STUDY AND IMPLEMENTATION OF MULTIPATH ROUTING PROTOCOLS IN WIRELESS SENSOR NETWORK

Enrollment No. -101235

Name of Student - SANIDHYA SINGH

Name of supervisor - Dr. YASHWANT SINGH

MAY - 2014

Project report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

DEPARTMENT OF **COMPUTER SCIENCE AND ENGINEERING**

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

Table of Contents

S. No.	Topics	Page no.
	Certificate from the Supervisor	III
	Acknowledgement	IV
	Abstract	V
	Table of Figures	VI
	List of Symbols and acronyms	VIII
1	**Chapter 1: Introduction**	1
	1.1 Introduction	1
	1.2 Genesis of Problem	1
	1.3 Problem Statement	2
	1.4 Objective	2
	1.5 Proposed System	3
	1.6 Approach Used	4
	1.7 Organization of thesis	5
2	**Chapter 2: Literature Survey**	**6**
	2.1 Introduction	6
	2.2 Parts of WSN	6
	2.3 Applications of WSN	8
	2.4 Routing Protocols In WSN	9
	2.5 Routing types in WSN	11

	2.6 AODV Routing Protocol	12
	2.7 AAESS Routing Protocol	14
3	**Chapter 3: Implementation**	**17**
	3.1 Feasibility Study	17
	3.2 Requirement Analysis	17
	3.3 Assumptions	18
	3.4 Project Application Overview	19
	3.5 Block Diagram	20
	3.6 Activity Diagram	21
	3.7 Collaboration Diagram	22
	3.8 EESAA Flow chart	23
	3.9 Flow Chart Of AODV	25
	3.10 Data Flow Diagram	26
	3.11 Simulation	25
	3.12 Output	39
4	**Chapter 4: Conclusion**	**45**
5	**References**	**46**

CERTIFICATE

This is to certify that the work titled "**STUDY AND IMPLEMENTATION OF MULTIPATH ROUTING PROTOCOL IN WIRELESS SENSOR NETWORK**" submitted by "**SANIDHYA SINGH**" in partial fulfillment for the award of degree of B.Tech in 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	………………	Signature of Supervisor	……………………
Name of Student	………………	Name of Supervisor	…………………….
Enrollment No.	.………………	Designation	…………………….
Date	……………….	Date	…………………….

ACKNOWLEDGEMENT

I would like to express my greatest gratitude to my mentor **Dr. Yashwant Singh** who has helped and supported me throughout our project. I'm grateful to him for his continuous support for the project, from the initial advice in the early stages of conceptual inception and through ongoing advice to this day.

Signature of the student ……………………..

Name of Student ……………………..

Date ……………………..

Abstract

Designing efficient and reliable communication protocols for wireless sensor networks in monitoring applications is a challenging task, due to the uncertainty and dynamics of the environment. In this project, two routing protocols namely AODV routing protocol (a reactive protocol) and EESAA routing protocol have been compared. AODV being a reactive protocol it still uses characteristics of a proactive protocol. Routes in AODV are discovered and established and maintained only when and as long as needed. To ensure loop freedom sequence numbers, which are created and updated by each node itself, are used. These allow also the nodes to select the most recent route to a given destination node. Whereas in EESAA, minimize the energy consumption by using the concept of pairing. The algorithm makes many improvements on the way of dividing clusters, strategy of electing the cluster head and construction method of data relay path, the two aspects of inter-cluster energy balance and energy balance among the cluster are taken into account at the same time.

The experiment is carried out using MATLAB environment. Simulation result shows that the proposed algorithms give better performances and reduces the energy consumption.

