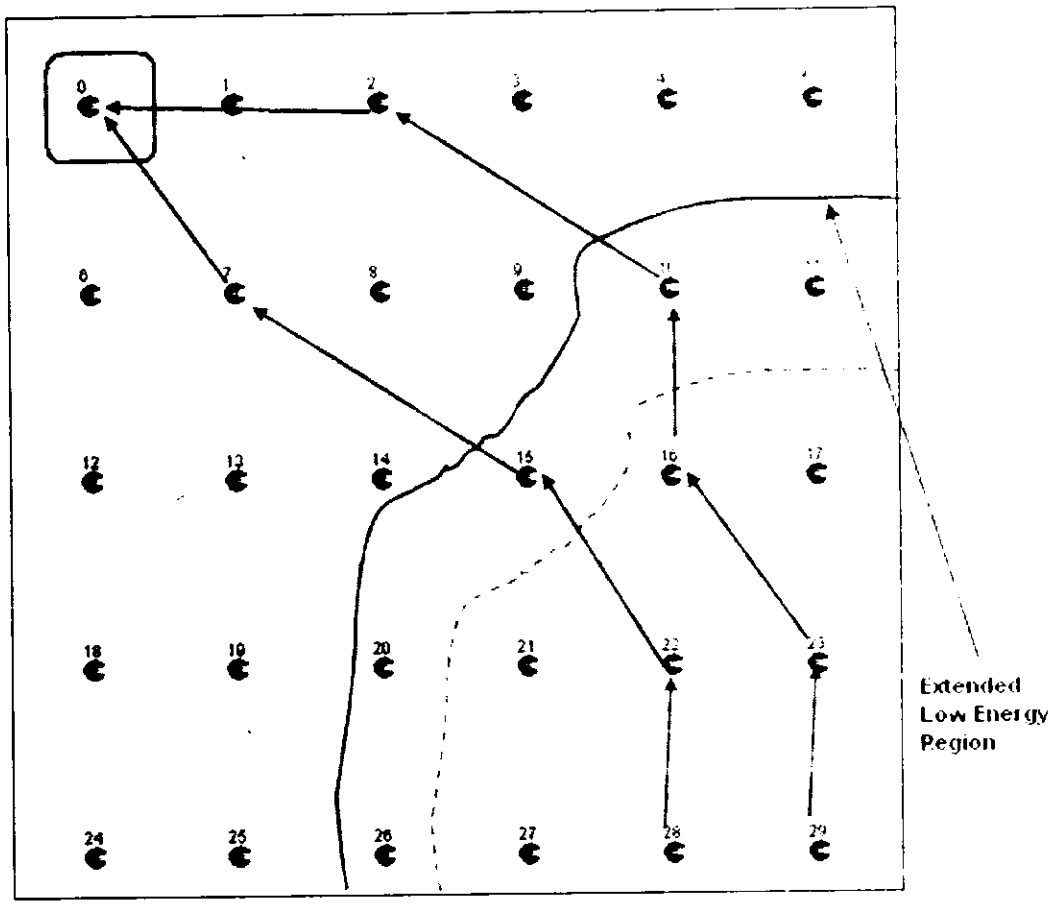


Figure 4-4: Situation at time t_{n-1}

At time t_n few nodes which are close to the cluster head will be near to die because of the full load of "Low Energy Region" on them. Like in above figure all load of Low Energy Region is at node number 2 and 7 and these are acting as a gateway for Low Energy Region to Cluster Head. So at this point of time a new close region will start forming from the nodes closer to the Cluster Head. Figure 4-5 shows another "Low Energy Region".

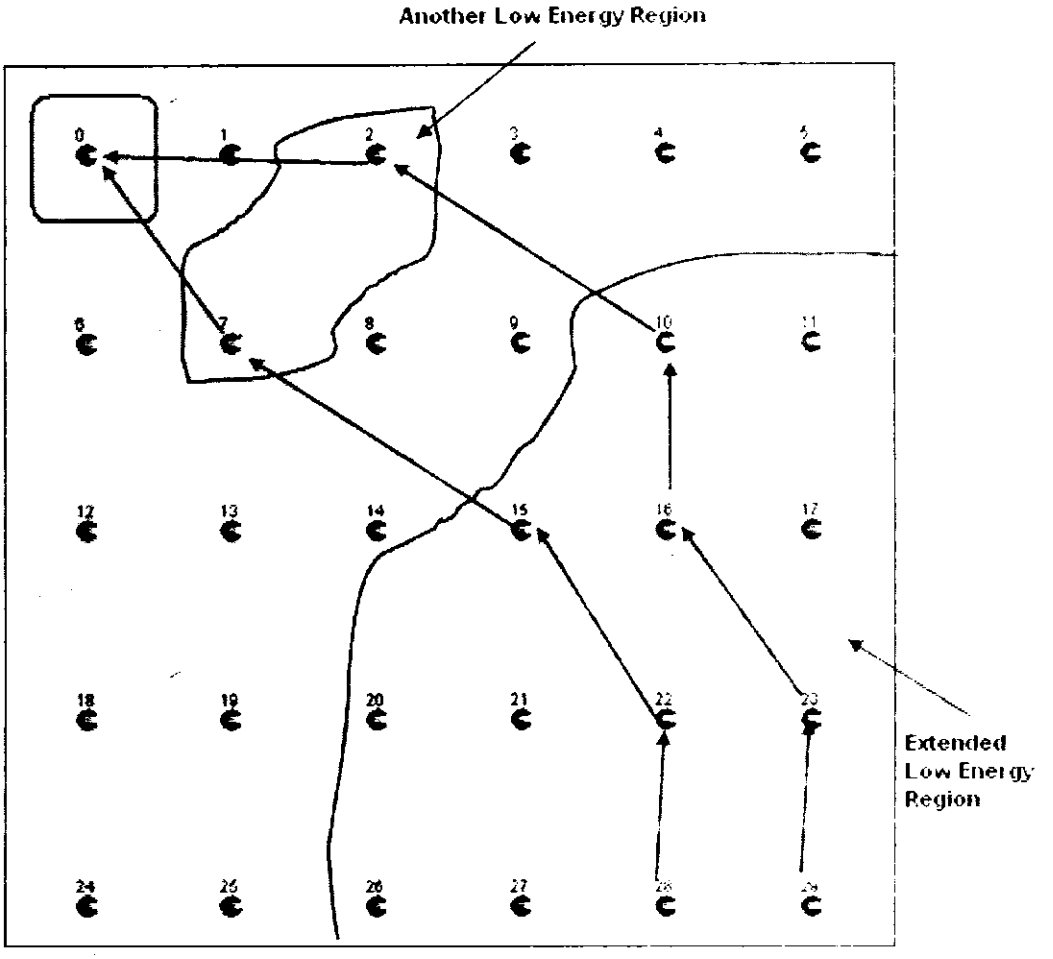


Figure 4-5: Situation at time t_n

By implementing this mechanism energy of nodes became more balanced. Number of packets sent in the network is increased. Every node utilizes maximum energy, where as average energy consumption is decreased. Hence network life increases.

5

SIMULATION DETAILS

5. SIMULATION DETAILS

In this chapter the implementation details of our research work has been discussed.

5.1 Tiny OS

We have used TinyOS as working environment for Wireless Sensor Network. TOSSIM simulator is available with TinyOS operating system.

5.2 TOSSIM Simulator

A simulator is "a device, instrument, or piece of equipment designed to reproduce the essential features of something" [27]. It is recommended that before going towards real implementations one should go through the simulation process because in simulation it is quite easy to check and track the errors and debug the application. In case of WSN, the real mote is a costly hardware and is not commonly available for research. Only few research institutes have real motes for research work. So the only option remained behind is simulation. There are many simulators available in the market for simulation of wireless networks e.g. NS-2 [28], GloMoSim [29], EmStar [30], SensorSim [31], QualNet [32], OPNet [33] and SENSE [34] but TOSSIM [4, 5] is selected due to its best features and easy to understand.

TOSSIM is a top ranked and dominant simulator for TinyOS. Simulation of a network with thousand plus nodes can be easily handled by TOSSIM. TOSSIM is offering a best deal that code can be burnt in motes without any further modification after compiling it in TOSSIM. TOSSIM is quite easy and friendly in usage and in debugging of code as compared to other simulators.

TOSSIM mainly focuses on simulation rather than simulating real world. It like other simulators makes some assumptions and focuses on some special behaviors. Some of the characteristics of TOSSIM are as follows.

5.2.1 Scalability

TOSSIM is able to handle network with large number of nodes. Thousands of nodes in a network can be entertained. Individual mote has very small resources. Static component memory model simplifies the state management.

5.2.2 Completeness

Behavior of TinyOS is simulated by TOSSIM at very low level, and captures complete system behavior. TOSSIM bridges the gap between algorithms and implementation. It has instrumented nesC compiler. It has capability to deploy the tested code directly without requiring any change in application.

5.2.3 Fidelity

Fidelity is to deal with accuracy. Reliability is a major concern in any simulator, which can be judged by its fidelity. Bit-level simulation of TOSSIM can capture the network at very high fidelity. TOSSIM helped in debugging some of the TinyOS network stack problems, which were overlooked during the test bed deployment.

5.2.4 Simulation Time

In TOSSIM, simulation time is 4 MHz which is same as CPU clock of mica platform. 4 MHz is equal to 4×10^6 ticks/second. The code in TOSSIM runs without any preemption.

5.2.5 Flexibility

TOSSIM has a flexibility of adding new models. Some real world models are not provided in TOSSIM but by connecting some external tools required functionality can be achieved and used for the perfection of a given simulation. Connecting to any other external tool couldn't affect its efficiency.
Few external models are as follows:

5.2.5.1 Radio

Radio model is not available in TOSSIM. A java based tool named as "LossyBuilder", can be used to generate the desired radio model. By "LossyBuilder"

An input file will be given to TOSSIM at run time to add radio model while simulating the specific application. LossyBuilder help us to attain characteristics like Bit corruption, Delay and Propagation like real world. Chances of loss of each packet may vary from others as not dependent on other packets.

There are two main radio models which are normally in use, Free Space Model and Two Ray model. As we are considering that there is hindrance in the line of sight of each model so in our research work "Free Space Model" is used.

5.2.5.2 Power / Energy

To calculate the energy consumption is not included in TOSSIM. But energy consumption parameter can be simply added for any component used. The overall energy consumption of that component can be calculated after the simulation by using that parameter.

5.2.5.3 Building

TinyOS implementation of any protocol is required for its simulation because TOSSIM directly builds from the code. Apparently it is difficult task to do but one can simply run that code on real motes with no change.

5.2.5.4 Imperfections

Nothing is 100% in this world and TOSSIM also has some limitations with it. As discussed before TOSSIM run the simulation instantaneously so no preemption is involved during the simulation. But in real motes there can be preemption by any interrupt occurrence, so the simulation never stops due to none of any interrupt handling in TOSSIM.

