STUDY AND IMPLEMENTATION OF CROSS LAYER PROTOCOL IN WIRELESS SENSOR NETWORKS

BY

NITISH KUMAR MATHUR 101285

PROJECT SUPERVISOR: DR YASHWANT SINGH

MAY -2014

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

Table of Contents

CERTIFICATE

This is to certify that the work titled "**STUDY AND IMPLEMENTATION OF CROSS LAYER PROTOCOL IN WIRELESS SENSOR NETWORK** " Submitted by **NITISH KUMAR MATHUR** in partial fulfillment for the award of degree of B.Tech Computer Science and Engineering of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Student:

Name of Student: Nitish Kumar Mathur

Enrollment No: 101285

Date: 15th May 2014

Signature of Supervisor:

Name of Supervisor: Dr. Yashwant Singh

Designation: Asst. Professor, Dept. of CSE and ICT

Date: 15th May 2014

ACKNOWLEDGEMENT

We owe a great thanks to many people who have been helping and supporting us during this project. Our heartiest thanks to **Dr Yashwant Singh**, the Project Guide for guiding and correcting us at every step of our work with attention and care. He has taken pain to go through the project and make necessary correction as and when needed. Thanks and appreciation to the helpful people at college for their support. We would also thank our university and my faculty members without whom this project would have been a distant reality. We also extend our heartfelt thanks to our family and well-wishers.

Signature of Student:

Name of Student: Nitish Kumar Mathur (101285)

Date: 15th May 2014

ABSTRACT

The rapid evolution of wireless network technology and the explosive growth of wireless network services have made wireless communication an ubiquitous means for transporting information across many different domains. However, there are still some fundamental challenges that need to be overcome in the design of the next generation of wireless sensor networks. The sensor nodes present in the wireless sensor networks are constrained of energy as they are powered with the help of battery. Due to energy limitations there is a great need of providing any energy efficient way of communication for the wireless sensor networks.

The traditional method of designing networks using the layered approach might be unsuitable and inefficient for designing wireless networks. Cross layer design is an emerging methodology, which is been implemented in this project to provide congestion control, better routing over the cross layers. This cross layer protocol is designed based on the initiative determination present in cross layer module .The implementation of this initiative determination involves the comparison with the threshold values. The central idea of cross layer design is to optimize the control and exchange of information over two and more layers to achieve significant performance improvements by exploiting the interactions between various protocol layers.

List of Figures

Chapter 1: INTRODUCTION

1.1 Introduction

The wireless sensor networks which have gained huge importance these days are actually event based networks which continuously sense the surroundings in order to collect the required information .The main aim of this network is to extract the event features based on the information that is collected by the sensor nodes with less energy and storage abilities. On the other hand huge research has been taking place to develop several networking protocols in order to provide better communication in energy efficient manner.

In traditional communication networks, the Open System Interconnection (OSI) layered architecture has been widely adopted and has been served many communications systems well in the past; however, evolving wireless networks of today are seriously challenging this design philosophy. The layered architecture defines a stack of protocol layers in which each layer operate within its well defined function and boundary and thus allowing changes to the underlying technology at each layer without imposing the need to change the overall system architecture. This approach has been successful in its ability to provide modularity, transparency and standardization in the wire line networks but might be unsuitable in the wireless networks domain.

Even though these protocols can offer higher performance matrix with respect to that particular layer, they are not better to improve the overall performance of the network and also to minimize the energy consumption. Hence it can be stated that there is a need of an efficient way of unifying the functions of all protocol layers so as to offer efficient communication between the nodes of a wireless sensor network. Cross layer design that combines all the layers is considered as the best solution for the wireless sensor networks. Research also specifies that the protocol design based on this cross layer approach can provide an energy efficient way of communication in a wireless sensor network.

In this project a concept of initiative determination is considered and it is applied for the cross layer operations in order to provide congestion control, distributed routing, medium access control etc. Using this initiative determination method any node present in the wireless sensor network can decide whether to take part in the communication depending on several aspects like energy level, link quality, buffer level, location and traffic load. All these parameters are integrated and provided in a particular decision that a node has to provide whether is willing to participant in a particular communication. Based on this concept cross layer protocol is implemented in this project to provide a better communication between the sensor nodes of a wireless sensor networks in an energy efficient manner. The cross layer design jointly optimize the control and exchange of information over two or more layers and significant performance improvements can be exploiting the interactions between various layers of the protocol stack . However, the drawback to such a design is the potential to destroy modularity, and thus making the overall system fragile. The study of cross layer design for wireless networks is an interesting area and the same will be implemented in this project.

1.2 Problem Statement & Motivation

The wireless sensor networks are regarded as a collection of various sensor nodes which are aimed in collecting information from the surrounding environment. A traditional layered protocol is being used in the existing system for providing communications in sensor networks. This existing system suffers from several limitations related to overhead and the congestion of traffic. So I am developing a framework for the study of cross layer design and to explore the various areas where performance gains can be achieved in the context of WSNs.

Hence I am highly motivated to this topic because currently and in future there is a need of an efficient way for providing communications between the nodes of a wireless sensor network. The study of cross layer protocol is a relatively new research area and this design in wireless networks is strongly recommended where the characteristics of the wireless channel permeate the functions of all protocol layers in the traditional protocol stack.

1.3 Objectives

The main aim of the project is to study and implement a reliable and efficient communication method for the wireless sensor networks through cross layer protocol (XLP).

1.4 Approach

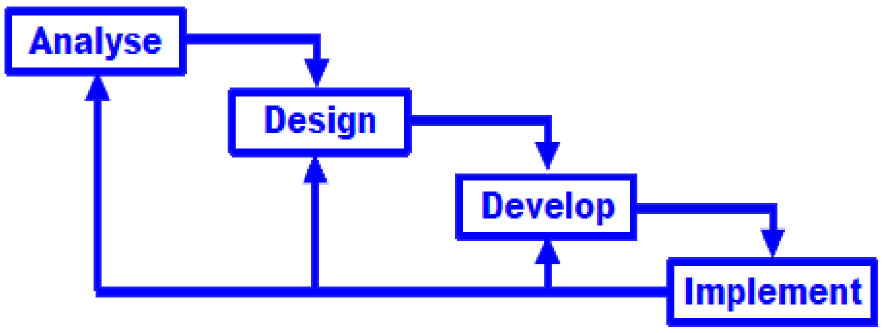


Figure 1.1 – Iterative Model

Advantages

- Efficient show & throw model
- Works even if customer needs are unclear
- Once a phase is completed, coming back to that phase for recheck is permitted

Criticisms

- Not a disciplined approach
- For every change in requirement, a new iteration is required

1.5 Organization of the Report

Chapter 1: Introduction

Covers the various project related things such as, Introduction, Problem statement , Motivation and tells us about the objectives behind the choice of this project.

Chapter 2: Literature Survey

Covers the literature surveyed as well as tells about the concepts that have been studied and understood.

Chapter 3: Design and Implementation

It tells us about the System Design - various design diagrams. Also, it covers the implementation of the project and the Project snapshots.

Chapter 4: Conclusion & Future work

In this we have concluded our given project and the work which we have plan for next semester.

Chapter 2: LITERATURE REVIEW

2.1 What is Wireless Sensor Networks?

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

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. These devices can be deployed under various conditions or at different locations and will collaborate with each other in order to accomplish a pre-defined task, such as environmental monitoring or battlefield surveillance. WSNs are usually characterized by dense node deployment, unreliable sensor nodes, frequent topology changes and severe power, computation and memory constraints. These unique characteristics and constraints present many challenges and obstacles in the design and implementation of WSNs. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding. In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year.

It is to be noted that mainly due to these nodes are available in the low-cost, the deployment can be successful in order of perfect magnitude of thousands to ten million nodes in a given situation or environment. The said nodes can be successfully deployed either in a preengineered way or in random fashion way. Over a wireless channel, the sensor nodes performs successfully as per the desired measurements, perfectly process the measured information or data and then correctly transmit it a given base station, usually referred to as the sink node. The base station gathers the information or data from all the given nodes and then successfully analyzes this given information or data in order to draw the desired conclusion regarding the functions and activities in the given area of interest. It is to be remembered that sinks can also acts as a perfect gateways to other given networks, access points for human interface or a powerful data process. They are frequently utilized in order to extract information or data from the network or to disseminate control information or data.

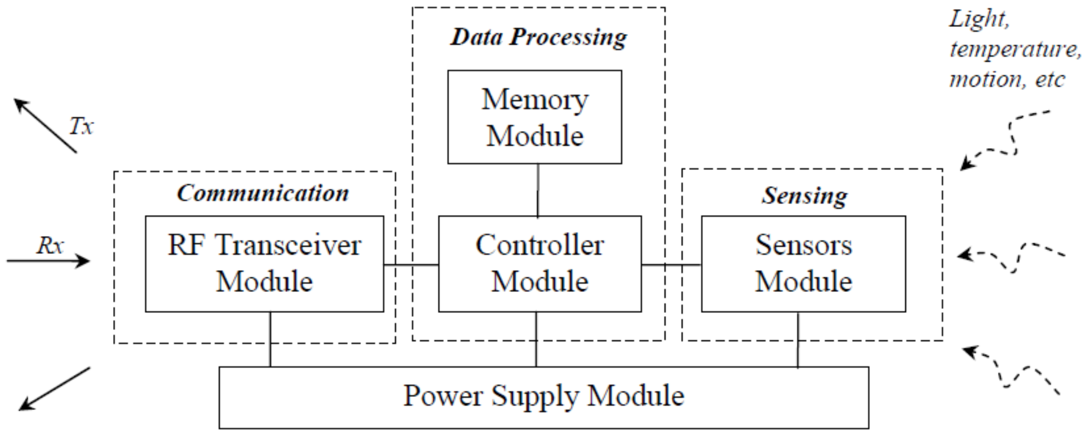


Figure 2.1 – Overview of a Wireless Sensor Node

The sensor module contains a transducer that performs the sensing operation of the physical environment. The sampled data is processed by the controller where it is stored in memory. The RF transceiver module is typically a short range radio communication device that performs both data transmission and reception. The power supply module is usually some form of power source such as batteries that provide the energy to power the other modules. All the four modules are inter-related to each other and their functions are dependent on the role of the sensor node.

2.1.1 Characteristics of WSN

The main characteristics of a WSN include:

- Power Consumption Constraint

The sensor nodes present in the wireless sensor networks are constrained of energy as they are powered with the help of battery. Deployment of the sensor nodes in the hostile environment makes it unfavorable for the people to change the battery of the senor nodes when it is expired.

- Large Scale Coordination

The sensors need to coordinate with each other to produce required results.