Table of Figures

S. NO	Figures	Page No
1	1.1 – AODV protocol messaging	4
2	1.1 Software Development Life Cycle	5
3	2.1: Wireless Network Sensor Components	7
4	2.2 : Routing protocol Types In WSN	11
5	2.3 AODV routing protocol example	13
6	2.4 AODV route lookup session	14
7	2.5 : EESAA example	15
8	3.1: Steps undertaken before analysis	19
9	3.2 Block Diagram	20
10	3.3: Activity Diagram	21
11	3.4 Activity Diagram	21
12	3.5 Collaboration Diagram	22
13	3.6 Collaboration Diagram	22
14	3.7 EESAA Flow chart	23
15	3.8 : AODV Flow chart	24
16	3.9 Received route reply handling in AODV	25
17	3.10 Data Flow Diagram	26
18	3.11 Input of AODV protocol - Matrix	39
19	3.12 Output of AODV protocol	40
20	3.13 Output of EESAA protocol – Node types	41
21	3.14 Output of EESAA protocol – No. of rounds	42

22	3.15 Graphical represenation of EESAA protocol	43
23	3.16 Graphical represenation of EESAA protocol	43
24	3.16 Graphical represenation of EESAA protocol	44
25	3.16 Graphical represenation of EESAA protocol	44

List of Symbols and Acronyms

WSN: Wireless Sensor Network.

MANETs: Mobile ad hoc networks.

AODV: Ad hoc On Demand Distance Vector Routing Protocol.

EESAA: Energy Efficient Sleep Awake Aware Routing protocol.

RREQ: Route Request.

RREP: Route Reply

CH: Cluster head

RSSI: Received Signal Strength Indication

SDLC: Software Development Life Cycle

GUI: Graphic User Interface

LAN: Local Area Network

RP: Routing Protocol

UML: Universal Modeling Language

WAN: Wide Area Network

WDS: Wireless Distributed System

Chapter 1: Introduction

1.1 INTRODUCTION

Wireless technology has expanded the limits of our world. Through this innovation, people have been given freedom to work away from their desks or even outside. The newfound freedom that people are beginning to enjoy with their computers has started making the world of technology and nature blend. Wireless Sensor Networks are the next stage of this technology-nature cohesion.

A WSN is formed by densely deployed sensor nodes in an application area. In most deployments, the sensor nodes have self-organizing capabilities, to form an appropriate structure in order to collaboratively perform a particular. The power of WSN lies in creating a pervasive environment capable of remote sensing, monitoring and control. The positive benefits of this are quite obvious; such a technology will be a will achieve fine granularity tracking of what is going on at far away and generally in inaccessible locations.

One important issue when designing WSN is the routing protocol that makes the best use of the severely limited resource presented by WSN, especially the energy limitation. A routing protocol defines about routers communication with each other, disseminating information that enables them to select routes between any two nodes on a computer network. Routing algorithms determine the specific choice of route. Every router has a prior information only of networks attached to it directly. They share this information first among immediate neighbors, and then throughout the network. This way, routers gain knowledge of the topology of the network.

1.2 Genesis of Problem

In present world, everything is becoming wireless, from home technology to military service and from environmental to hospital services. Therefore routing protocols, for network without infrastructures, have to be developed. These protocols determine how messages can be forwarded, from a source node to a destination node which is out of the range of the former, using other mobile nodes of the network. Routing, which includes for example maintenance and discovery of routes, is one of the very challenging areas in communication.

So, this is where my project comes into play. Two routing protocols have been implemented namely AODV and EESAA. In former one routes are discovered and established and maintained only when and as long as needed. This ensures conservation of energy, by switching off the nodes when not required. While later one, minimize the energy consumption by using the concept of pairing and switching between sleep and awake mode.

1.3 Problem Statement

In existing system, when a source wants to send data to destination, it does not know the best path and sends data to all the nodes, leading to more energy consumption than required. The traditional routing algorithms make use of a centralized approach which are basically cover a single path. They even consume more bandwidth than required while sending data.

Existing systems are not so reliable. Sometimes it is possible that data may crash in somewhere between the source and destination. It might happen that the node which received data lost all its energy and is unable to send the data to its adjacent node and thus data is lost. Receiving node can also send the data to wrong neighbor and thus data won't be able to reach its original destination

Existing systems are not so secure. Taking into consideration that WSN applications need to support critical infrastructures (i.e. military, healthcare, environmental etc), security becomes an issue.

1.4 OBJECTIVE

The main objectives of project are:

- To implement multipath routing protocols which overcomes the drawbacks of existing system.
    - Best route to destination will be calculated.
    - The data will be send via the best available route.