5.2.5.5 Networking

40Kbit RFM networking stack is simulated by TOSSIM. It also includes MAC, timing, encoding and acknowledgements.

5.2.5.6 Authority .

Performance analysis of the algorithms can be done through TOSSIM. But the results of respective simulation are not so much authoritative. TOSSIM can tell you high data

loss between two algorithms but like real mote the data loss scenario cannot be perfect.

5.3 How to Compile an Application in TOSSIM

TOSSIM is basically a Linux based application. But can be used and configured in Windows as well, for windows a Linux emulator is needed. We have used Cygwin, which is a CLI based Linux emulator like "Putty". For simulation first install the Cygwin then configure the TinyOS.

All versions of TOSSIM have capability to compile any piece of code written in NesC language. Open the Cygwin shell, go to the application directory. Write "make pc" and press enter. It will compile the application and show errors and warnings if any. In "*make pc*", *pc* is a platform, used only for simulation. Other valid platforms are *mica*, *mica2* and *mica2dot*. These platforms have different names in different versions, like this "*make pc*" is for TinyOS 1.x but for TinyOS 2.x "*make sim*" is used.

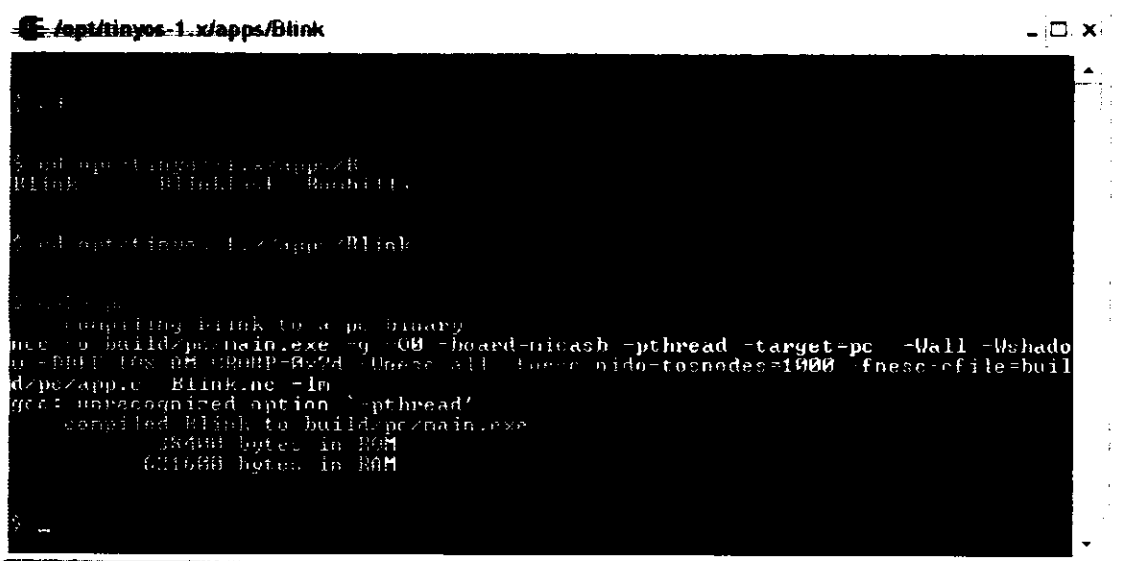


Figure 5-1: Compiling an Application

When compilation of the application is complete a new directory *build* will be created in the folder directory of application, as shown in Figure 5-2. This directory contains all the compiled files in it. All commands are clearly mentioned on the screenshot of Cygwin shell.

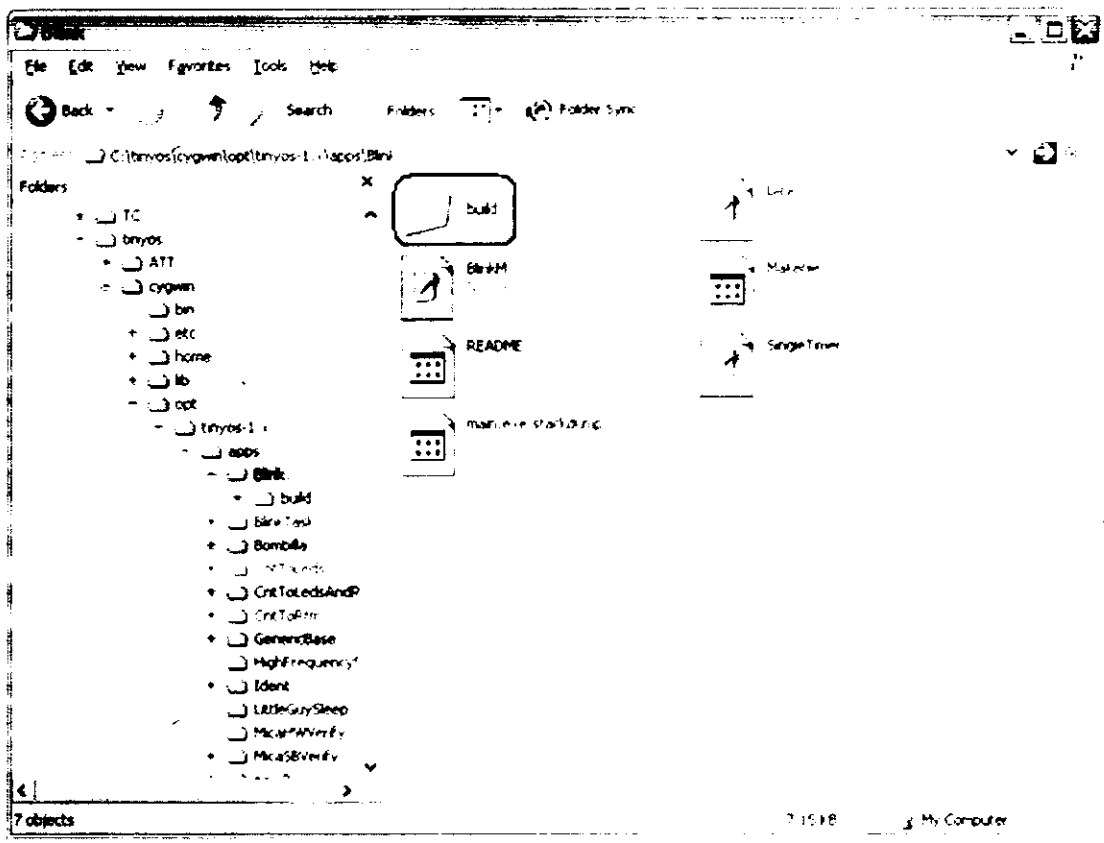


Figure 5-2: Directory Structure

Inside the *build* directory there are sub-directories according to the platforms we have compiled for, e.g. for simulation ("*make pc*") there will be a directory named "*pc*" and for hardware ("*make mica*") there will be a directory named as "*mica*" respectively. In *build* there is a further more subdirectory named as *pc*. This *pc* contains the compiled file "*main.exe*" in it, which is an executable compiled file. Figure 5-3 shows all the contents of *build* and *pc*.

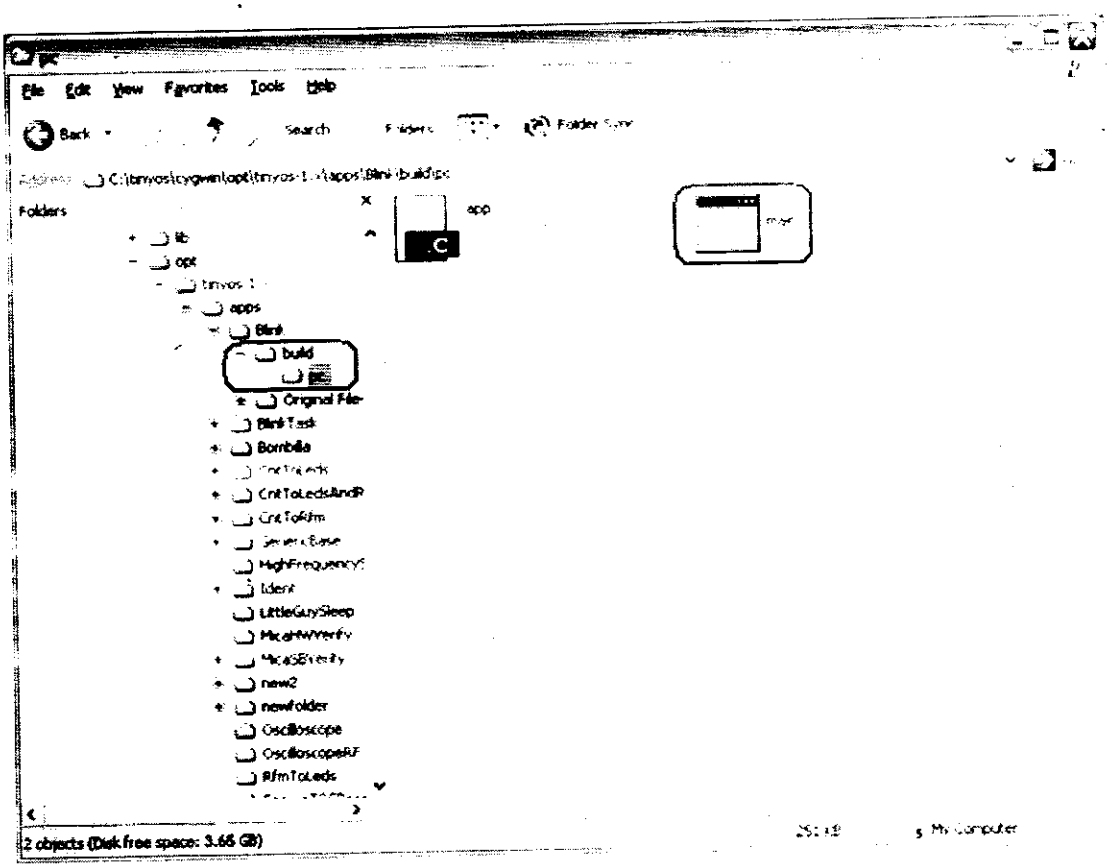


Figure 5-3: Directory Structure and main.exe location

5.4 How to Run an Application

After compilation of an application now we need to run that compiled application. To run a pre compiled application, open the Cygwin shell. Reach to the directory of application, which we need to run. This is the time to give the command for execution. But before running the application we need to specify the debug variables first, because it will enable all the debug variables which results in a huge output which is quite difficult to understand. First let me show you the output without specifying debug variables.