- Utilization of Sensors

The sensors should be utilized in a ways that produce the maximum performance and use less energy.

- Real Time Computation

The computation should be done quickly as new data is always being generated.

- Scalability to large scale of deployment

Support large number of nodes.

- Maintainability

WSN has to adapt to changes, self- monitoring, adapt operation , incorporate possible additional resources e g newly deployed nodes.

- Programmability

Re-programming of nodes in the field might be necessary, improve flexibility.

2.1.2 Design Challenges of WSN

- Reliability

For any WSN, the key goal is to have reliable data transmission, i.e., the highest probability of a packet being delivered in the presence of interference. Reliability refers to the packet error rate, error-correction coding, retransmission mechanisms and frequency planning. In the seven-layer OSI communication model, only the bottom four layers matter as far as reliability is concerned. Network topologies also affect the reliability of the communication system.

- Security

Data protection in a WSN is a major challenge. Data security be-comes even more significant when any security-oriented application is considered. Not only must the system ensure privacy, it must be able to authenticate data communication also. Security includes encryption and cryptographic authentication, which impose major challenges in terms of the cost of power and bandwidth.

- Performance

Performance encompasses bandwidth, data rates, latency and packet prioritization (QoS) techniques. In fact, transmission time has a major impact on performance.

- Cost- Effectiveness

With the advances in semiconductor technology, WSN embedded products have become cheaper, especially for single-chip solutions that comprise microcontrollers, transceivers and on-chip memory. A key factor in the cost of wireless embedded processors is the amount of memory required. Developers of sensor networks will expect access to a range of chips or wireless microprocessors/microcontrollers with optimized memory footprints to meet the needs of a range of applications.

- Flexibility

WSN architecture must be flexible and adaptive enough to deploy in a wide range of usage scenarios. The number of sensors in different deployments may vary from thousands to millions. This will have a significant impact on network maintenance and diagnostics. Therefore network management must be flexible enough to sustain this huge network of sensors.

- Fault tolerance ,Redundancy and Robustness

Node loss is an important issue to consider in a WSN. Since sensors can be deployed everywhere, e.g., on bridges, industrial sites, homes, farms, etc, these will be subjected to harsh environmental conditions and may be vandalized, stolen or damaged. Additionally, power surges, blackouts or brownouts can cause nodes to fail. So it’s important to configure and deploy a network that can isolate individual points of failure, eliminate or mitigate their impact, and continue to operate as a whole, maintaining very high end-to-end fault tolerance and robustness against nodes failure.

- Self-Organizing and self-healing

A WSN should provide extremely high reliability and predictability for years on end by applying self-organizing and self-healing intelligence to continuously adapt to unpredictable conditions. There are some applications where sensors are deployed sporadically in a geographical location and these have to organize themselves and find out where they are in relation to each other. The exact structure of a WSN cannot be predetermined in most situations due to variable dependencies such as location of devices and radio propagation during network formation. In addition to joining a network, devices could leave, be forced to leave or potentially re-associate elsewhere within the network all of which could cause a change in network structure.

- Routing

In addition to managing network joining, de-association and re-association, the network formation process requires various routing methodologies. Traditional routing schemes are not designed to be energy or processing efficient and are of little use in a sensor network .In that case ,it is best to keep the routing to a minimum and activate only when absolutely necessary .

- Transmission time

An issue some-times overlooked is the length of time it takes for transmission of data packets. In fact, transmission time has a major influence on performance, quality, power consumption and interference. So reducing it is a major design challenge.

2.1.3 Quality of Service (QoS) parameters in WSN

Quality of Service (QoS) in general largely refers to perfect quality as perceived by the application and the user. It is to be highly noted that from various perspectives, Qos assistance in WSNs is an open area of research studies. On the other hand, QoS is largely understood by several technical communities through different manners. Among the community of networking, QoS is well understood as a measure or step of service quality which said network provides to the given application or to the end user. According to the experts in this particular field, QoS has been perfectly well-defined as a unit of different types of service requirements in order to be fulfilled when successfully transmitting a given stream of packets from the said source to the given destination. It is to be highly noted that QoS cites certain parameters in the traditional data network like bandwidth, jitter, delay, packet loss, etc. The other requirements of QoS in WSNs like network lifetime, fault tolerance, coverage, aggregation delay, data accuracy, etc. On the other hand, the said network is totally different from the given traditional end-to-end QoS immediate requirements mainly due to the differences in network properties and in the application domains.

QoS solutions are well-developed for the said traditional networks; these can be ported successfully in WSNs mainly due to several reasons such as data-centric and application specific communication protocols in wireless sensor networks, random and large scale deployment of sensor nodes and constraints of severe resource in sensor nodes. It is to be highly noted that researchers and experts in this particular field are working with committed and dedication to find means for QoS perfect support in wireless sensor networks. The most important kinds of approaches for QoS support in WSNs are Middleware layer based QoS support; Cross Layer based QoS support and Network Layer that is largely based on QoS support in terms of routing.

2.1.4 Importance of better communication method in WSN

Wireless sensor networks play an essential role in the modern methods of communication. It is one of the most better and efficient technique or method of perfect communication in any given situation or environment to a large extent. In this fast paced life style of personal and professional situation or environment, WSNs plays a significant role in the perfect development and advancement of better communication standards. It is to be highly noted that all the tools and implemented techniques should perfectly co-ordinate and co-operate for the exchange of all types of information or data. In case if any of the tools and implemented techniques doesn't function properly or are inactive in a given situation or environment then the exchange of information will be hampered to a large extent and may cause several difficulties of temporary or permanent nature to a great extent. There are many situations in the personal and professional life that WSNs are used for several reasons, if there isn't any better communication method adopted in WSNs then it will prove to be a total failure for the exchange of information or data to large extent.

For example, WSNs is utilized in the military department usually for their daily administration and control and largely utilized in serious situation in the war front or battle fields. A lot of confidential or secretive information or data are exchanged among the troops and between the base stations and troops on the ground, air or water in the entire given situation or environment. Better communication is part and parcel of all types of activities and function in the military department. There is a large need to improve and advance the tools and techniques implemented in the methods of better communication in WSNs. There is also a greater need to find the loopholes and negative aspects in the better communication in WSNs so that steps or measures can be taken to improve or advance the areas that really need to be looked upon for improvement and advancement with perfection. On the other hand, WSNs is utilized in various other fields like healthcare, civil administration, media and entertainment sector etc. A better communication in WSNs is the need of the hour because exchange of information or data is directly or indirectly depended on WSNs. Several functions and activities also directly or indirectly are depended on the better communication in WSNs that is perfectly implemented. There is an immediate need for finding the best means to be perfectly implemented in WSN for better and better communication methods or techniques. Many departments of the government sectors, private sectors and commercial sectors are actively utilizing WSN in all the functions and activities within their given sector for better exchange of information or data that are confidential or secretive in nature.

2.2 Cross Layer Protocol

Cross-layer protocol is an escape from the pure waterfall-like concept of the OSI communications model with virtually strict boundaries between layers. The cross layer approach transports feedback dynamically via the layer boundaries to enable the compensation for e.g. overload, latency or other mismatch of requirements and resources by any control input to another layer but that layer directly affected by the detected deficiency.

A Cross layer protocol is the initial protocol which perfectly integrates all type of functionalities of all the given layers from transport to physical into a cross-layer protocol. Cross-layer design can help to exploit the interactions between layers and promotes adaptability at various layers based on information exchange. In the original OSI networking model, strict boundaries between layers are enforced, where data are kept strictly within a given layer. Cross-layer design removes such strict boundaries to allow communication between layers by permitting one layer to access the data of another layer to exchange information and enable interaction.

2.2.1 Initiative determination

The initiative determination concept coupled with the receiver-based contention mechanism provides freedom to each node participating in communication. In WSNs, the major goal of a communication suite is to successfully transport event information by constructing (possibly) multihop paths to the sink. To this end, the cross-layer initiative determination concept constitutes the core of the XLP and implicitly incorporates the intrinsic communication functionalities required for successful communication in WSNs.

Consider a node, i, which initiates transmission by informing its neighbors that it has a packet to send. This is achieved by broadcasting a request to send (RTS) packet. Upon receiving this packet, each neighbor of node i decides to participate in the communication or not. This decision is made through the initiative determination based on the current state of the node. The initiative determination is a binary operation where a node decides to participate in communication if its initiative is 1.

$$\mathcal{I} = \begin{cases} 1, & \text{if} \begin{cases} \xi_{RTS} \geq \xi_{Th} \\ \lambda_{relay} \leq \lambda_{relay}^{Th} \\ \beta \leq \beta^{max} \\ E_{rem} \geq E_{rem}^{min}, \end{cases} \\ 0, & \text{otherwise}. \end{cases} \qquad (1)$$

Figure 2.2 – Conditions of Initiative Determination

The initiative is set to 1 if all four conditions in (1) are satisfied, where each condition constitutes a certain communication functionality in XLP. The first condition ensures reliable links to be constructed for communication based on the current channel conditions. For this purpose , it is required that the received SNR of an RTS packet is above some threshold for a node to participate in communication .The second and third conditions are used for local congestion control in XLP. The second condition prevents congestion by limiting the traffic a node can relay. More specifically, a node participates in the communication if its relay input rate is below some threshold. The third condition ensures that the buffer occupancy level of a node does not exceed a specific threshold so that the node does not experience buffer overflow and the congestion is prevented. The last condition ensures that the remaining energy of a node stays above a minimum value .This constraint helps preserve uniform distribution of energy consumption throughout the network.

The cross layer functionalities of XLP lie in these contraints that define the initiative of a node to participate in communication. Using the initiative concept ,XLP performs receiver based contention, initiative based forwarding ,local congestion control ,hop-by-hop reliability, and distributed operation.

2.2.2 Receiver Contention

The receiver contention operation of XLP leverages the initiative determination concept with the receiver based routing approach. After an RTS packet is received, if a node has an initiative to participate in the communication ,it performs receiver contention to forward the packet. The receiver contention is based on the routing level of each node, which is determined based on the progress a packet would make if a node forwards the packet. The feasible region is divided into Np priority regions.

Each priority region corresponds to a backoff window size CWi . Based on its location , a node backs off for $\sum_{j=1}^{i-1} CWj + cwi$,where cwi is randomly chosen such as cwi∈ $[0, CWmax]$, where CWmax=CWi-CWi-1.The winner of the contention sends a CTS packet to node i indicating that it will forward the packet. On the other hand , if during backoff, a potential receiver k receives a CTS packet, it determines that another potential receiver j with a longer progress has accepted to forward the packet and node k switches to sleep for the duration of the communication.When node i receives a CTS packet from a potential receiver, it determines that the receiver contention has ended and sends a DATA packet with the position of the winner node in the header. The CTS and DATA packets both inform the other contending nodes about the transmitter-receiver pair. Hence, other nodes stop contending and switch to sleep. In the case of two nodes sending CTS packets without hearing each other, the DATA packet sent by node i can resolve the contention. It may happen that multiple CTS packets from the same priority region can collide and a node from a lower priority region can be selected. XLP does not try to resolve this problem as this probability is very low since the contention region is already divided into multiple regions and the cost of trying to resolve this outweighs the gains.