- When there is demand for the data, then only the route from Source node to Destination node will be formed.
    - It will help in energy conservation.

- To provide security to the protocol.
    - It will help in sending the data safely to its destination.
    - It will protect the data from malicious activity.

1.5 PROPOSED SYSTEM

The proposed work is to implement secured multipath routing protocols in wireless sensor network. Two multipath routing protocols have been proposed namely AODV and EESAA. First, former will be discussed then the later one.

1.5.1 Ad hoc On-Demand Distance Vector Routing (AODV)

Ad hoc On-Demand Distance Vector (AODV) Routing is a routing protocol for mobile ad hoc networks (MANETs) and other wireless ad hoc networks. It is a reactive protocol, even though it still uses characteristics of a proactive protocol. AODV takes the interesting parts of DSR and DSDV, in the sense that it uses the concept of route discovery and route maintenance of DSR and the concept of sequence numbers and sending of periodic hello messages from DSDV.
Routes in AODV are discovered and established and maintained only when and as long as needed. To ensure loop freedom sequence numbers, which are created and updated by each node itself, are used. These allow also the nodes to select the most recent route to a given destination node.

The AODV Routing Protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. In AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. In an on-demand routing protocol, the source node floods the RouteRequest packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single RouteRequest. The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number (DestSeqNum) to determine an up-to-date path to the destination. A node updates its path information only if the DestSeqNum of the current packet received is greater or equal than the last DestSeqNum stored at the node with smaller hopcount.

1.5.2 Energy Efficient Sleep Awake Aware (EESAA)

EESAA minimizes the energy consumption by using the concept of pairing. Sensor nodes of same application and which are at the minimum distance between them will form a pair for data sensing and communication. This protocol will also use the Cluster Heads selection technique, by selecting CHs on basis of the remaining energy of the nodes. Information from the sensor nodes is forwarded to the cluster heads (CHs) and these CHs are responsible to transmit this information to the base station which is placed far away from the field.

1.6 APPROACH USED

I have implemented two multipath routing protocols, which overcomes the drawback of single routing problems and other protocols.

1. **AODV:** Routes in AODV are discovered and established and maintained only when and as long as needed. To ensure loop freedom sequence numbers, which are created and updated by each node itself, are used. These allow also the nodes to select the most recent route to a given destination node. AODV takes advantage of route tables. In these it stores routing information as destination and next hop addresses as well as the sequence number of a destination. Next to that a node also keeps a list of the precursor nodes, which route through it, to make route maintenance easier after link breakage. To prevent storing information and maintenance of routes that are not used anymore each route table entry has a lifetime. If during this time the route has not been used, the entry is discarded.

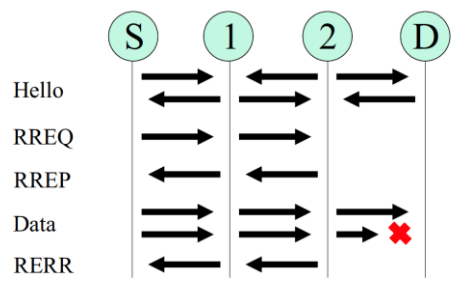


Figure 1.1 – AODV protocol messaging

2. **EESAA**: According to this scheme, the nodes switch between "Sleep" and "Awake" modes during a particular communication Interval. Firstly node in a pair, switch into Awake mode also called "Active-mode". This happens if its distance from the sink is less than its coupled node. Node which is in "Active-mode" will gather data from surroundings and transmits it to the CHs. During this time period transceiver of the coupled node will remain "off" and switches into "Sleep-mode". In the next step, nodes in "Active-mode" will switch into "Sleep-mode" and "Sleep-mode" nodes switches into "active-mode". By this process the consumption of energy can be minimized because nodes which are in "Sleep-modes" save their energy by not interacting with the CHs. "Unpaired nodes" remain in "Active-mode" for each round till their energy resources died. This method lessons the energy consumption but is still not up to the mark for the upcoming network requirements. Another thing is that there can be large number of isolated nodes in the cluster which are left out in the coupling process, remain active for whole network life time, hence consuming considerable amount of energy.