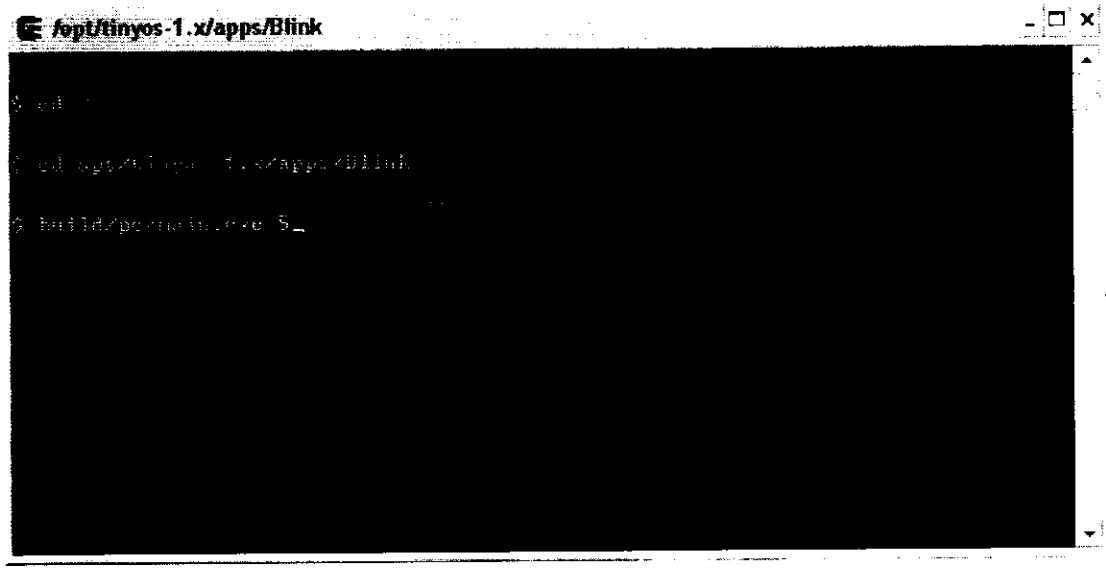


Figure 5-4: Command to run the application

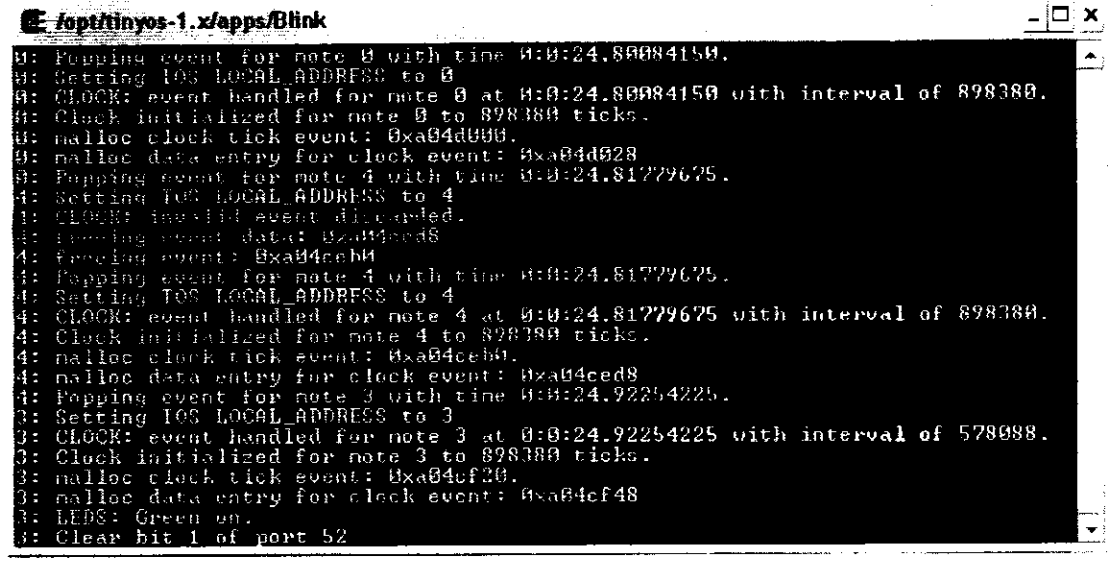


Figure 5-5: Output of running Application

This output is quite complex and difficult to understand but it will be quite easy to understand the output, if we specify the debug modes before running the application. See the next figures for debug command and output.

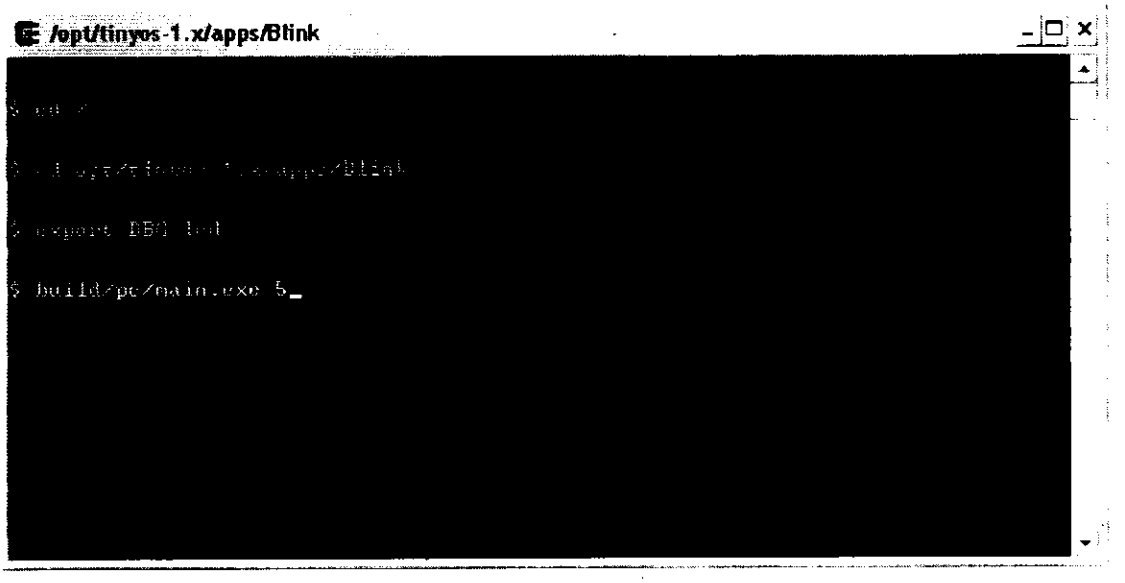


Figure 5-6: Command for debug and to run the application

Here in this Figure 5-6 we only enabled the "led" variable i.e. DBG=led is enabled. This will only show the status of LEDs on the nodes.

In 2nd command of running the application we have mentioned "5" after a space. This means that application is running for 5 nodes only. If we write 20. then application will run for 20 nodes.

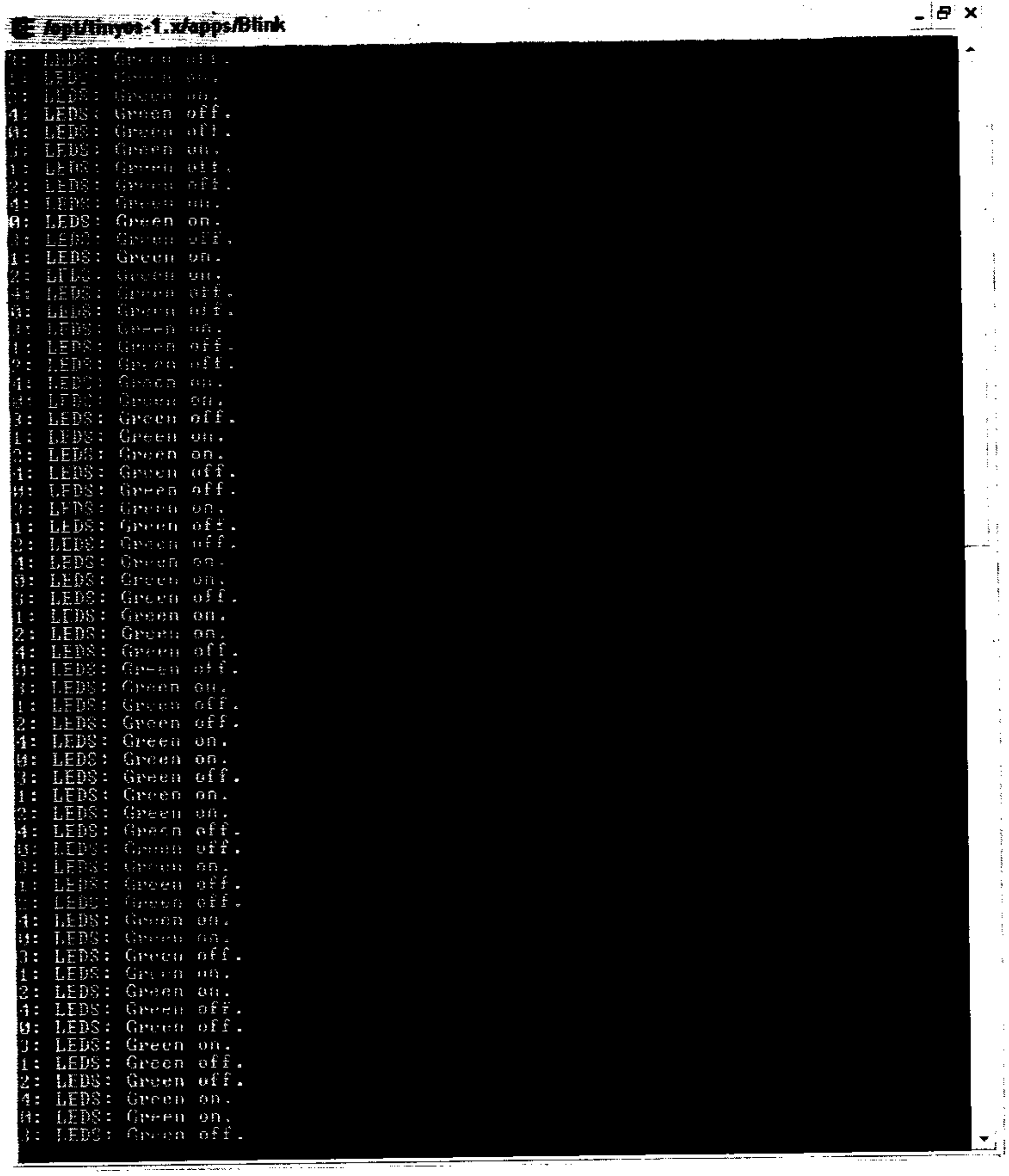


Figure 5-7: Output after specifying the debug variables

To save the output of screen on the screen in a file

$ build/pc/main.exe 10 > result.txt

This command will save the results of output in the file named as result.txt. Location of this file is *build/pc/result.txt*

5.4.1 Help for TOSSIM

-h or –help shows the help page. Shown in Figure 5-8.