Note that node I may not receive a CTS packet because of three reasons: 1)CTS packets collide,2)there exists no potential neighbors with initiative =1 or 3)there exists no nodes in the feasible region. Hence ,the neighbors of node I send a keep alive packet after $\sum_{j=1}^{Np} CWj + cw$ if no communication is overheard. However, if a keep alive packet is not received, the node continues retransmission in case there is a CTS packet collision. If no response is received after k retries, node i determines that a local minimum is reached and switches to an angle-based routing mode as explained next.

2.2.3 Advantages and Disadvantages

- **Advantages**

➢ Congestion Control

➢ Low Energy Consumption

➢ Better Routing

➢ Medium Access Control

- **Disadvantages**

➢ Simulation Errors

➢ Difficult to interpret the Simulation results

Chapter 3: DESIGN & IMPLEMENTATION

3.1 Design

3.1.1 UML Sequence Diagram

The sequence diagrams are an easy and intuitive way of describing the system's behavior, which focuses on the interaction between the system and the environment. This notational diagram shows the interaction arranged in a time sequence. The sequence diagram has two dimensions: the vertical dimension represents the time, the horizontal dimension represents different objects. The vertical line also called the object's *lifeline* represents the object's existence.

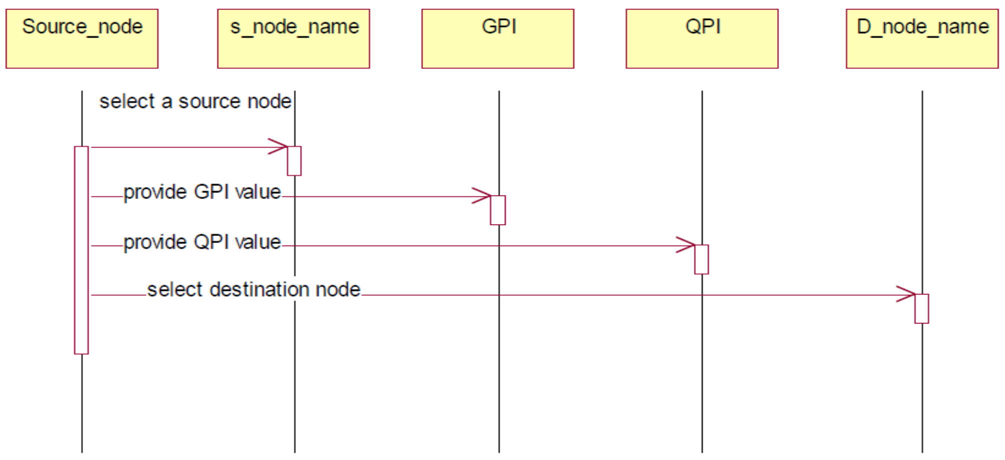


Figure 3.1 – Sequence Diagram of the Source node

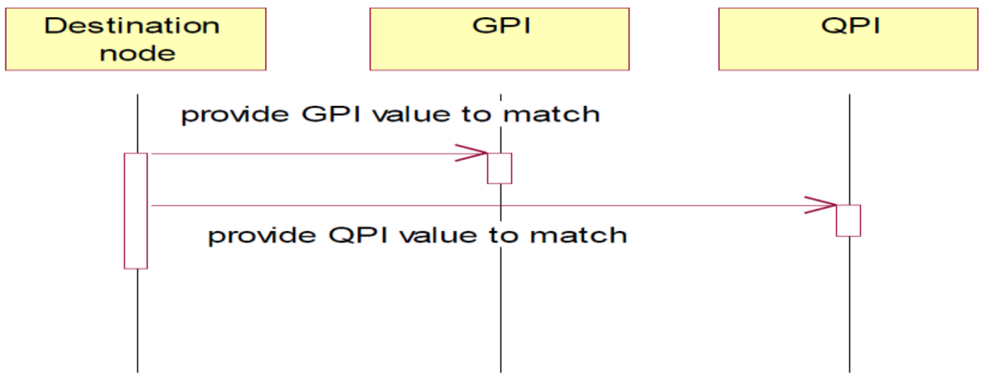


Figure 3.2 – Sequence Diagram of the Destination node

3.1.2 Activity Diagram

An activity diagram is characterized by states that denote various operations. Transition from one state to the other is triggered by completion of the operation. The purpose of an activity is symbolized by round box, comprising the name of the operation. An operation symbol indicates the execution of that operation. This activity diagram depicts the internal state of an object.

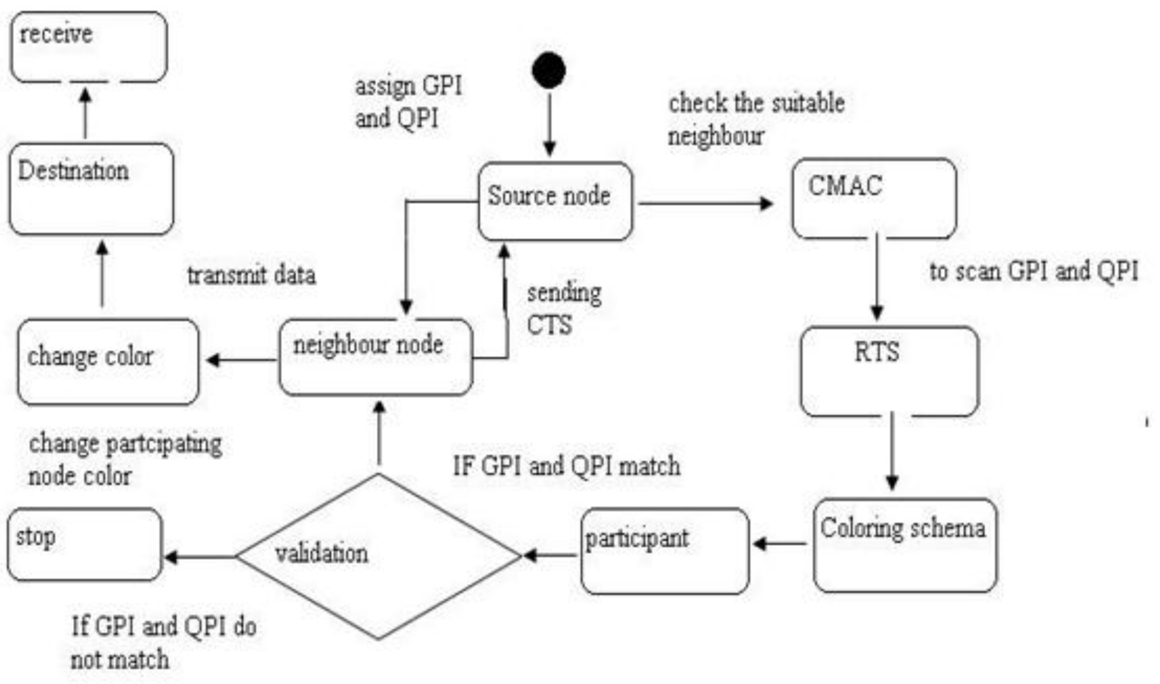


Figure 3.3 – Activity Diagram of the overall System

3.1.3 Use Case Diagram

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

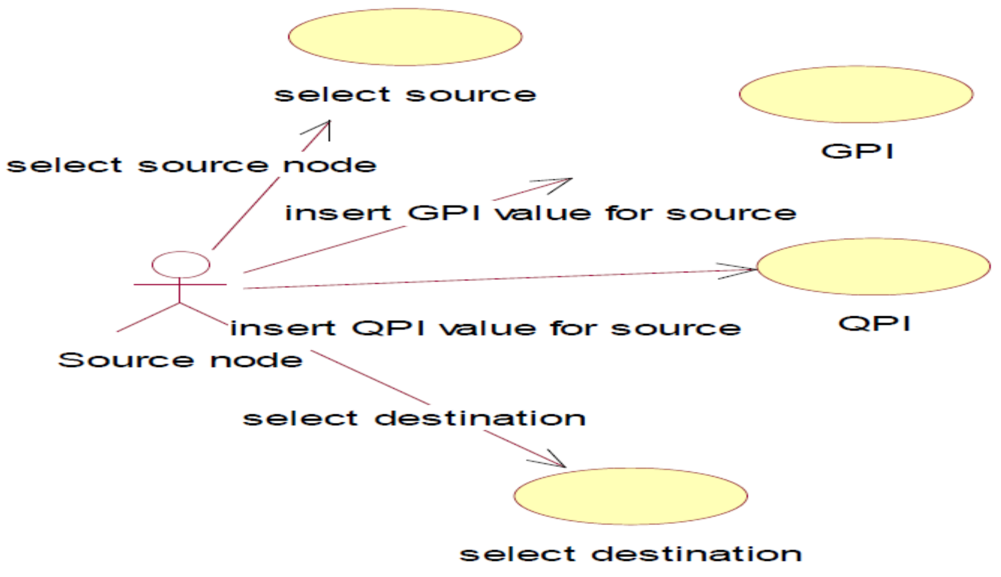


Figure 3.4 – Use Case Diagram of the Source Node

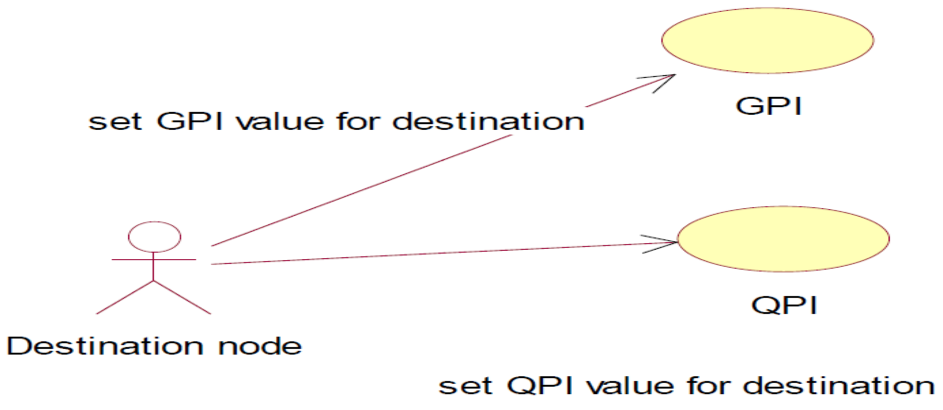


Figure 3.5 – Use Case Diagram of the Destination Node

3.2 Implementation

3.2.1 Proposed System

In the proposed system, a cross-layer protocol (XLP) is introduced, which achieves congestion control, routing, and medium access control in a cross-layer fashion. The design principle of XLP is based on the cross-layer concept of initiative determination, which enables receiver-based contention, initiative-based forwarding, local congestion control, and distributed duty cycle operation to realize efficient and reliable communication in WSNs. XLP is the first protocol that integrates functionalities of all layers from PHY to transport into a cross-layer protocol XLP significantly improves the communication performance and outperforms the traditional layered protocol architectures in terms of both network performance and implementation complexity.