A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective. Approach used in developing this project is waterfall model. The following figure shows the various aspects of waterfall model approach of SDLC (Software Development Life Cycle).

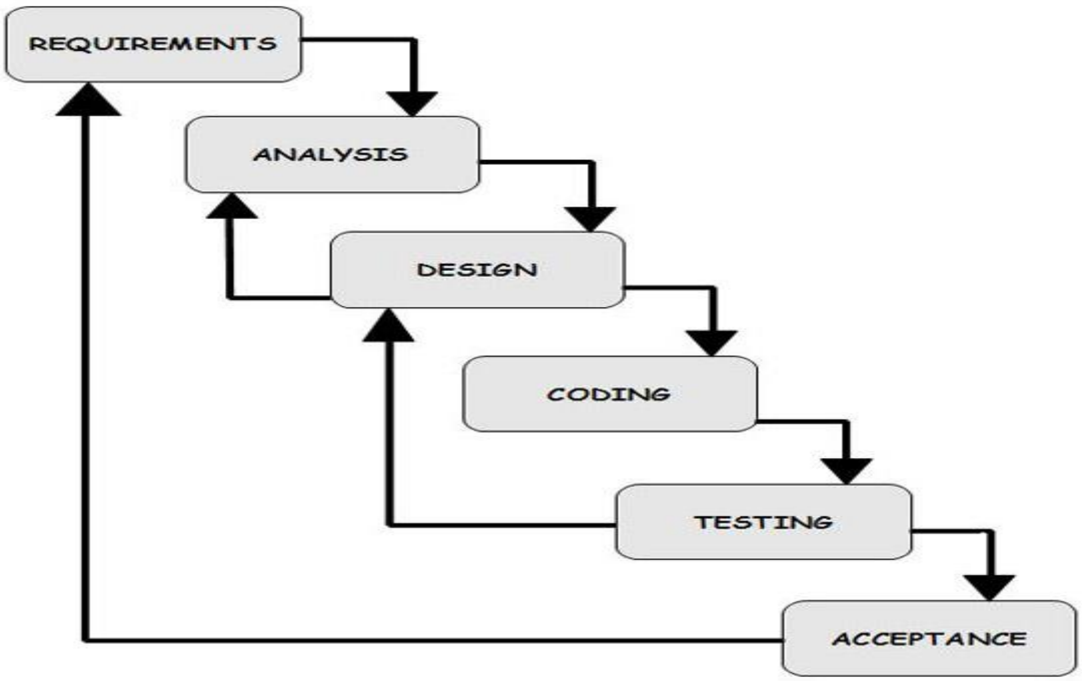


Figure 1.2: Software Development Life Cycle

1.7 Organization of Thesis

In Chapter 2, the literature survey has been discussed. This chapter, has looked upon the various parts of WSN and its applications in various fields from health monitoring to military services and from home security system to environmental system. AODV and EESAA have been discussed in detail and their algorithm has also been discussed. In Chapter 3 we have done feasibility study and requirement analysis. We have discussed the various designs like Data Flow Diagrams, Flow Chart, and Use Case etc. Next part includes coding which discusses the basic functioning, the working of clusters, etc.

In Chapter 4 we have thereby concluded our report.

CHAPTER 2: LITERATURE SURVEY

2.1 Introduction

A wireless sensor network consists 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. 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.

2.2 Parts of a WSN

While we associate a computer with a PC, the technical definition of a computer is a thing that computes, be it human or machine. Most sensors or WSN consist of five crucial components. These components include a number of sensors, such as temperature, moisture, and vibration sensors, a power source, in the case of older motes, 2 AA batteries, a radio transmitter/receiver, and an electric.

Sensors: When motes are under construction, their intended purpose often dictates the sensors that are added to the mote. The mote in Figure 2 contains three types of sensors: temperature, moisture, and vibration. This is a fairly typical mote, but some motes have many more functions. There are motes that take photographs of the surroundings, sense motion, measure light intensity, and much more. The sensors are attached to the mote base and communicate readings to the electronic brain.