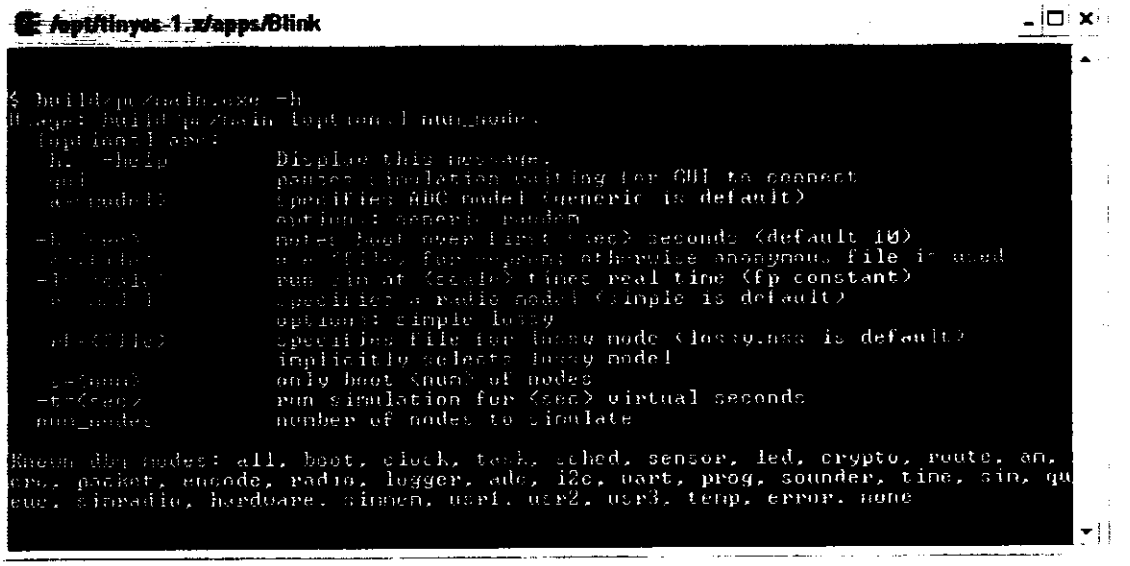


Figure 5-8: Screenshot of commands of Help

In the Figure 5-9 it shows all the possible options we can use.

-gui: This option halts the execution until the GUI application is connected.

-a=<model>: Here specify the ADC models. Out of two models Generic and Random, Generic is default.

-b=<sec>: Booting time of nodes can be specified here.

-e=<file>: File for EEPROM, anonymous file is used by default.

-l=<scale>: Runs the simulation at the rate according to real time. E.g. -l=3.0 means simulator runs the thrice of the real time.

-r=<model>: To specify the radio models. Default is "simple" having options simple lossy.

-rf=<file>: A topology file can be given as input, TOSSIM places the nodes according to this topology file. By default is lossy model.

-s=<num>: From specified network, only those numbers of nodes will be booted which are mentioned here.

-t=<sec>: Run the simulation for specific interval of time.

num_nodes: Total number of nodes used in simulation.

5.4.2 DBG Modes

In TOSSIM debugging environment can be provided before the start of simulation. DBG flags are set for each debug statement in code. Before running the application DBG modes should be specified else all the dbg modes will be enabled and we will

have to understand the huge number of statements which will be very complex. To enable single specific mode:

export DBG=led

to enable multiple modes in one go.

export DBG=led,usr1,usr2,adc,boot

The known debug modes are mentioned in help command.

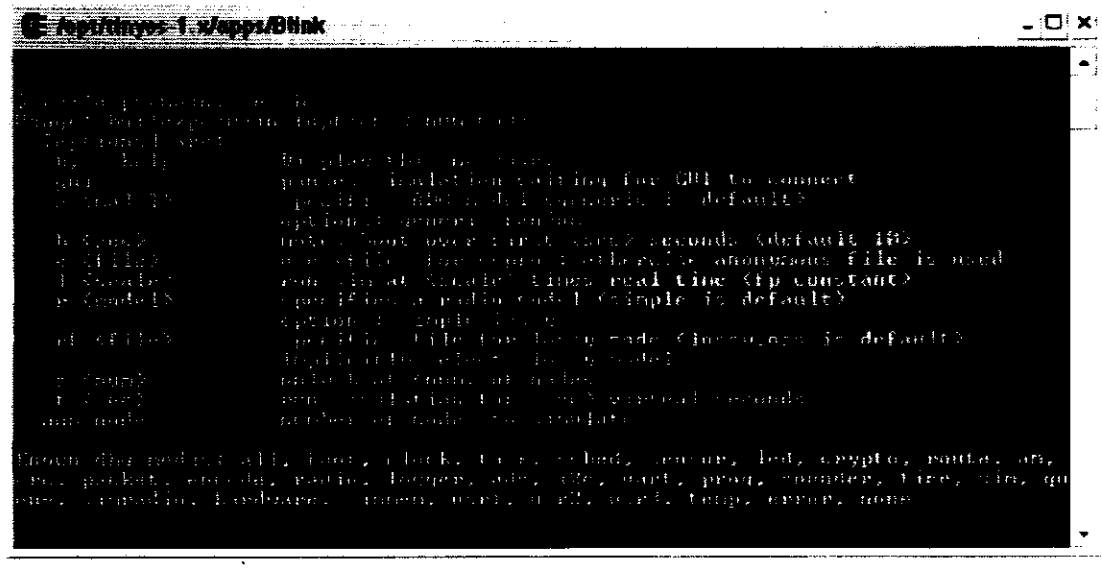


Figure 5-9: Screenshot of all modes of DBG

Let us see a little bit detail of each of the mode shown in Figure 5-9.

all: Default value: Enables all DBG messages

boot: For Boot and StdControl modes

clock: For Hardware clock

task: Tasks in Progress

sched: Scheduler of TinyOS

sensor: For sensed readings of sensors

led: LEDs of Mote

crypto: For Cryptographic operations

route: For controlling routing

am: Active messages (Transmission and Reception)

crc: For Cyclic Redundancy Checks on active messages

packet: Transmission and Reception of Packet

encode: Encoding and Decoding of Packet

radio: For radio operations like bits and bytes

logger:	For controlling of Non-volatile storage
adc:	For Controlling Analog to Digital Converter
i2c:	I2C bus controlling
uart:	For Serial Port configuration
prog:	Reprogramming the Network
sounder:	For controlling sounder available on sensor board
time:	Controlling of Timers
sim:	TOSSIM Internals
queue:	Event queue of TOSSIM
simradio:	Radio models of TOSSIM
hardware:	Hardware abstractions of TOSSIM
simmem:	Allocation and De-allocation of memory in TOSSIM
usr1:	For User mode 1 outputs
usr2:	For User mode 2 outputs
usr3:	For User mode 3 outputs
temp:	For User Temporary Mode

DBG modes are basically the flags that are set in the code and being manipulated at run time. The code debugging is done by using these dbg modes. The user can define his / her own flags using four modes which are usr1, usr2, usr3 and temp. Other than these four modes are known as system modes.

5.5 TinyViz

TinyViz is a popular GUI tool for TOSSIM. As already described that –gui option while running the application connects this tool with the application. To connect the TinyViz we first give the command to the application with -gui option.

build/pc/main.exe -gui 20

As shown in Figure 5-10.

Figure 5-10: Application waiting for GUI tool to be connected

Then open a new Cygwin shell and apply this command:

java net.tinyos.sim.TinyViz

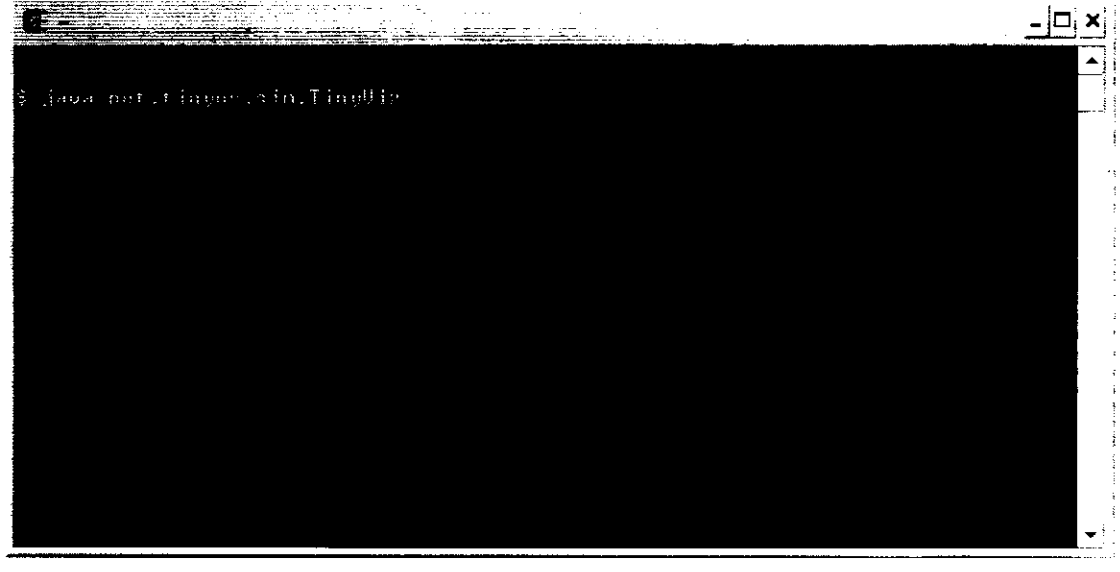


Figure 5-11: Command to Run the TinyViz

This command will run a java based tool, TinyViz.

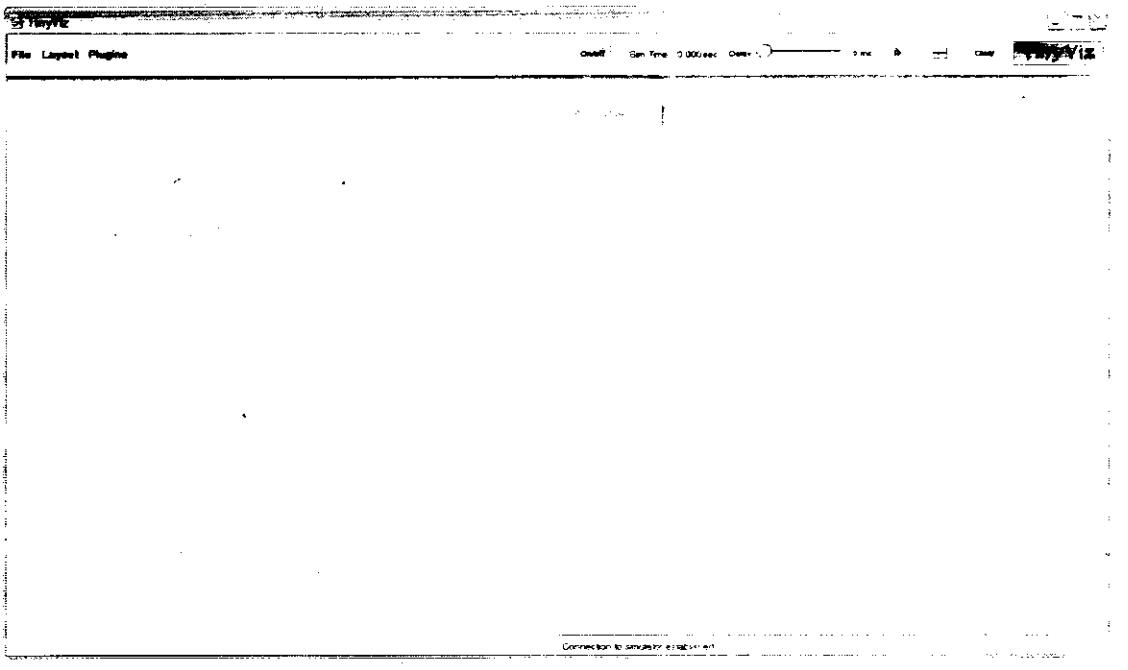


Figure 5-12: Screenshot of IDE of TinyViz

Figure 5-12 is the IDE of TinViz. There is a start button at the top toolbar. Upon clicking that button simulation will start and same button can be used to pause the simulation. On/Off button will quit the simulation. Sim Time tells the simulation time. Delay is used for controlling the speed of the simulation. By default there is no delay.