3.2.2 Working

Initially the details of all the nodes are added in to the wireless sensor network .As you are entering a dialog box will appear that the data are successfully stored in the database .On clicking the next button an applet will be shown depicting the participated and non-participated nodes. On clicking the node button a priority regions are generated and finally a file is transferred from source to destination using a receiver contention or angle based machanism

3.2.3 System Requirements

- **Hardware**

• System	:	Pentium IV 2.4 GHz.
• Hard Disk	:	80GB.
• Ram	:	512 MB.

- **Software**

• Operating System	:	Windows XP, Windows 7.
• Environment	:	NetbeansIDE6.5
• Language	:	JAVA,SWING
• Database	:	MySql

3.2.4 Codes

AngleCommu.java

```
package xlp;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.PrintStream;
import java.sql.*;
import javax.swing.*;
import java.io.File;
 import javax.swing.JComboBox ;
import javax.swing.JFileChooser;

  class AngleCommu extends JFrame
    implements ActionListener
{

    public AngleCommu()
    {

        l = new JLabel("                  ANGLEBASED MECHANISM      ");
        l1 = new JLabel("LOCAL MINIMUM NODE");
        f1 = new JTextField();
                      l2 = new JLabel("NODE1 ANGLE");
        f2 = new JTextField();

                l3=  new JLabel(" NODE2");
        f3= new JTextField();
                     l4=  new JLabel("NODE2 ANGLE");
        f4= new JTextField();
                      l5 = new JLabel("NODE3 ");
        f5 = new JTextField(10);
                      l6 = new JLabel("NODE3 ANGLE");
        f6 = new JTextField(10);
        b1 = new JButton("SUBMIT ");

        Container container = getContentPane();
        l.setBounds(120, 1, 400, 92);

        l1.setBounds(50, 81, 200, 25);
        f1.setBounds(250, 81, 80, 25);
                     l2.setBounds(300, 81, 250, 25);
        f2.setBounds(450, 81, 80, 25);
                      l3.setBounds(50, 141, 160, 25);
        f3.setBounds(150, 141, 80, 25);
                      l4.setBounds(300, 141, 220, 25);
      f4.setBounds(450, 141, 80, 25);
               l5.setBounds(50, 200, 160,25);
      f5.setBounds(150, 200, 80,25);
              l6.setBounds(300, 200, 220, 25);
```

```
f6.setBounds(450, 200, 80, 25);

  b1.setBounds(245,270, 100, 25);

  container.add(l);
  container.add(l1);
  container.add(f1);

                //container.add(l2);
// container.add(f2);

                //container.add(l3);
 // container.add(f3);

                //        container.add(l4);
                //        container.add(f4);

                container.add(l5);
                container.add(f5);

                //container.add(l6);
                //container.add(f6);
  container.add(b1);
  b1.addActionListener(this);
  container.setLayout(null);
  setSize(new Dimension(550, 400));
  setLocation(new Point(200, 125));
  setVisible(true);
  setDefaultCloseOperation(3);
  container.setBackground(new Color(240, 240, 240));
  setTitle("XLP FOR COMMUNICATION");
  l.setFont(new Font("Verdana", 1, 15));
  l1.setFont(new Font("VERDANA", 2, 13));
  l1.setFont(new Font("Verdana", 1, 13));
                 l2.setFont(new Font("VERDANA", 2, 13));
  l2.setFont(new Font("Verdana", 1, 13));
   l3.setFont(new Font("VERDANA", 2, 13));
  l3.setFont(new Font("Verdana", 1, 13));
                l4.setFont(new Font("VERDANA", 2, 13));
  l4.setFont(new Font("Verdana", 1, 13));
                l5.setFont(new Font("VERDANA", 2, 13));
  l5.setFont(new Font("Verdana", 1, 13));
                l6.setFont(new Font("VERDANA", 2, 13));
  l6.setFont(new Font("Verdana", 1, 13));
  b1.setFont(new Font("VERDANA", 2, 13));
  b1.setFont(new Font("Verdana", 1, 13));

                try
  {
      Class.forName("com.mysql.jdbc.Driver");
```

```
          conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");
          st = conn.createStatement();
                              st1 = conn.createStatement();
                              addConnectedNode();

       }
       catch(Exception exception)
       {
          exception.printStackTrace();
       }
          }

    public void actionPerformed(ActionEvent actionevent)
    {


          }
          public void addConnectedNode()
    {
       try
       {
          rs = st.executeQuery("select node1angle,node2angle,node3angle from angledetails ");
                             while(rs.next())
          {

              s = Integer.parseInt(rs.getString(1));
                                        s1 = Integer.parseInt(rs.getString(2));
                                        s2 = Integer.parseInt(rs.getString(3));

                                        if((s<s1)&&(s<s2))
                                                f1.setText(rs.getString(1));
                                        if((s1<s)&&(s1<s2))
                                                f1.setText(rs.getString(2));
                                        if((s2<s)&&(s2<s1))
                                                f1.setText(rs.getString(3));
                             }

                    }
                     catch(Exception exception)
       {
          exception.printStackTrace();
       }
    }


    public static void main(String args[])
    {
       JFrame.setDefaultLookAndFeelDecorated(true);
```

```
        JDialog.setDefaultLookAndFeelDecorated(true);
        try
        {
            UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
        }
        catch(Exception exception)
        {
            System.out.println("Failed loading L&F: ");
            System.out.println(exception);
        }
        new AngleCommu();
    }

    Connection conn;
    Statement st,st1;
    ResultSet rs,rs1;
            int s,s1,s2,s3;
            String node3,node3angle,node1,nodelangle,node2,node2angle;
    JLabel l;
    JLabel l1;
    JTextField f1;
            JLabel l2;
    JTextField f2;
            JLabel l3;
    JTextField f3;
            JLabel l4;
    JTextField f4;
             JLabel l5;
    JTextField f5;
             JLabel l6;
    JTextField f6;
    JButton b1;
```

Main.java

```
package xlp;
import java.io.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;



  class Main extends JFrame implements ActionListener{


            JFrame f;
            JButton b1,b2;
```

```
        JLabel l;



        Main()
        {


           f= new JFrame("MENU ");
                  f.setDefaultCloseOperation(EXIT_ON_CLOSE);
                  f.setLayout(null);

                  l=new JLabel("XLP FOR COMMUNICATION");
                  l.setFont(new Font("Georgia",Font.BOLD,20));
                  l.setBounds(100,50,300,50);
                  f.getContentPane().add(l);

                  b1=new JButton("PRIORITIZATION MECHANISM");
                  b1.setFont(new Font("Georgia",Font.BOLD,14));
                  b1.setBounds(150,150,150,20);
                  f.getContentPane().add(b1);
                  b1.addActionListener(this);

                  b2=new JButton("ANGLEBASED MECHANISM");
                  b2.setFont(new Font("Georgia",Font.BOLD,14));
                  b2.setBounds(150,250,150,20);
                  f.getContentPane().add(b2);
                  b2.addActionListener(this);
                   f.setVisible(true);
            f.setSize(430,350);
                                f.setResizable(false);


         }
        public void actionPerformed(ActionEvent e)
    {
                  try
                  {
                            String s=e.getActionCommand();

                                     if(s.equals("PRIORITIZATION MECHANISM"))
                                     {
                                               dispose();
                                               new source();

                                     }
                                     if(s.equals("ANGLEBASED MECHANISM"))
                                     {
                                               dispose();
                                               new Anglebased();
```

```
                                    }


                        }
                        catch(Exception e1)
                        {
                                System.out.println("Error");
                        }
                }



                public static void main(String args[])
                {
              new Main();
                }
}
```

NodeCommunication.java

```
package xlp;
import java.io.*;
import java.util.*;
import java.sql.*;

class NodeCommunication
{

   Connection conn;
   Statement st,st1;
   ResultSet rs,rs1;
        static int j=1;
public NodeCommunication()
        {
                try
                {

          Class.forName("com.mysql.jdbc.Driver");

        conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");
Connection
conn1=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root","root")
;
                                    st = conn.createStatement();
                                   st1 = conn1.createStatement();


             rs=st.executeQuery("select snode,sgpi,sqpi,threshold from source");
```

```
                                        int res=0;

                                    while(rs.next())
                                        {

                                                snode=rs.getString(1);

                                                    sgpi=Integer.parseInt(rs.getString(2));
                                                    sqpi=Integer.parseInt(rs.getString(3));

                                                threshold=Integer.parseInt(rs.getString(4));
                                                res=compare(snode,sgpi,sqpi,threshold);



                                                if(res==1)
                                              {
                                                    st1.executeUpdate("insert into
communication(participate,pid) values('"+snode+"','"+ j++ +"')");

                                              }


                                              else
                                              {
                                              st1.executeUpdate("insert into
communication(nonparticipate,pid) values('"+snode+"','"+ j++ +"')");


                                              }

                                        }
                                  //dispose();
                    new Nodedetails();

                    }
                    catch(Exception e)
                          {
                          System.out.println("node communication");
                                  e.printStackTrace();
                          }
          }


          public int compare(String  snode,int sgpi,int sqpi,int threshold)
                    {
                                  int b=sgpi+sqpi;
                                  if((b<buffer) && (RTS>threshold) && (Energy>Remienergy))
                                              {
```

```
                                                                              return 1;

                                                            }
                                                            else
                                                                        {
                                                                                    return 0;

                                                                        }
                        }


            public static void main(String args[])
                        {
                                    new NodeCommunication();
                        }
String snode;
int sgpi;
int sqpi;
int threshold;
int v;
 int[] qg = new int [15];
int RTS=100;
int buffer=50;
int Energy=100;
int Remienergy=50;
            }
```