Power Source: The power source for the mote also depends the mote’s intended use. If the mote is designed to last a very long time, say one year, it will have a larger power source than a mote that is only meant to run for a month. The power sources usually range between a couple of AA batteries, and a watch battery, but with the new smart-dust motes, also called “Spec,” they can collect enough energy to sustain themselves from ambient light, or even vibrations. The power source is connected to the mote base and provides energy required to run the sensors, electronic brain, and radio.

Radio: The radio consists of a radio transmitter and a radio receiver. Both of these parts must exist for any mote to fully communicate with the other motes. The radio, when transmitting, receives information from the electronic brain and broadcasts the data to other motes according to the network connections. In the other direction, when receiving, the radio receives information from another mote’s radio and transmits it to the electronic brain. The radio is connected to the mote base.

The Electronic Brain: The older motes’ brains consist of a microprocessor and some flash memory. Many of them have connectors to add other processes and sensors with ease. The MEMS motes also contain an analog-digital converter. The basic functions of the electronic brain are to make decisions and deal with collected data. The electronic brain stores collected data in its memory until enough information has been collected. Once this point is reached, the microprocessor portion of the electronic brain then puts the data in “envelopes,” or packages of data formatted for greatest transferring efficiency. These envelopes are then sent to the radio for broadcast. The brain also communicates with other motes to maintain the most effective network in much the same way it deals with data. The electronic brain is connected to the base and interacts with the sensors and radio.

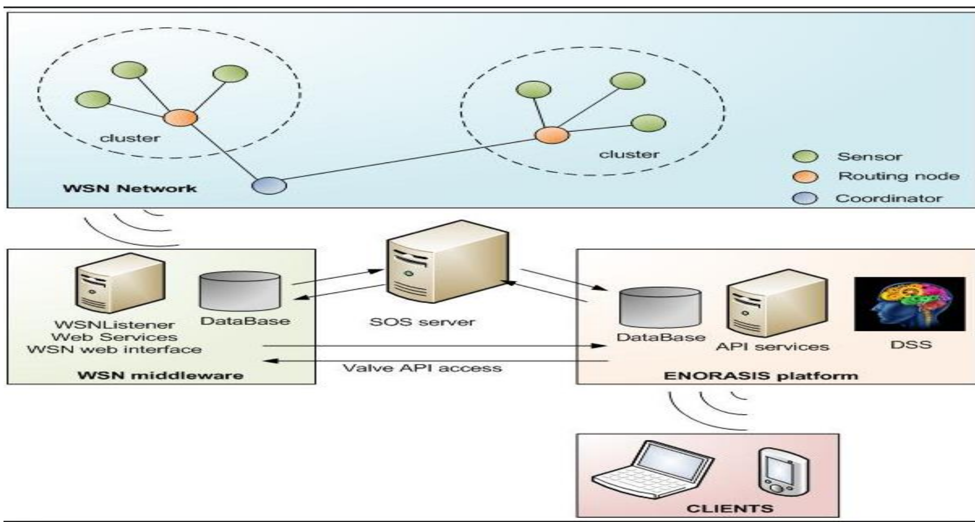


Figure 2.1: Wireless Network Sensor Components

2.3 Applications of WSN

- **Area monitoring**

  Area monitoring is a common application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. A military example is the use of sensors detect enemy intrusion; a civilian example is the geo-fencing of gas or oil pipelines.

- **Health care monitoring**

  The medical applications can be of two types: wearable and implanted. Wearable devices are used on the body surface of a human or just at close proximity of the user. The implantable medical devices are those that are inserted inside human body. There are many other applications too e.g. body position measurement and location of the person, overall monitoring of ill patients in hospitals and at homes.

- **Air pollution monitoring**

  Wireless sensor networks have been deployed in several cities (Stockholm, London and Brisbane) to monitor the concentration of dangerous gases for citizens. These can take advantage of the ad hoc wireless links rather than wired installations, which also make them more mobile for testing readings in different areas.

- **Forest fire detection**

  A network of Sensor Nodes can be installed in a forest to detect when a fire has started. The nodes can be equipped with sensors to measure temperature, humidity and gases which are produced by fire in the trees or vegetation. The early detection is crucial for a successful action of the firefighters; thanks to Wireless Sensor Networks, the fire brigade will be able to know when a fire is started and how it is spreading.