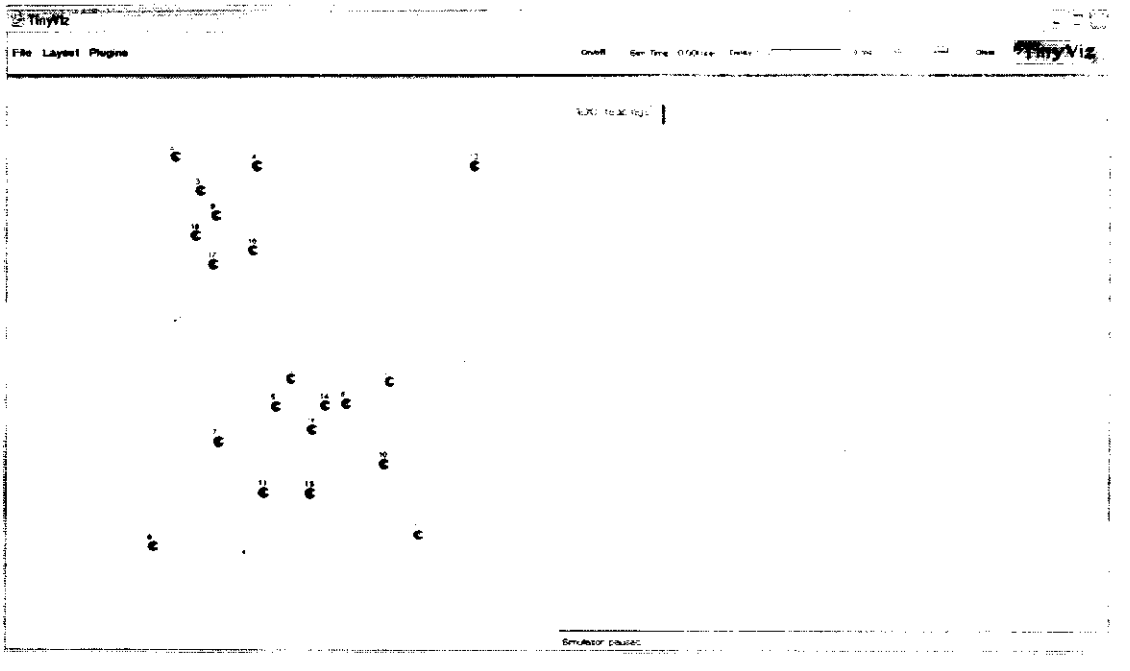


Figure 5-13: Screen Dump of TinyViz

5.6 nesC Language

"Network Embedded Systems C" abbreviated as nesC is an extension of C programming language having syntax like C language. nesC is a component-based, event-driven programming language used to write programs for the embedded devices operated by the TinyOS, which are used for wireless sensor networks.

5.6.1 Basic Concepts Behind nesC

- In nesC component based applications are developed, these applications can have bidirectional interfaces. TinyOS uses nesC programming language to develop applications to run for it.
- To install the nesC compiler we need to have its rpm package as well as need cygwin to run Linux applications in windows environment. First install the cygwin then using this cygwin shell install rpm package of nesC compiler and then configure it and set the class path.
- For the communication of components with others components nesC has interfaces as ports with the components. Only these interfaces are the gateway to interact with the components, without interfaces components cannot be accessed or used. Interfaces are bidirectional because components may provide or use these interfaces.
- A component can be an interface provider or an interface user, as key words *uses* or *provides* are used to access the interfaces. For communication every component must need an interface to communicate with other components or to access the other components.
- An interface must be implemented by the user component if it is once used. So it is quite clear that interfaces make complex interaction between the components, for example after using the interface "*Send*" the Send.send instance can't be called until and unless Send.sendDone instance is implemented.

Class path setting is of vital importance in configuration of nesC compiler in TinyOS environment. For all applications two files for nesC programming must be created.

one for the wiring of interfaces called configuration file and other for actual implementation is called Module file. Extension of nesC file is .nc.

5.6.1.1 Configuration File

Components' connection or wiring with each other is mentioned in configuration file. In addition to wiring one component to another, configurations also need to export interfaces.

The code mentioned below is taken from Blink application written in nesC. How components are bound together through their interfaces is quite clear. This bounding process is denoted by an arrow ->. The component at left side of the arrow uses the interface provided by the component at right side of the arrow.

```
configuration BlinkTask {
}
implementation {
  components Main, BlinkTaskM, SingleTimer,
LedsC;

  Main.StdControl -> BlinkTaskM.StdControl;
  Main.StdControl -> SingleTimer;

  BlinkTaskM.Timer -> SingleTimer;
  BlinkTaskM.Leds -> LedsC;
}
```

Here interface StdControl is used by Main component which is provided by component BlinkTaskM. The configuration file starts the block with its own name enclosed by braces, as shown in code above.

5.6.1.2 Module File

There are two main files with same name like BlinkTask, but 2nd file has an extra letter M (Capital M) at the last of filename, i.e. BlinkTaskM. This 2nd file with M is the module file which contains the actual implementation of the application. Few lines of module file are as below:

```
module BlinkTaskM {
  provides {
    interface StdControl;
  }
```

In module file interfaces used or provided are mentioned by the key word uses and provides. As mentioned above Main component is using the StdControl interfaces provided by component BlinkM. In module file this will be shown as:

```
module BlinkM
{
Provides
{
      interface StdControl;
      }
}
```

It shows that the component BlinkTaskM is providing the interface StdConrol.
How to set the dbg flags in nesC code. For debugging we need to set the DBG flags in code and at run time we can easily debug the application by enabling the specific DBG mode.

```
dbg(DBG_USR2, "Timer fired:  %d" ;
```

The DBG flag USR2 is set and given a string value "Timer fired". Now at runtime when we will enable this mode then this output will be shown at screen.
We can get different variable values and can easily manipulate them for complex calculations using this facility.

5.7 Simulation Process of Proposed Solution

Here are the details of simulation of our proposed solution. To implement our proposed idea we used 30 nodes in a cluster with node IDs from 0 to 29. Node 0, at the top left corner is Cluster Head. The proposed algorithm of Hybrid Adaptive Intra Cluster Routing runs between 29 member nodes having IDs 1-29 of the cluster. The screenshot of our simulation is in the Figure 5-14:

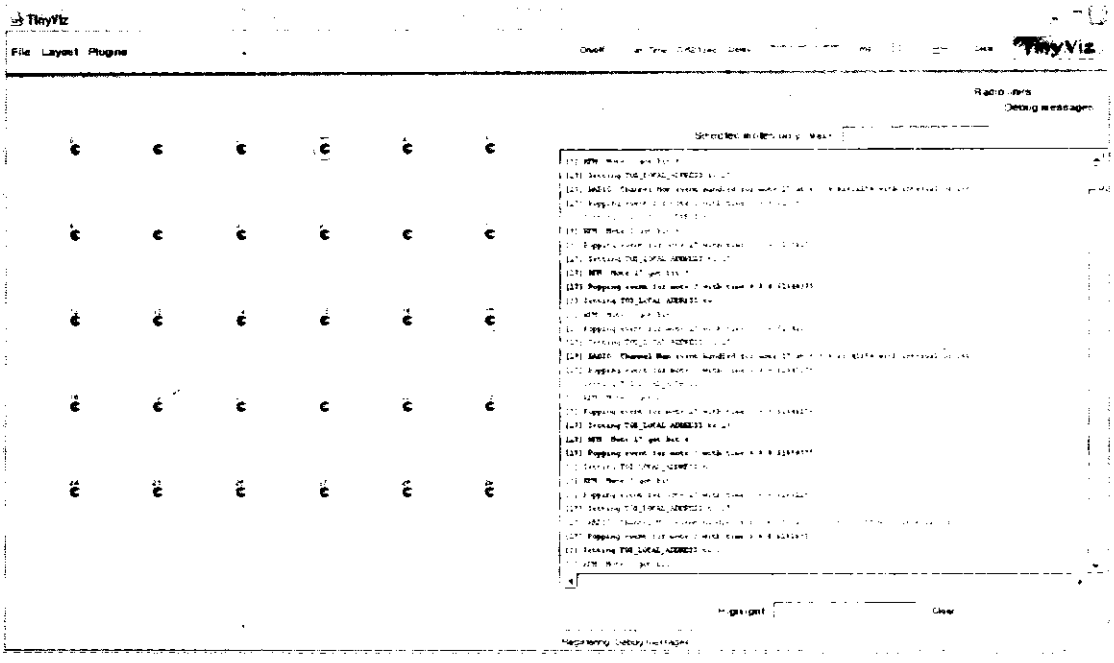


Figure 5-14: Screen Shot of Simulation

5.8 Parameters for Simulation and Tests

We performed multiple tests and experiments by simulating different scenarios and using different parameters.

- All nodes are in grid layout.
- Distance among node to node is fixed, that is 10 feet.
  ["Lossy Builder" a Java application, is used to set the distance parameter]
- All nodes have same physical environment.

Three tests are stated below. Results are shown in next chapter, keeping in view the parameters described below.

5.8.1 Test No. 1

Same energy level of each node is considered to check the energy consumption of our proposed algorithm that is 300 mJ. This simulation is only for 30 nodes, same algorithm can be checked with large number of nodes and different physical environment in future, which is mentioned in future work. Simulation time is approximately 20 minutes. In the table below contains the list of parameters considered for the simulation.