Source.java

```
package xlp;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.PrintStream;
import java.net.InetAddress;
import java.sql.*;
import javax.swing.*;

public class source extends JFrame
    implements ActionListener
{

    public source()
    {

            l=new JLabel(" Source Node Details");
        l1 = new JLabel("Source Node");
        f1 = new JTextField(10);
                        l5 = new JLabel("Node Angle");
        f5 = new JTextField(10);
        l2 = new JLabel("SGPI");
```

```
f2 = new JTextField(10);
                l3 = new JLabel("SQPI");
f3 = new JTextField(10);
                l4 = new JLabel("Threshold");
                f4 = new JTextField(10);

b1 = new JButton("APPLY");
b2 = new JButton("NEXT");

Container container = getContentPane();
   l.setBounds(100,20,300,50);
l1.setBounds(35, 71, 160, 25);
f1.setBounds(240, 71, 100, 25);
               l5.setBounds(35, 121, 160, 25);
f5.setBounds(240, 121, 100, 25);
l2.setBounds(35, 171, 160, 25);
f2.setBounds(240, 171, 100, 25);
               l3.setBounds(35, 221, 160, 25);
f3.setBounds(240, 221, 100, 25);
               l4.setBounds(35, 271, 160, 25);
f4.setBounds(240, 271, 100, 25);

b1.setBounds(60,340, 80, 25);
b2.setBounds(210, 340, 80, 25);
   container.add(l);
container.add(l1);
container.add(f1);
               container.add(l5);
container.add(f5);
container.add(l2);
container.add(f2);
               container.add(l3);
container.add(f3);
                          container.add(l4);
container.add(f4);

container.add(b1);
b1.addActionListener(this);
container.add(b2);
b2.addActionListener(this);
container.setLayout(null);
setSize(new Dimension(400,450));
setVisible(true);
               setResizable(false);

container.setBackground(new Color(240, 240, 240));
setTitle("CROSS LAYER PROTOCAL");
setLocation(new Point(200, 125));
l.setFont(new Font("Georgia",Font.BOLD,20));
l1.setFont(new Font("VERDANA", 2, 13));
```

```
                l1.setFont(new Font("VERDANA", 1, 13));
                l2.setFont(new Font("VERDANA", 2, 13));
        l2.setFont(new Font("Verdana", 1, 13));
                l3.setFont(new Font("VERDANA", 2, 13));
        l3.setFont(new Font("Verdana", 1, 13));
                l4.setFont(new Font("VERDANA", 2, 13));
        l4.setFont(new Font("Verdana", 1, 13));
                l5.setFont(new Font("VERDANA", 2, 13));
        l5.setFont(new Font("Verdana", 1, 13));



        b1.setFont(new Font("VERDANA", 2, 13));
        b1.setFont(new Font("Verdana", 1, 13));
        b2.setFont(new Font("VERDANA", 2, 13));
        b2.setFont(new Font("Verdana", 1, 13));
        try
        {
            Class.forName("com.mysql.jdbc.Driver");

          conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");
                                        st = conn.createStatement();


        }
        catch(Exception exception)
        {
          exception.printStackTrace();
        }

    }

    public void actionPerformed(ActionEvent actionevent)
    {
        if(actionevent.getSource() == b1)
                {
          try
          {
              snode = f1.getText();
              sgpi=f2.getText();
                                        sqpi=f3.getText();
                                        threshold=f4.getText();
                                        angle=f5.getText();
                                        if(snode.length() <= 0)
                                        {
                                                JOptionPane.showMessageDialog(this,"Node name
field should not be empty");
                                                f1.requestFocus();
                                                return;
                                        }
                                        if(sgpi.length() <= 0)
                                        {
                                                  JOptionPane.showMessageDialog(this,"Sgpi field
should not be empty");
                                                  f2.requestFocus();
                                                  return;
                                        }if(sqpi.length() <= 0)
                                        {
                                                  JOptionPane.showMessageDialog(this,"Sqpi field
should not be empty");
                                                  f3.requestFocus();
                                                  return;
                                        }if(threshold.length() <= 0)
                                        {
                                                  JOptionPane.showMessageDialog(this,"Threshold field
should not be empty");
                                                  f4.requestFocus();
                                                  return;
                                        }

                                        if(angle.length() <= 0)
                                        {
                                                  JOptionPane.showMessageDialog(this,"Threshold field
should not be empty");
                                                  f4.requestFocus();
                                                  return;
                                        }
             rs = st.executeQuery("select * from source where snode='"+snode+"' ");
             if(rs.next())
             {
                                                   String mname = "NODE DETAILS ALREADY
EXISTS";
                JOptionPane.showMessageDialog(null, mname, "CLICK Ok", 1);
             }
                                        else
             {
            int i=st.executeUpdate("insert into source values
('"+snode+"','"+angle+"','"+sgpi+"','"+sqpi+"','"+threshold+"')");
                if(i>0)
                                                  {
              String node = "Nodes Added To XLP";
                JOptionPane.showMessageDialog(null, node, "CLICK Ok", 1);
                                                  f2.setText("");
                                                  f3.setText("");
                                                  f4.setText("");
                                                  f5.setText("");
                }
            }
        }
        catch(Exception exception)
        {
            exception.printStackTrace();
        }
                    }
    if(actionevent.getSource() == b2)
    {

                                dispose();
                    new NodeCommunication();


    }
}


public static void main(String args[])
{
    JFrame.setDefaultLookAndFeelDecorated(true);
    JDialog.setDefaultLookAndFeelDecorated(true);
    try
    {
        UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
    }
    catch(Exception exception)
    {
        System.out.println("Failed loading L&F: ");
        System.out.println(exception);
    }
    new source();
}

Connection conn;
Statement st;
ResultSet rs;
int i;
JLabel l;
JLabel l1;
JTextField f1;
JLabel l2;
JTextField f2;
        JLabel l3;
JTextField f3;
                    JLabel l4;
JTextField f4;
         JLabel l5;
JTextField f5;
```

```
    JButton b1;
    JButton b2;
    String snode;
   String sgpi;
            String sqpi;
            String angle;
   String threshold;
}
```

Nodedetails.java

```
package xlp;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.net.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.util.*;
public class Nodedetails implements ActionListener
{
            JFrame frame = new JFrame();
            JLabel l=new JLabel();
            JLabel l1=new JLabel();
            JLabel l2=new JLabel();
            JLabel h1,h2,h3;
            JPanel panel;
            JButton b;
            Statement stmt,stmt1,stmt2,stmt3,stmt4,stmt5,stmt6,stmt7;
            ResultSet rs,rs1,rs2,rs3,rs4,rs5,rs6;
             Connection con,con1,con2,con3,con4,con5;

            static  int k=1,d=1;

            String active;
            Vector v2=new Vector();
            Vector v3=new Vector();
    boolean flag1=false,flag2=false;
            int xloc=0,yloc=15,p=5;
            int j=0;

            String s[]=new String[40];

            String pt[]=new String[40];
  public   Nodedetails()
              {

            h1=new JLabel("SOURSE");
            h1.setBounds(20,38,100,30);
```

```
        h1.setFont(new Font("vardana",Font.ITALIC+Font.BOLD,12));
        h3=new JLabel("PATHS");
        h3.setBounds(225,38,100,30);
        h3.setFont(new Font("vardana",Font.ITALIC+Font.BOLD,12));

        l.setFont(new Font("VARDANA",Font.ITALIC,25));
        l.setText("Initiative Node Details");
        l.setBounds(80,30,250,20);
                l1.setFont(new Font("Courier New",Font.PLAIN,16));
                l1.setText("Participate");
                l1.setBounds(120,100,150,30);

                l2.setFont(new Font("Courier New",Font.PLAIN,16));
                l2.setText("Non Participate");
                l2.setBounds(100,150,180,30);
        b=new JButton();
        b.setText("Node Communication");
        b.setBounds(130,200,200,25);
b.addActionListener(this);
        frame.add(l);
        frame.add(b);


        frame.setSize(450, 350);
        frame.setTitle("Initiative Node Details");
        frame.setLayout(null);
        frame.setLocation(new Point(200,100));
        frame.setVisible(true);


try{
                        Class.forName("com.mysql.jdbc.Driver");

        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root","r
oot");

        con1=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");

        con2=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");

        con3=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");

        con4=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");

        con5=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");
```

```
                        stmt=con.createStatement();
                        stmt1=con1.createStatement();
                        stmt2=con.createStatement();
                        stmt3=con.createStatement();
                stmt4=con2.createStatement();
                stmt5=con3.createStatement();
                stmt6=con4.createStatement();
                stmt7=con5.createStatement();

        ResultSet rs=stmt.executeQuery("select participate from communication where
participate!='NULL'");
        while(rs.next())
        {
       v2.addElement(rs.getString(1));
        }
ResultSet rs1=stmt1.executeQuery("select  nonparticipate from communication where
nonparticipate!='NULL'");
        while(rs1.next())
        {
      v3.addElement(rs1.getString(1));
        }
        JComboBox s1=new JComboBox(v2);
        s1.setBounds(260,100,90,30);

        JComboBox s2=new JComboBox(v3);
        s2.setBounds(260,150,90,30);

        Vector v1=new Vector();

        frame.add(l1);
        frame.add(l2);

        frame.add(s1);
        frame.add(s2);

        //l1.setBounds(10+xloc,70,80,25);
        xloc=xloc+130;
        frame.repaint();
}

catch(Exception ex){ex.printStackTrace();}
        }

          public void actionPerformed(ActionEvent actionevent)
   {

                if(actionevent.getSource() == b)
       {
                        try{
```

```
/*ResultSet rs=stmt.executeQuery("select participate from communication where
participate!='NULL'");
          while(rs.next())
          {

      v2.addElement(rs.getString(1));
          }*/

                    Enumeration e = v2.elements();
                    int k=0;
                    while(e.hasMoreElements() )
                    {
                              active=(String)e.nextElement();


   if(k<5)
                    {

                    int m=stmt3.executeUpdate("insert into region(active,aid,reg)values
('"+active+"','"+k+++"','A1' )");


                    }
                    else if(k<10)
                              {

                              int m=stmt5.executeUpdate("insert into region(active,aid,reg)values
('"+active+"','"+k+++"','A2' )");

                              }
                              else if(k<25)
                              {
                              int m=stmt7.executeUpdate("insert into region(active,aid,reg)values
('"+active+"','"+k+++"','A3' )");

                              }
                              else{ }

                    }

                                        /* if(d<=5)
                                        {
                                                   System.out.println("Inside try 5");
                              rs2=stmt2.executeQuery("Select participate from communication
where participate!='NULL'  AND pid BETWEEN 1 AND 5");
                                                  while(rs2.next())
                                                  {
                                                            String reg;
                                        active=rs2.getString(1);
                                         int m=stmt3.executeUpdate("insert into
region(active,aid,reg)values ('"+active+"','"+k+++"','A1' )");
                                         System.out.println(m);
                                         d++;
                                         System.out.println(d);

                                       }

                               }
                         if(d<=10)
                               {
                                       rs3=stmt4.executeQuery("Select participate from
communication where participate!='NULL'  AND pid BETWEEN 6 AND 10");
                                       while(rs3.next())
                                       {

                                        active=rs3.getString(1);
                                         int m=stmt5.executeUpdate("insert into
region(active,aid,reg)values ('"+active+"','"+k+++"','A2' )");
                                         System.out.println(m);
                                         d++;

                                       }

                               }
                         if(d<=18)
                                       {
                                       rs4=stmt6.executeQuery("Select participate from
communication where participate!='NULL'  AND pid BETWEEN 11 AND 18");
                                       while(rs4.next())
                                       {

                                        active=rs4.getString(1);
                                         int m=stmt7.executeUpdate("insert into
region(active,aid,reg)values ('"+active+"','"+k+++"','A3' )");
                                         System.out.println(m);
                                         d++;

                                       }

                               }*/

                               //dispose();

                        new PriorityRegions();



                        }
```

```
                              catch(Exception e){System.out.println(e);}


                        }

                  }



  public static void main(String rags[])
        {
          new Nodedetails();
    }
}
```

Anglebased.java

```
package xlp;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.PrintStream;
import java.sql.*;
import javax.swing.*;
import java.io.File;
 import javax.swing.JComboBox ;
import javax.swing.JFileChooser;
import java.io.*;

  public class Anglebased extends JFrame
    implements ActionListener
{
            String mys;
            String mys1;
    public Anglebased()
    {

        l = new JLabel("                    ANGLEBASED MECHANISAM      ");
        l1 = new JLabel("SOURCE SINK");
        f1 = new JComboBox();
                    l2 = new JLabel("DESTINATION SINK");
        f2 = new JComboBox();

              l3=  new JLabel("SOURCE  NODES");
        f3= new JComboBox();
                   l4=  new JLabel("DESTINATION  NODES");
        f4= new JComboBox();
                    l5 = new JLabel("BROWSE THE FILE");
        f5 = new JTextField(10);
        b1 = new JButton("SEND ");
        b2 = new JButton("COMPLETE");
```

```
                b3 = new JButton("BROWSE");
                b4 = new JButton("REQUEST");

                jta=new JTextArea(5,10);
                jsp=new JScrollPane(jta);

        Container container = getContentPane();
        l.setBounds(120, 1, 400, 92);

        l1.setBounds(39, 81, 120, 25);
        f1.setBounds(180, 81, 80, 25);
                l2.setBounds(360, 81, 250, 25);
        f2.setBounds(550, 81, 80, 25);
                 l3.setBounds(45, 171, 120, 25);
        f3.setBounds(180, 171, 80, 25);
                 l4.setBounds(360, 171, 220, 25);
       f4.setBounds(550, 171, 80, 25);
           l5.setBounds(100, 231, 220, 25);
       f5.setBounds(280, 231, 100, 25);
          jsp.setBounds(200,275,200,200);

        b1.setBounds(245, 500, 100, 25);
        b2.setBounds(400, 500, 100, 25);
                b3.setBounds(390,231,100,25);
                b4.setBounds(100,500,100,25);
        container.add(l);
        container.add(l1);
                    container.add(jsp);

        container.add(f1);

                container.add(l2);
       container.add(f2);

                container.add(l3);
        container.add(f3);

                    container.add(l4);
                    container.add(f4);

                container.add(l5);
                container.add(f5);

        container.add(b1);
                b1.setEnabled(false);
        b1.addActionListener(this);
    f1.addActionListener(this);
    f2.addActionListener(this);
    f3.addActionListener(this);
    f4.addActionListener(this);
```

```
        container.add(b2);
        b2.addActionListener(this);
                        container.add(b3);
       b3.addActionListener(this);
                        container.add(b4);
       b4.addActionListener(this);
        container.setLayout(null);
        setSize(new Dimension(800, 600));
        setLocation(new Point(200, 125));
        setVisible(true);
        setDefaultCloseOperation(3);
        container.setBackground(new Color(240, 240, 240));
        setTitle("XLP FOR ANGLEBASED");
        l.setFont(new Font("Verdana", 1, 15));
        l1.setFont(new Font("VERDANA", 2, 13));
        l1.setFont(new Font("Verdana", 1, 13));
                         l2.setFont(new Font("VERDANA", 2, 13));
        l2.setFont(new Font("Verdana", 1, 13));
         l3.setFont(new Font("VERDANA", 2, 13));
        l3.setFont(new Font("Verdana", 1, 13));
                        l4.setFont(new Font("VERDANA", 2, 13));
        l4.setFont(new Font("Verdana", 1, 13));
                        l5.setFont(new Font("VERDANA", 2, 13));
        l5.setFont(new Font("Verdana", 1, 13));
        b1.setFont(new Font("VERDANA", 2, 13));
        b1.setFont(new Font("Verdana", 1, 13));
        b2.setFont(new Font("VERDANA", 2, 13));
        b2.setFont(new Font("Verdana", 1, 13));
                         b3.setFont(new Font("Verdana", 1, 13));
                          b4.setFont(new Font("Verdana", 1, 13));
                        try
        {
             Class.forName("com.mysql.jdbc.Driver");

            conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root",
"root");
            st = conn.createStatement();
                                     st1 = conn.createStatement();
                                      addConnectedNode();

        }
        catch(Exception exception)
        {
            exception.printStackTrace();
        }
            }
            public void addConnectedNode()
    {
        try
        {
           rs = st.executeQuery("select sink from region where sink!='NULL' ");
                              while(rs.next())
           {
              String s = rs.getString(1);
                                        f1.addItem(s);
                              }

                    rs1 = st1.executeQuery("select sink from region where sink!='NULL'  ");
                                        while(rs1.next())
                                                            {
                                                                      String s1= rs1.getString(1);
                                                                      f2.addItem(s1);
                                                                       s1 = "";

                                                            }
                    }
                     catch(Exception exception)
        {
           exception.printStackTrace();
        }
    }
    public void actionPerformed(ActionEvent actionevent)
    {

                    if(actionevent.getSource() == b1)
        {

                              Integer i1=Integer.parseInt(mys);
                              Integer i2=Integer.parseInt(mys1);
                              if(i1<i2){
                                        JOptionPane.showMessageDialog(this," Clockwise direction
to:"+temp1);
                              }
                              else{
                                        JOptionPane.showMessageDialog(this," Anti-Clockwise
direction to:"+temp1);
                              }
                              try{
                    System.out.println("temp Button Node"+temp);
                      srcnode= (String)f3.getSelectedItem();
                      System.out.println("Button 1111"+srcnode);
                       desnode = f4.getSelectedItem().toString();
                       file=f5.getText();
                       System.out.println(file);
                       if(srcnode.equals(desnode))
                                        {
                                JOptionPane.showMessageDialog(this," Source Node And
Destination Node should not be Same");
                                                  f3.requestFocus();
```

```
                                                return;
                                        }


                    if(srcnode.length() <= 0)
                                        {
                                                JOptionPane.showMessageDialog(this," Source Node
field should not be empty");
                                                f3.requestFocus();
                                                return;
                                        }if(desnode.length() <= 0)
                                        {
                                                JOptionPane.showMessageDialog(this," Destination
Node field should not be empty");
                                                f4.requestFocus();
                                                return;
                                        }if(file.length() <= 0)
                                        {
                                                JOptionPane.showMessageDialog(this," file field
should not be empty");
                                                f5.requestFocus();
                                                return;
                                        }if(regsink.length() <= 0)
                                        {
                                                JOptionPane.showMessageDialog(this," Source reg
field should not be empty");
                                                f1.requestFocus();
                                                return;
                                        }if(neisink.length() <= 0)
                                        {
                                                JOptionPane.showMessageDialog(this," Destination
Reg field should not be empty");
                                                f2.requestFocus();
                                                return;
                                        }
                                         long block = getBlocks(chooser.getSelectedFile());
                                         random = new
RandomAccessFile(chooser.getSelectedFile().getPath(),"r");
                                        createBlocks(block,chooser.getSelectedFile().getName());
                                        random.close();
                                                Statement st2=conn.createStatement();

          System.out.println(regsink);System.out.println(neisink);System.out.println(srcnode);S
ystem.out.println(desnode);
                                        int i=st2.executeUpdate("insert into file
values('"+regsink+"','"+srcnode+"','"+neisink+"','"+desnode+"','"+file+"')");
                                                        if(i>0)
                                        {
                                                                System.out.println(i);
```

```
                                                            String mname = "FILE  FROM
SOURCE "+srcnode+" TO DESTINATION    "+desnode+"";
                JOptionPane.showMessageDialog(null, mname, "REQUEST", 1);
                                                Thread.sleep(2000);
                                                String kill="Acknowldgement from    "+desnode+"";
                                                 JOptionPane.showMessageDialog(null, kill,
"REQUEST", 1);
                                                b1.setEnabled(true);
                                        }
                              }catch(Exception e){System.out.println(e);}

      }

      if(actionevent.getSource() == b2)
                  {
                              dispose();
                  }
         if(actionevent.getSource() == b3)
                           {

                                 try
                                   {
                                          chooser = new JFileChooser();
                                          chooser.setCurrentDirectory(new File("."));
                                          chooser.setFileFilter(new
javax.swing.filechooser.FileFilter()
                                                   {
                                                            public boolean accept(File f)
                                                                        {
                                                                              return
f.getName().toLowerCase().endsWith(".gif")
                                                                                  || f.isDirectory();
                                                                        }

                                                            public String getDescription()
                                                                      {
                                                                              return "GIF Images";
                                                                       }
                                                   }




                           );

                                          int r = chooser.showOpenDialog(new JFrame());
                                          if (r == JFileChooser.APPROVE_OPTION)
                                          {
                                                  String name =
chooser.getSelectedFile().getName();
```

```
                                        System.out.println(name);
                                        //f5.setText(name);
                                    }
                               }
                              catch(Exception e)
                              {
                                   e.printStackTrace();
                              }
                              try
                         {

                              if(chooser.getSelectedFile().getName().endsWith(".txt") ||
chooser.getSelectedFile().getName().endsWith(".java")){
                              FileInputStream fos=new
FileInputStream(chooser.getSelectedFile());
                              f5.setText(chooser.getSelectedFile().getName());
                              int b;
                              String st="";
                              while((b=fos.read())!=-1)
                              {
                                   st+=(char)b;
                              }
                              jta.setText(st);
                              fos.close();
                              }else{
                              f5.setText(chooser.getSelectedFile().getName());
                              jta.setText("Selected file type cannot be viewed");
                         }
                         }catch(Exception e)
                              {
                              JOptionPane.showMessageDialog(this,"Error in uploading a
file");
                              }

                         }
                         if(actionevent.getSource() == b4)
    {

                         String dnode = f4.getSelectedItem().toString();
                         String snode= (String)f3.getSelectedItem();
                          if(snode.equals(dnode))
                              {
                           JOptionPane.showMessageDialog(this," Source Node And
Destination Node should not be Same");
                                   f3.requestFocus();
                                   return;
                              }

                                   //   container.add(b1);
                          String mname = "RTS  FROM SOURCE  "+snode+"    TO
DESTINATION      "+dnode+"";
          JOptionPane.showMessageDialog(null, mname, "REQUEST", 1);

                                   b1.setEnabled(true);

                                   try
                                   {
                                         Thread.sleep(2000);

                                          String lilly="CTS FROM DESTINATION
"+dnode+" TO SOURCE    "+snode+"";
                                          JOptionPane.showMessageDialog(null, lilly,
"REQUEST", 1);

                                   }
                                   catch (Exception k)
                                   {
                                         k.printStackTrace();
                                   }
    }
            if(actionevent.getSource()==f1)
            {
                        try
                        {
                          f3.removeAllItems();
                          regsink = (String)f1.getSelectedItem();
                          if(regsink.equals("S1"))
                          {
                                Statement st2=conn.createStatement();
                          System.out.println("selected Region Sink:"+regsink);
                        ResultSet rs2=st2.executeQuery("select active from region
where reg='A1' ");

                          while(rs2.next())
                                {

                                                  regnode=rs2.getString(1);
                                                  System.out.println("selected
Region Sink: Nodes"+regnode);
                                                            if(regnode!=null)
                                         {
                                                  f3.addItem(regnode);

                                         }

                                }st2.close();
                                rs2.close();
                          }
                          if(regsink.equals("S2"))
```

```
                                        {
                                                 Statement st2=conn.createStatement();
                                                System.out.println("selected Region
Sink:"+regsink);
                                  ResultSet        rs2=st2.executeQuery("select active from region
where reg='A2' ");
                                                System.out.println("    RegNodes attching to
combobox");
                                                while(rs2.next())
                                                {

                                                 regnode=rs2.getString(1);
                                                            System.out.println("selected
Region Sink: Nodes"+regnode);
                                                            f3.addItem(regnode);


                                                }st2.close();rs2.close();
                                        }
                                         if(regsink.equals("S3"))
                                         {
                                                       Statement st2=conn.createStatement();
                                        System.out.println("selected Region Sink:"+regsink);
                                        ResultSet        rs2=st2.executeQuery("select active from
region where reg='A3' ");
                                        System.out.println("    RegNodes attching to
combobox");
                                        int a=0;
                                        while(rs2.next())
                                         {

                                                regnode=rs2.getString(1);

                                                System.out.println("selected Region Sink:
Nodes"+regnode);
                                                if(regnode!=null)
                                                 {
                                                f3.addItem(regnode);

                                                 }

                                        }st2.close();rs2.close();

                                  }
                            }
                                  catch (Exception e)
                                  {
                                        e.printStackTrace();
                                  }
                      }
```

```
                              if(actionevent.getSource()==f3)
                    {
                                         try
                                         {
                                                  String temp=(String)f3.getSelectedItem();
                                                  srcnode1=temp;
                                                   Class.forName("com.mysql.jdbc.Driver");
                                                  Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root","root");
                                                  PreparedStatement ps=conn.prepareStatement("select
angle from source where snode=?");
                                                  ps.setString(1,srcnode1);

                                                  ResultSet rs=ps.executeQuery();
                                                  if(rs.next()){
                                                            mys=rs.getString(1);

                                                  }
                                                  System.out.println("srcnode1 details"+srcnode1);

                                                  System.out.println("Before Null temp"+temp);
                                                   f3.removeAllItems();
                              f3.setForeground(Color.red);
                                        f3.addItem(temp);

                                         }
                                         catch(Exception e){System.out.println(e);}
                    }
                    if(actionevent.getSource()==f2)
                    {
                     try
                                         {
                                                   f4.removeAllItems();
                                                   neisink = (String)f2.getSelectedItem();
                                                   if(neisink.equals("S1"))
                                                   {
                                                             Statement st2=conn.createStatement();
                                                  System.out.println("selected Region Sink:"+regsink);
                                        ResultSet rs2=st2.executeQuery("select active from region
where reg='A1' ");

                                                  while(rs2.next())
                                                            {

                                                                                    String regnode=rs2.getString(1);
                                                                                    System.out.println("selected
Region Sink: Nodes"+regnode);

                                                                                    f4.addItem(regnode);
                                                            }st2.close();rs2.close();
                                                  }
                                                   if(neisink.equals("S2"))
                                                   {
                                                             Statement st2=conn.createStatement();
                                                            System.out.println("selected Region
Sink:"+regsink);
                                        ResultSet         rs2=st2.executeQuery("select active from region
where reg='A2' ");
                                                            System.out.println("     RegNodes attching to
combobox");
                                                            while(rs2.next())
                                                            {

                                                                                    String regnode=rs2.getString(1);
                                                                                    System.out.println("selected
Region Sink: Nodes"+regnode);
                                                                                    f4.addItem(regnode);


                                                            }st2.close();rs2.close();
                                                  }
                                                   if(neisink.equals("S3"))
                                                   {
                                                                      Statement st2=conn.createStatement();
                                                  System.out.println("selected Region Sink:"+regsink);
                                                  ResultSet         rs2=st2.executeQuery("select active from
region where reg='A3'  ");
                                                  System.out.println("     RegNodes attching to
combobox");
                                                  while(rs2.next())
                                                   {

                                                            String regnode=rs2.getString(1);
                                                            System.out.println("selected Region Sink:
Nodes"+regnode);
                                                            f4.addItem(regnode);


                                                  }st2.close();rs2.close();

                                         }
                              }
                                         catch (Exception e)
                                         {
                                                   e.printStackTrace();
                                         }
               }

               if(actionevent.getSource()==f4)
```

```
                {
                try
                                {
                                        temp1=(String)f4.getSelectedItem();
                                        System.out.println("angle based"+temp1);
                                        Class.forName("com.mysql.jdbc.Driver");
                                        Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/crosslayer","root","root");
                                        PreparedStatement ps=conn.prepareStatement("select
angle from source where snode=?");
                                        ps.setString(1,temp1);

                                        ResultSet rs=ps.executeQuery();
                                        while(rs.next()){
                                                mys1=rs.getString(1);

                                        }

                                         f4.removeAllItems();
                                        f4.addItem(temp1);
                                        f4.setForeground(Color.red);

                                }
                                catch(Exception e){System.out.println(e);}
                }
        }
        public long getBlocks(File file){
        long length = file.length();
        tot_blocks=0;
        long size = 0;
        if(length >= 1000){
                size = length/10;
                tot_blocks = 10;
        }
        if(length < 1000 && length > 500){
                size = length/5;
                tot_blocks = 5;
        }
        if(length < 500 && length > 1){
                size = length/3;
                tot_blocks = 3;
        }
        System.out.println("Size Details "+length+" "+size);
        return size;
}
public void createBlocks(long block,String name){

         try{
                 FileOutputStream fout=new FileOutputStream("D:/received/"+name,true);
                 String ext = name.substring(name.lastIndexOf(".")+1,name.length());
```