Table 5.1: Simulation Parameters - Test No. 1

Parameters	Value
Layout Topology	Grid
Simulation Time	20 min
Maximum Energy level	300 mJ
Distance Unit	Feet
Cluster Size (Nodes)	30
Cluster Head	Node 0
Broadcast ID	65535
Geographical Area	2000 Sq Feet

5.8.2 Test No. 2

In this 2nd test we doubled the energy level. Initial energy level of each node is taken 600 mJ. Time consumed to simulate the results is 40 minutes. Rest all the parameters are kept constant. Table 5.2 shows the parameters of simulation considered for this test.

Table 5.2: Simulation Parameters - Test No. 2

Parameters	Value
Layout Topology	Grid
Simulation Time	40 min
Maximum Energy level	600 mJ
Distance Unit	Feet
Cluster Size (Nodes)	30
Cluster Head	Node 0
Broadcast ID	65535
Geographical Area	2000 Sq Feet

5.8.3 Test No. 3

Table 5.3 contains the parameters considered for next test. Here initial energy level of each node taken is 900 mJ and simulation is run for 1 hour. Rest all the parameters are kept constant.

Table 5.3: Simulation Parameters - Test No. 3

Parameters	Value
Layout Topology	Grid
Simulation Time	60 min
Maximum Energy level	900 mJ
Distance Unit	Feet
Cluster Size (Nodes)	5
Cluster Head	Node 0
Broadcast ID	65535
Geographical Area	2000 Sq Feet

6

RESULTS

6. RESULTS

Simulation mechanism, simulation assumptions and simulation results has been discussed here and explained by graphs. These results show the efficiency of suggested solution and objectives of research are attained.

To simulate the proposed solution we used TOSSIM simulator and operating system is TinyOS. Results are deeply observed then compare them with the existing solutions and found it more result oriented in terms of energy consumption. We used 30 sensor nodes for the simulation of our proposed technique and evaluated it with the Adaptive, Multihop and Single-hop routings. Multiple tests have been conducted to achieve the most favorable results. After that we evaluated the proposed mechanism with all three above stated techniques to demonstrate the effectiveness of proposed algorithm using following performance metrics.

6.1 Parameters

We considered the following metrics of performance:

1. Number of Packets sent in the network
2. Total packets sent to the Cluster Head
3. Energy consumed by the network at a certain time t_n
4. Energy left at a node after a certain time t_n
5. Network lifetime

6.2 Results of Simulation

In the first graph Figure 6-1 Hybrid Adaptive techniques is compared with other three techniques of Direct, Multihop and Adaptive routing. Resulting graph shows that our proposed technique has capability of sending extra packets of data in network than rest of three. In our proposed technique all nodes first send data using Direct method until their energy level reaches to a certain level. After attaining that certain level of energy each node will start sending data using Multihop. This technique in contrast to Adaptive, does not specify the nodes in the start that these specific nodes will always use Direct routing and the others will always send data using Multihop, but this

decision of selection of routing techniques is made at runtime depending upon the energy level of the nodes.

As in the start all nodes send data by using Direct method so nodes being close to cluster head require less energy to send data. but they have to send the data of other nodes as well when far most nodes reach at a certain energy level so . That's the reason the nodes near to cluster head have not sent maximum data items as like Adaptive technique. Graph shows that in our proposed technique nodes near to cluster head have high rate of data sending because in start they had less distance to send data and no load of far nodes on them. Whereas the far nodes first send less amount of data but later on send more data by using Multihop. So it is clear from graph that through put of our suggested mechanism is more than the three other techniques.

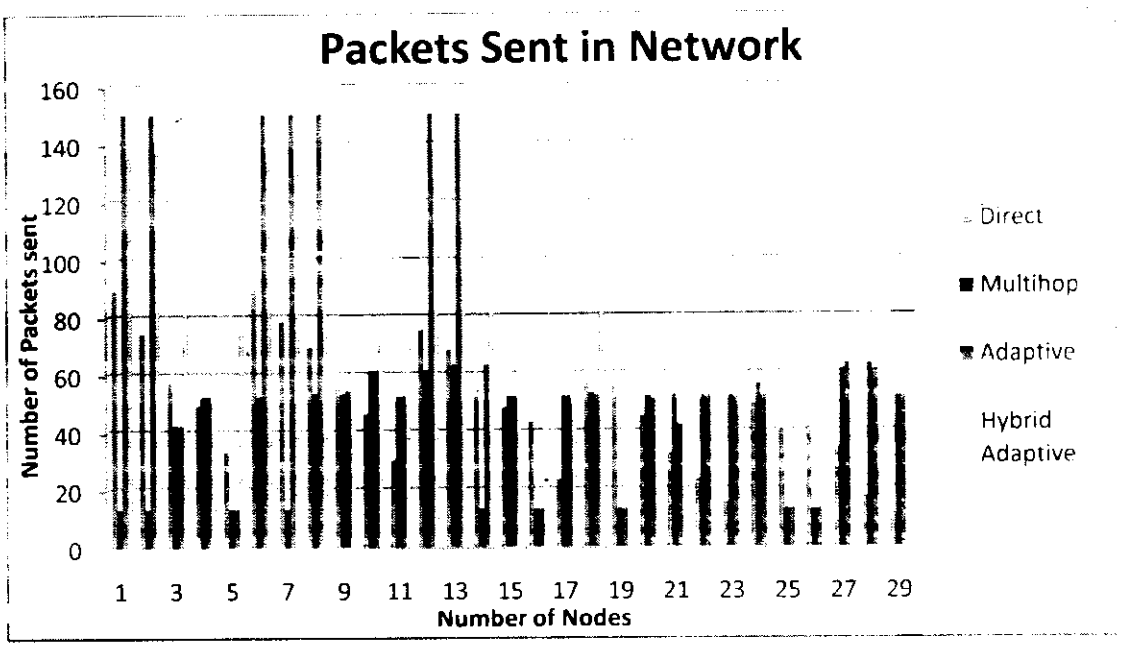


Figure 6-1: Packets Sent in Network

Multiple tests have been made while considering different energy levels to get the overall performance of the algorithms. Graph below in Figure 6-2 shows a comparison of average number of packet sent in the network of having energy levels of 300, 600, 900 and 1200. Graph clearly shows that packets sent to the cluster head are directly proportional to the energy level. But comparison with other techniques it is quite clear that our suggested technique has capacity to send more packets of data.

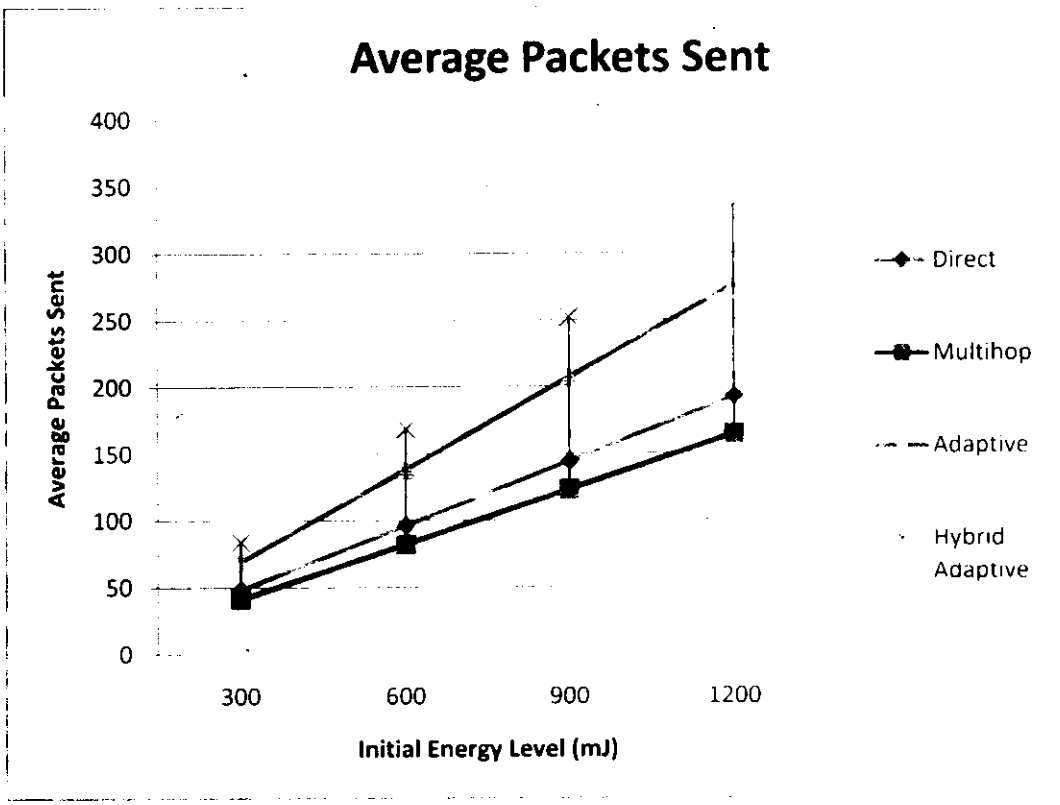


Figure 6-2: Average Packet Sent in the Network

Figure 6-3 shows the level of energy left at each node. This is the comparison of all the four routing techniques. At time t_n after the simulation starts, this energy level was observed.

Graph indicates that using proposed mechanism nodes have more power left with them than the others. Upon the deeply analysis it is noticed that nodes close to the Cluster Head have more energy, as at these nodes less energy is required to send data to cluster head. But the nodes far from cluster head have less energy as they have to spent more energy to send data packets. By the graph it is quite clear that accumulatively "Hybrid Adaptive Routing" consumes less energy. So this mechanism leads to an energy efficient routing for Wireless Sensor Networks.

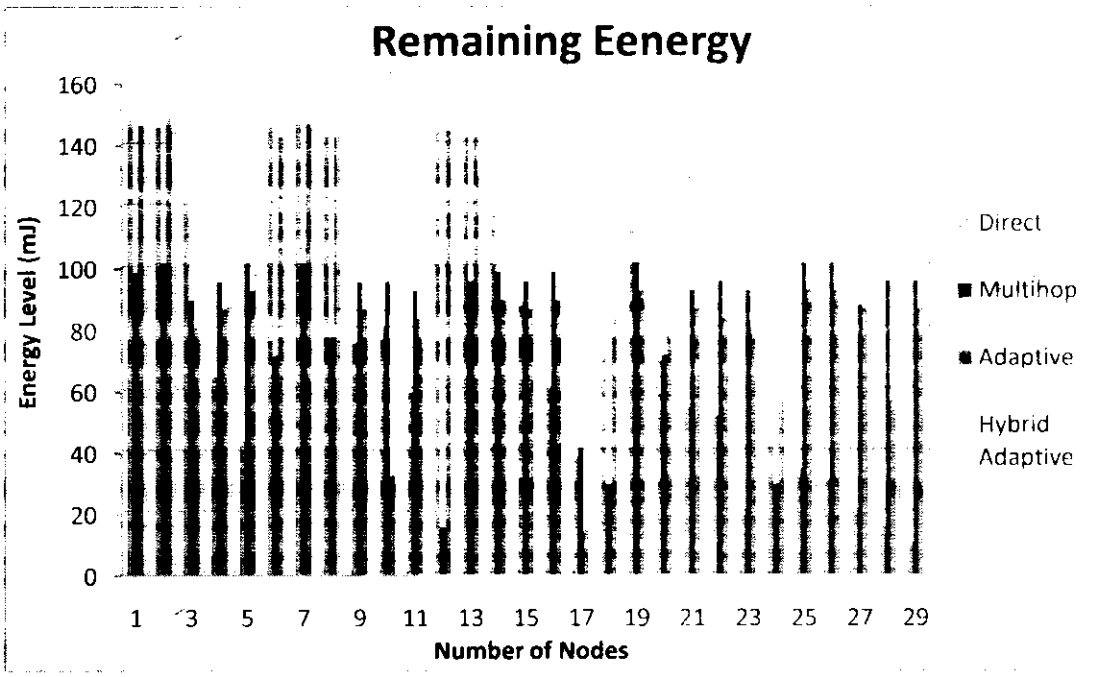


Figure 6-3: Remaining Energy of Network at Time t_n

Graph in Figure 6-4 indicated the total energy consumed by the nodes at any time t_n. Hence from the graph it is depicted that our proposed mechanism is more energy efficient as compared to other routing techniques i.e. Adaptive, Multihop and Direct routing.

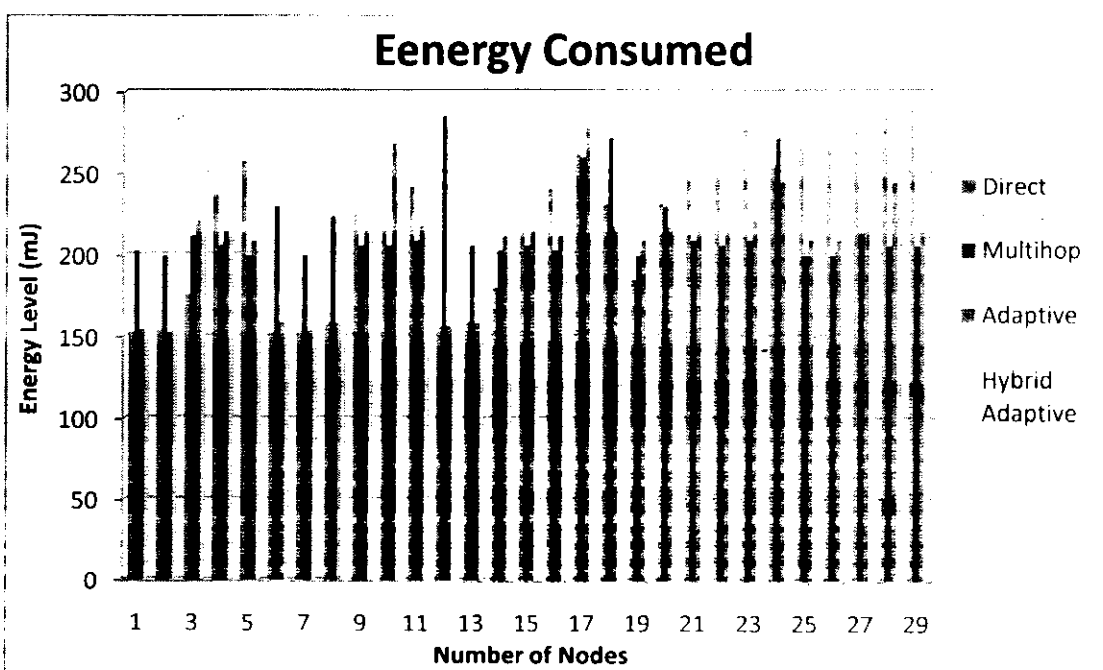


Figure 6-4: Energy Consumed at Time t_n

Graph in Figure 6-5 shows the total number of packets sent in the network. Comparison shows that our proposed mechanism "*Hybrid Adaptive Routing*" is competent enough to send more packets towards the Cluster Head. If we statically compare all the techniques then "Hybrid Adaptive routing" sent 35% more packets towards CH than the "Direct routing". 42% more packets than "Multihop routing" and 14% more packets than "Adaptive routing". If energy remained with nodes is high so that nodes can send more data towards Cluster Head. This is the proof of our proposed solution that Hybrid Adaptive routing technique is more energy efficient although having same level energy as others.

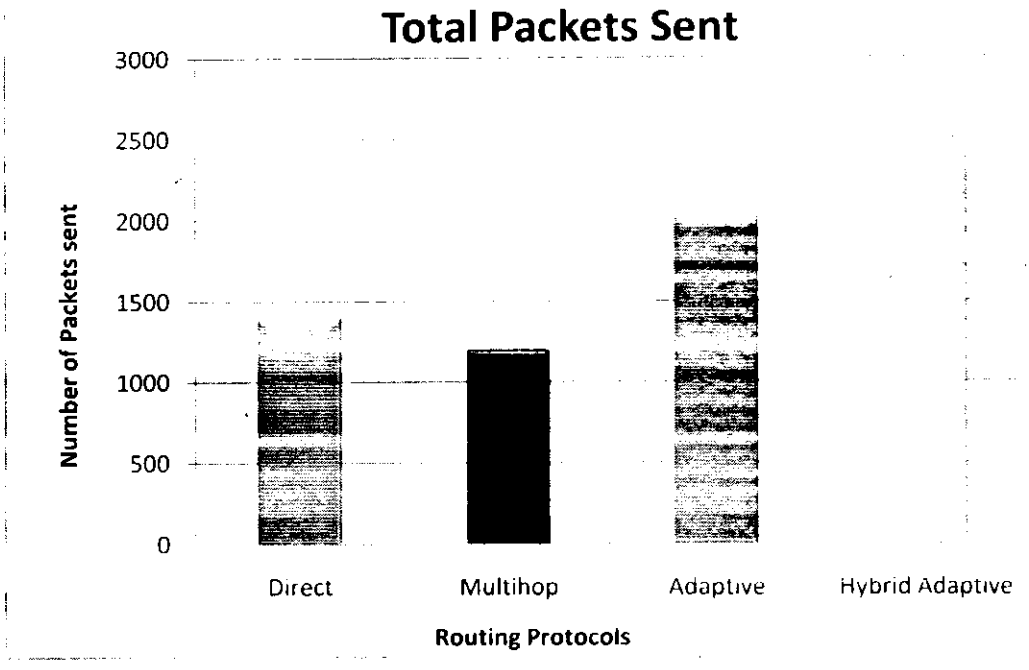


Figure 6-5: Total no. of Packet Sent in the Network

Network lifetime of the Wireless Sensor Network is core objective of our research effort and there is a prominent growth in it by the proposed technique.

Lifetime of a network is a time from the start of network till the death of first node in that network. Graph Figure 6-6 indicates the evaluation of network lifetime in all the techniques. Death of nodes occurring is quite delayed by using "*Hybrid Adaptive*" as compared by using Direct, Multihop and Adaptive. So there is a remarkable increase in the network lifetime. This was the core objective to start this research work so this is most major achievement of our proposed algorithm.

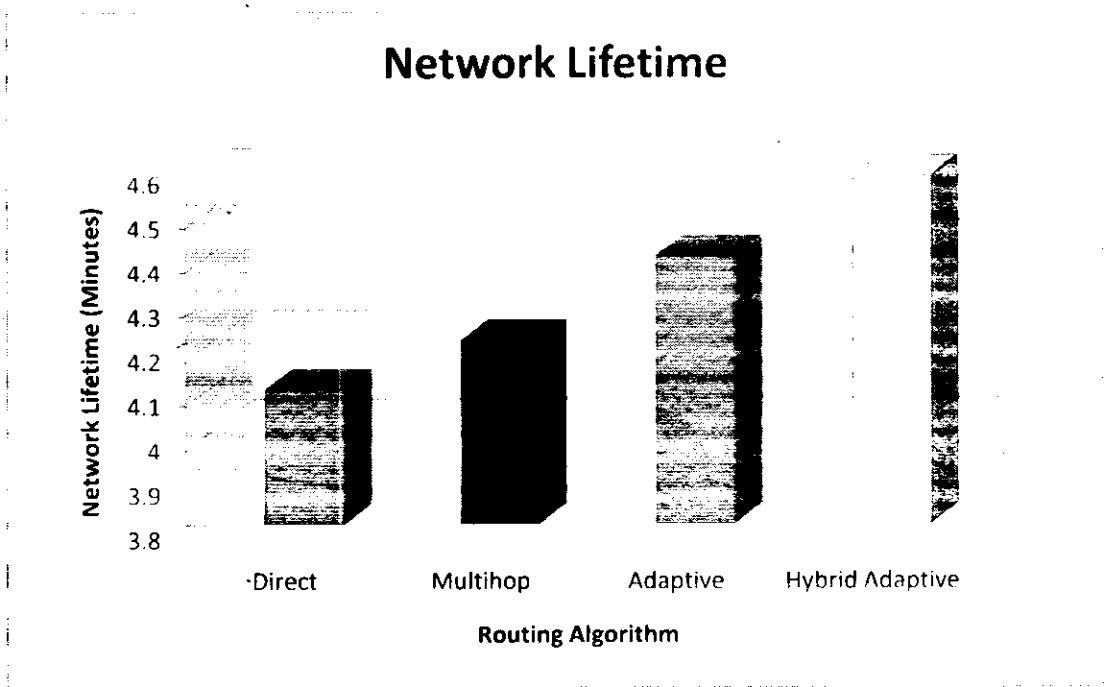


Figure 6-6: Network Lifetime

Experiment was not single in number, but multiple experiments have been performed by taking in account of different simulation parameters. We have compared the network lifetime of network by considering different energy levels in the start of simulation. Results are shown in Figure 6-7. Graph clearly shows if we double the energy level, double will be the difference in network lifetime between "Hybrid Adaptive routing" and other routing techniques. Proposed technique is the top ranked in achieving higher lifetime than the others.