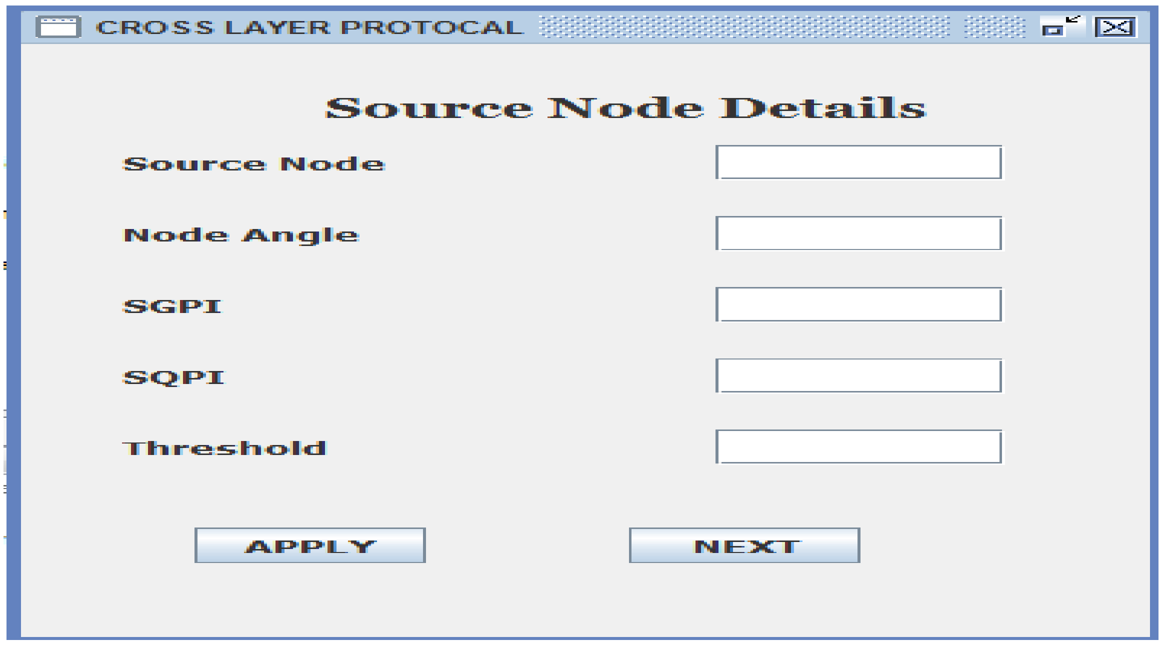
```
                    for(int i=0;i<tot_blocks;i++){
                            byte b[]=new byte[(int)block];
                            random.read(b);
                            random.seek(random.getFilePointer());
                            File file = new File("D:/send/"+i+"_"+name);
                            FileOutputStream out = new FileOutputStream(file);
                            out.write(b,0,b.length);
                            out.flush();
                            out.close();
                            fout.write(b,0,b.length);
                            fout.flush();
                    }
                    fout.close();
            }catch(Exception e){
                    e.printStackTrace();
            }

    }
      public static void main(String args[])
      {
         JFrame.setDefaultLookAndFeelDecorated(true);
         JDialog.setDefaultLookAndFeelDecorated(true);
         try
         {
            UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");
         }
         catch(Exception exception)
         {
            System.out.println("Failed loading L&F: ");
            System.out.println(exception);
         }
         new Anglebased();
      }

      Connection conn;
      Statement st,st1,st2,st3,st4,st5,st6;
      ResultSet rs,rs1,rs2,rs3,rs4,rs5,rs6;
            String s3,s4;
            String
regsink,regnode,neisink,neinode,regsrc,neides,srcnode,desnode,file,temp,temp1,srcnode1;
      JLabel l;
      JLabel l1;
      JComboBox f1;
            JLabel l2;
      JComboBox f2;
            JLabel l3;
      JComboBox f3;
            JLabel l4;
            String src[]=new String[10];
      JComboBox f4;
```

```
            JLabel l5;
    JTextField f5;
    JButton b1;
    JButton b2;
            JButton b3;
            JButton b4;
            JTextArea jta;
            JScrollPane jsp;
            RandomAccessFile random;
            int tot_blocks;
            JFileChooser chooser =null;
}
```

3.2.5 Snapshots


CROSS LAYER PROTOCAL

Source Node Details

Source Node	
Node Angle	
SGPI	
SQPI	
Threshold	

Figure 3.6 – Source Node Details Enter Form

Before starting the cross layer protocol for communicating, the details of the source node must be given clearly in the window that is displayed above. The details that have to be given are source node, threshold, GPI and the QPI value of the source. After entering all these values, apply button has to be pressed by the user.

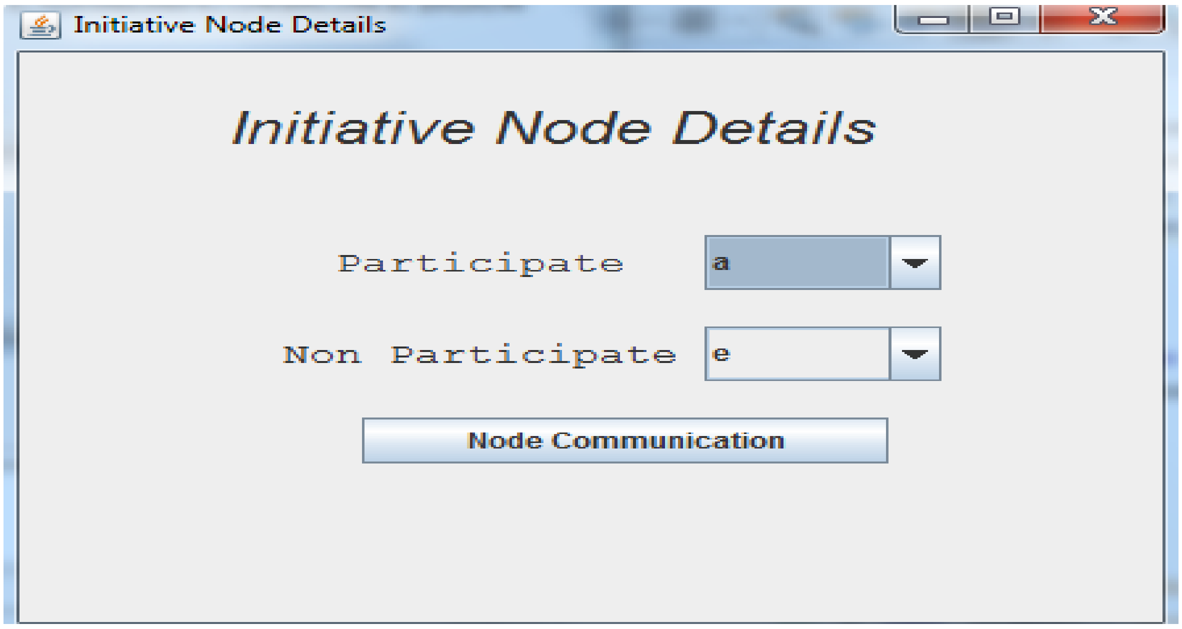
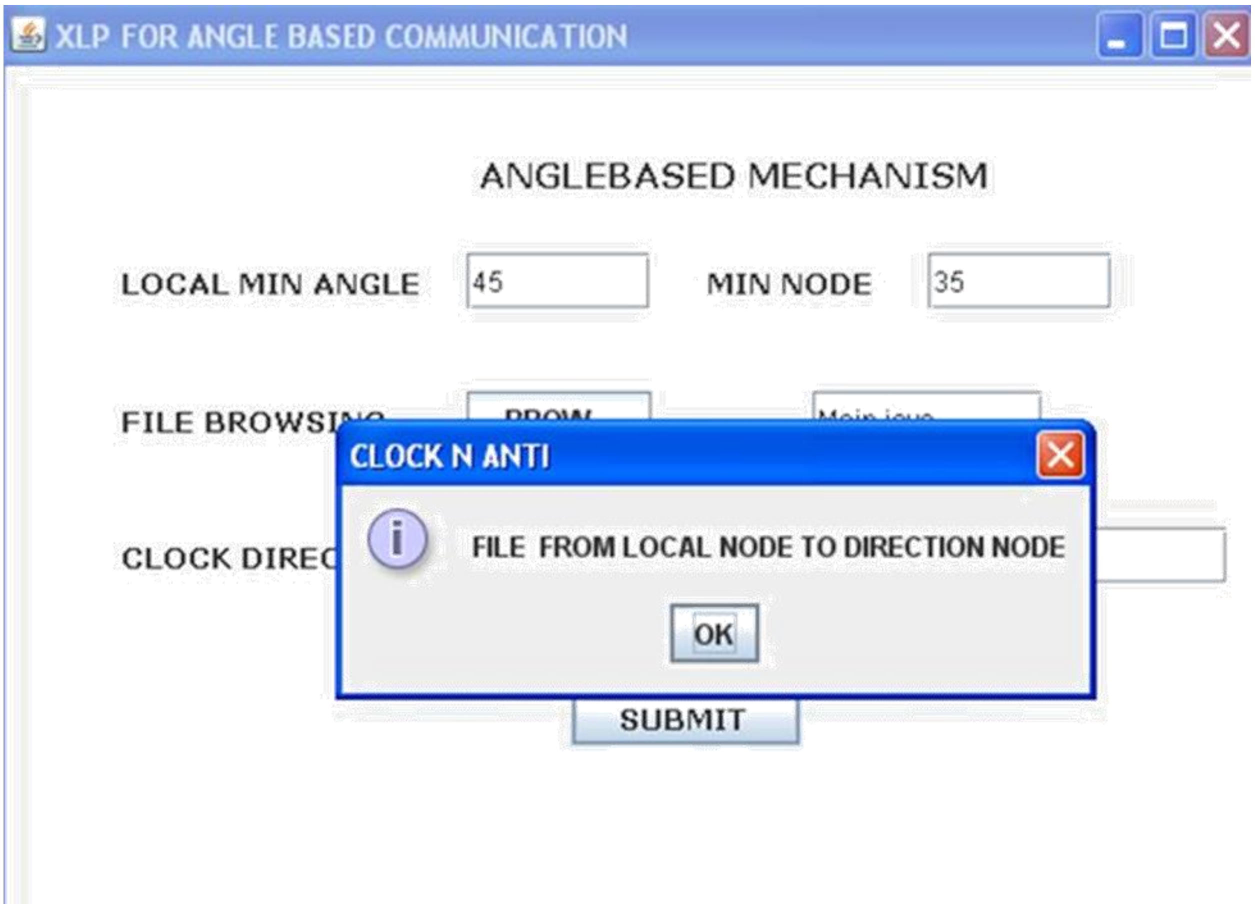


Figure 3.7- Participated and Non participated List.

In addition to the source node there will be another type of nodes known as the initiative node. The details of this initiative node are shown clearly in the above screen shot. Initiative nodes are like the intermediate nodes which can be used to transmit the information from the source node to the destination node. The participative nodes will be participating in the communication and non-participative are the ones which will not be used for the communication.


XLP FOR ANGLE BASED COMMUNICATION

ANGLEBASED MECHANISM

LOCAL MIN ANGLE 45 MIN NODE 35

CLOCK N ANTI

FILE FROM LOCAL NODE TO DIRECTION NODE

Figure 3.8 – Angle based Mechanism

As shown in the above screen shot, a dialog box is displayed that shows that a file is been transmitted successfully from the source node to the angel node. This proves the working of the cross layer protocol in offering better communication for the wireless sensor network

Chapter 4: CONCLUSION

4.1 Conclusion

Cross-layer design is strongly recommended as a new methodology for designing and optimizing the performance for future wireless networks because of the many possible benefits it could bring. The goal of this thesis was to develop a framework for the study of cross-layer design and optimization. Such a framework not only can be used in the context of WSNs but it can also be adapted and applied across different domains of wireless networks. To meet this goal, a generic cross-layer framework and the concept of initiative determination is to improve and optimize the performance of a wireless system was proposed.

The cross layer protocol is considered in this project to provide an efficient communication method for these networks. In the recent past, cross-layering in perfectly well-designed as a communication stack so that state information or data directly flows throughout the given stack has been thoroughly investigated to a great extent. Most of the research studies conducted on WSN and its cross-layer integration techniques have revealed that the feedbacks and results indicated significant energy gains. In a single protocol, certain developed concepts permitted several communication and networking functionalities be successfully implemented to a large extent. It is to be remembered that the cross-layer protocol (XLP) is perfectly implemented in order to provide the functionalities of congestion control, medium access and better routing in any given situation or environment. It is to be highly noted that based on the concept of the initiative determination, the cross-layer protocol (XLP) perfectly serves as a better proof of best concept and better performs receiver based contention, reliable communication, distributed duty cycle operation to realize efficient and initiative-based forwarding in WSNs.

APPENDIX B: REFERENCES

[1] Mehmet c Vuran, Ian F.Akyildiz " XLP: A Cross Layer Protocol for Efficient communication in Wireless Sensor Networks "

[2] Akyildiz,I.F. and Vuran, M.C. (2006) „Cross-Layer Analysis of Error Control in Wireless Sensor Networks‟. „*Proc. IEEE Int'l Conf. Sensor and Ad Hoc Comm. And Networks (SECON).*Reston, VA. 28 September. Available at: http://0 eeexplore.ieee.org.brum.beds.ac.uk/stamp/stamp.jsp?tp=&anumber=4068316. (Accessed on 7 April 2013).

[3] Sanjay Shakkottai, Peter C. Karlsson, and Theodore S. Rappaport, "Cross-Layer Design for Wireless Networks," IEEE Vol. 41, Issue 10, pp. 74-80, Oct 2003.

[4]Piyush Charan, Rajeev Paulus , Mukesh Kumar ,Arvind Kumar Jaiswal "A survey on the performance optimization using cross layer Approach"

[5]Wikipedia :Wireless Sensor Networks