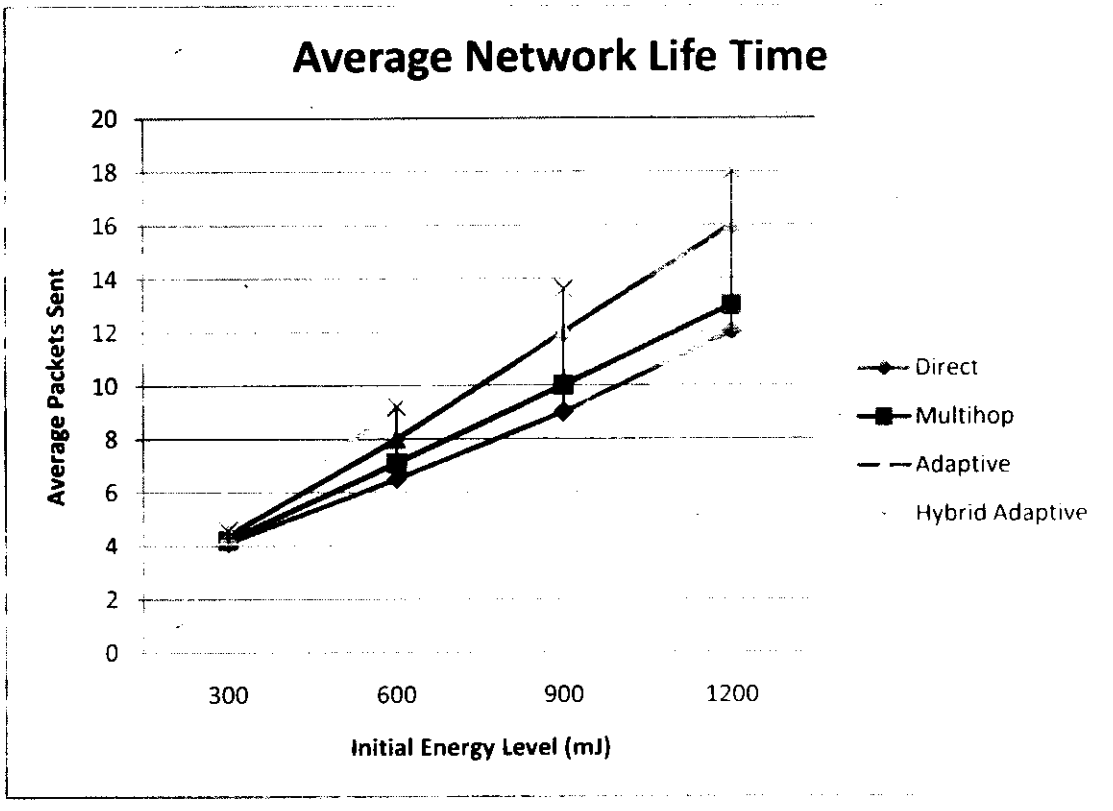


Figure 6-7: Lifetime of Network

Results produced by our technique shows that required objectives have been achieved.

7

CONCLUSIONS AND FUTURE WORK

7. CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This research work is a new technique for routing among the nodes of cluster. This is fully adaptive, energy efficient, hybrid, scalable, dynamic, etc. it enhances the scalability of WSN and prolongs the lifetime of network.

In our solution Direct routing is used along with the MultiHop routing at a time. A node on the basis of its current energy level decides whether to use Direct routing or MultiHop routing. The adaptive nature of the solution enables the node to choose one of these two routing strategies. The Hybrid Adaptive routing algorithm sends more packets as compared to the available three techniques. So decreased the overall energy consumption and increased the network life.

7.2 Future Work

TOP is always vacant, it's very rare when there become an end in research of any field, unless some new or more attractive filed is not introduced which grasp the features of parent or child field. There are much more things to be explored in this area. Like Data Communication is full-fledge challenge for researchers to make this technology more efficient and trustworthy. Even the utility of proposed solution can be explored in different dimensions that include:

7.2.1 Real Environmental Testing

We used TOSSIM simulator which can't calculate the impact of real environmental factors. This research can be tried on actual motes to get the doubt free and mature results.

7.2.2 Efficient Inter Cluster routing

We have applied this research work in intra cluster routing and showing best results but it can be tested for inter cluster routing as well, which can have few more issues in communication.

7.2.3 More Efficient Intra Cluster routing

This technique is showing best results for communication of data to the cluster head. but this work can be extended to more generalized level of routing. i.e.. In case of Multihop routing what should be the best route for data communication. what will be the more appropriate next hop.

7.2.4 Infrastructure of Network

We assumed the nodes at a fix distance of 10 feet from each other and are in specific direction as layout topology is Grid having same physical environment. But this can be checked on nodes at different distances and at different directions. As in real life in some case it may not be possible to place nodes at exact distance in an uneven area.

7.2.5 Size of Network

Currently we have tested the proposed technique on a cluster of 30 sensor nodes only. It can be tested on the number of nodes double or triple to it. because real life networks may require dense deployment.

7.2.6 Security

Data security hazard. data can be sniffed at the intermediate nodes during Multihop routing. There should a secure mechanism that data reached from source to destination without any theft.

REFERENCES

1. National Instruments ("NI"). Tutorial on "Wireless Sensor Network", Jul 23, 2009. http://zone.ni.com/devzone/cda/tut/p/id/8707 [accessed 23-Dec-10]
2. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D. E., and Pister, K. S. J. System architecture directions for networked sensors. *In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*. Cambridge, MA, 93–104, 2000.
3. Official website of Tiny Operating System. http://www.tinyos.net
4. Philip Levis, Nelson Lee, Matt Welsh, and David Culler. "*Tossim: Accurate and scalable simulation of entire tinyos applications*", in Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003). (http://webs.cs.berkeley.edu/users/nestfr/nestfr_paper_display.php). November 2003.
5. "TOSSIM: A Simulator for TinyOS Networks", Philip Levis and Nelson Lee. UC Berkeley September 17, 2003.
6. http://Sensor node - Wikipedia, the free encyclopedia.htm
7. Official website of Crossbow Technologies. http://www.xbow.com
8. Official website of Dust Networks. http://www.dust-inc.com
9. Telos Corporation. http://www.telos.com
10. http://www.howstuffworks.com/framed.htm?parent=motes.htm&url=http://www.cs.berkeley.edu/~jhill/spec/index.htm
11. Eric Brewer (UC Berkeley), David Culler (UC Berkeley), David Gay (Intel Research), Phil Levis (UC Berkeley), Rob von Behren (UC Berkeley), Matt Welsh (Harvard University). *nesC: A Programming Language for Deeply Networked Systems*, [http://nesec.sourceforge.net]. December 2004
12. Stefan Dulman and Paul Havinga, "*Operating system fundamentals for the eyes distributed sensor networks*", University of Twente, Department of

Computer sciemce Enschede, the Netherlands. [http://www.eyes.eu.org/publications/], October 2002.

13. Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. "*Contiki - a lightweight and flexible operating system for tiny networked sensors*", 29th Annual IEEE International Conference on Local Computer Networks, (LCN'04), 2004.

14. The Operating System for the Internet of Things http://www.sics.se/contiki/

15. Chih-Chieh Han, Ram Kumar, Roy Shea, Eddie Kohler, and Mani Srivastava," *A dynamic operating system for sensor nodes*" in MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services, ACM Press, pages 163–176, New York, USA, 2005.

16. Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, "*Energy-Efficient Communication Protocol for Wireless Micro sensor Networks*", in Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), volume 2, page 10, Maui, Hawaii, USA, January 2000.

17. Seema Bandyopadhyay and Edward J. Coyle, "*An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks*", in IEEE INFOCOM, 2003.

18. Jamil Ibriq and Imad Mahgoub, "*Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges*" In Proceedings of the 2004 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04), pages 759–766, San Jose, CA, USA, July 2004.

19. Ming Ma and Yuanyuan Yang, "*Clustering and Load Balancing in Hybrid Sensor Networks with Mobile Cluster Heads*", ACM Third International Conference in Quality Of Service In Heterogeneous Wired/Wireless Networks, Department of Electrical and Computer Engineering State University of New York, Stony Brook, NY 11794, USA, 2006.

20. M. Goyeneche, J. Villadangos, J.J. Astrain, M. Prieto, A. Cordoba, "*A Distributed Data Gathering Algorithm for Wireless Sensor Networks with*

Uniform Architecture", in PE-WASUN'06, Torremolinos, Malaga, Spain. October 6, 2006.

21. Nauman Israr and Irfan Awan, "*Multihop clustering algorithm for load balancing in wireless sensor networks*" I. J. of SIMULATION. Vol. 8 No. 1. Pages 13-25, Mobile Computing Networks and Security Research group School of Informatics, University of Bradford, UK, July 2006.

22. Anh Tuan Hoang and Mehul Motani, "*Collaborative Broadcasting and Compression in Cluster-Based Wireless Sensor Networks*", in ACM Transactions on Sensor Networks, Vol. 3, No. 3, Article 17, August 2007.

23. Li Li, Dong Shu-song, Wen Xiang-Ming, "*An energy efficient clustering routing algorithm for wireless sensor networks*", in the journal of china universities of posts and telecommunications, Volume 13, Issue 3, September 2006.

24. Tao Wu and Subir Biswas, "*Reducing Inter-cluster TDMA Interference by Adaptive MAC Allocation in Sensor Networks*", in Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Pages 507 – 511, 2005.

25. Irfan Ahmed, Mugen Peng, Prof. Wenbo Wang, "*A Unified Energy Efficient Cluster ID based Routing Scheme for Wireless Sensor Networks – A more Realistic Analysis*", in Third International Conference on Networking and Services, Pages 86, 2007.

26. Adeel Akhtar, "*Adaptive intra cluster routing for wireless sensor networks*", MS Thesis, Islamic International University, Islamabad, 2009.

27. Chiu-Kuo Liang; Jian-Da Lin; Chih-Shiuan Li; "Steiner Points Routing Protocol for Wireless Sensor Networks," *Future Information Technology (FutureTech), 2010 5th International Conference on* , vol., no., pp.1-5, 21-23 May 2010.
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5482701&isnumber=5482630

28. Yan Sun; Haiqin Liu; Min Sik Kim; , "Energy-Efficient Routing Protocol in Event-Driven Wireless Sensor Networks," *Communications Workshops*

(ICC). 2010 IEEE International Conference on , vol., no., pp.1-5, 23-27 May 2010.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp &arnumber=5503872&isnumber=5503872

29. Shancang Li; Xinheng Wang; Zongbin Li; , "A novel scalable routing scheme based on Polychromatic Sets theory for wireless sensor networks," *GLOBECOM Workshops (GC Wkshps), 2010 IEEE* , vol., no., pp.99-103, 6-10 Dec. 2010

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp &arnumber=5700476&isnumber=5700081

30. Guowei Wu; Heqian Li; Lin Yao; , "A Group-Based Mobile Agent Routing Protocol for Multitype Wireless Sensor Networks," *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)* , vol., no., pp.42-49, 18-20 Dec. 2010

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp &arnumber=5724809&isnumber=5724803

31. *Microsoft ENCARTA World English Dictionary*

32. "The Network Simulator - NS2," http://www.isi.edu/nsnam/ns/.

33. Lokesh Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. "GloMoSim: A Scalable Network Simulation Environment", UCLA Computer Science Department Technical Report 990027, May 1999.

34. Jeremy Elson, Lewis Girod, and Deborah Estrin. "Emstar: Development with high system visibility", in Wireless Communications, pages 70 – 77, December 2004.

35. "SensorSim"[http://nesl.ee.ucla.edu/projects/sensorsim/]

36. http://www.scalablenetworks.com/.

37. http://www.opnet.com/

38. http://www.cs.rpi.edu/~cheng3/sense/.