- **Landslide detection**

  A landslide detection system makes use of a wireless sensor network to detect the slight movements of soil and changes in various parameters that may occur before or during a landslide. Through the data gathered it may be possible to know the occurrence of landslides long before it actually happens.

- **Water quality monitoring**

  Water quality monitoring involves analyzing water properties in dams, rivers, lakes & oceans, as well as underground water reserves. The use of many wireless distributed sensors enables the creation of a more accurate map of the water status, and allows the permanent deployment of monitoring stations in locations of difficult access, without the need of manual data retrieval.

- **Natural disaster prevention**

  Wireless sensor networks can effectively act to prevent the consequences of natural disasters, like floods. Wireless nodes have successfully been deployed in rivers where changes of the water levels have to be monitored in real time.

- **Machine health monitoring**

  Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionality. In wired systems, the installation of enough sensors is often limited by the cost of wiring. Previously inaccessible locations, rotating machinery, hazardous or restricted areas, and mobile assets can now be reached with wireless sensors.

- **Data logging**

  Wireless sensor networks are also used for the collection of data for monitoring of environmental information, this can be as simple as the monitoring of the temperature in a fridge to the level of water in overflow tanks in nuclear power plants. The statistical information can then be used to show how systems have been working. The advantage of WSNs over conventional loggers is the "live" data feed that is possible.

2.4 Routing Protocols in WSN

A routing protocol defines about routers communication with each other, disseminating information that enables them to select routes between any two nodes on a computer network. Routing algorithms determine the specific choice of route. Every router has a prior information only of networks attached to it directly. They share this information first among immediate neighbors, and then throughout the network. This way, routers gain knowledge of the topology of the network. Different routing protocols are designed to circumvent the weaknesses of the resource constrained nature of the WSNs. WSN Routing Protocols can be classified in four main categories based upon:

- the type of communication routes processed within the network for data transmission from the source to sink
- the type of the network structure
- the network operations carried out using these protocols
- the initiator of communications

Proactive routing protocols: All the paths from sources to sinks are regularly computed before they are really needed and then these routes are stored in a routing table in each node so that routing information is kept for every node in the network. A certain amount of control traffic, is needed to keep routing tables up to date and consistent over the whole network.

Example: Optimized Link State Routing (OLSR).

Reactive routing protocols: Paths are acquired by nodes on demand when data need to be forwarded and no path to the destination is currently known. Whenever a sink wants to contact a particular node, the path values are calculated and the best path is selected for data transmission.

Example: Ad-hoc On-Demand Distance Vector (AODV).

Hybrid routing protocols: Combine features of proactive and reactive protocols. The network is divided into specified regions or zones. Data distribution within a zone is table driven (proactive) and when a node needs to send data to a node of another zone, it is accomplished through on-demand (reactive) routing protocol.

Examples: Dynamic Zone Topology Routing protocol (DZTR) and Zone Routing Protocol (ZRP).

Flat Routing protocols: All nodes participating in routing play the same role of collecting data and communicating with the sink.

Example: Sensor Protocols for Information via Negotiation (SPIN).

Hierarchical routing protocols: The goal of the protocol is to perform energy-efficient routing in WSNs by avoiding an overload of sink nodes by too many received messages, as well as reducing the amount of overall message transmissions. To achieve this, nodes are grouped into clusters, where the higher energy nodes are used to process and send the information while low energy nodes are used to perform the sensing in the proximity of the target.

Example: Low energy adaptive clustering hierarchy (LEACH).

Multipath routing protocols: Several paths are discovered between the source and the destination and are used to provide a backup route. When the primary path fails, the backup is used and this increases the network performance at the expense of increasing the cost of energy consumption and traffic generation.

Example: Ad hoc On-demand Multipath Distance Vector routing (AOMDV).

QoS based routing protocols

The network has to balance between energy consumption and data quality. In particular, the network has to satisfy certain QoS metrics (delay, energy, bandwidth, etc.) when delivering data to the base station.

Example: SPEED (Stateless Protocol for Real-Time Communication in Sensor Networks).

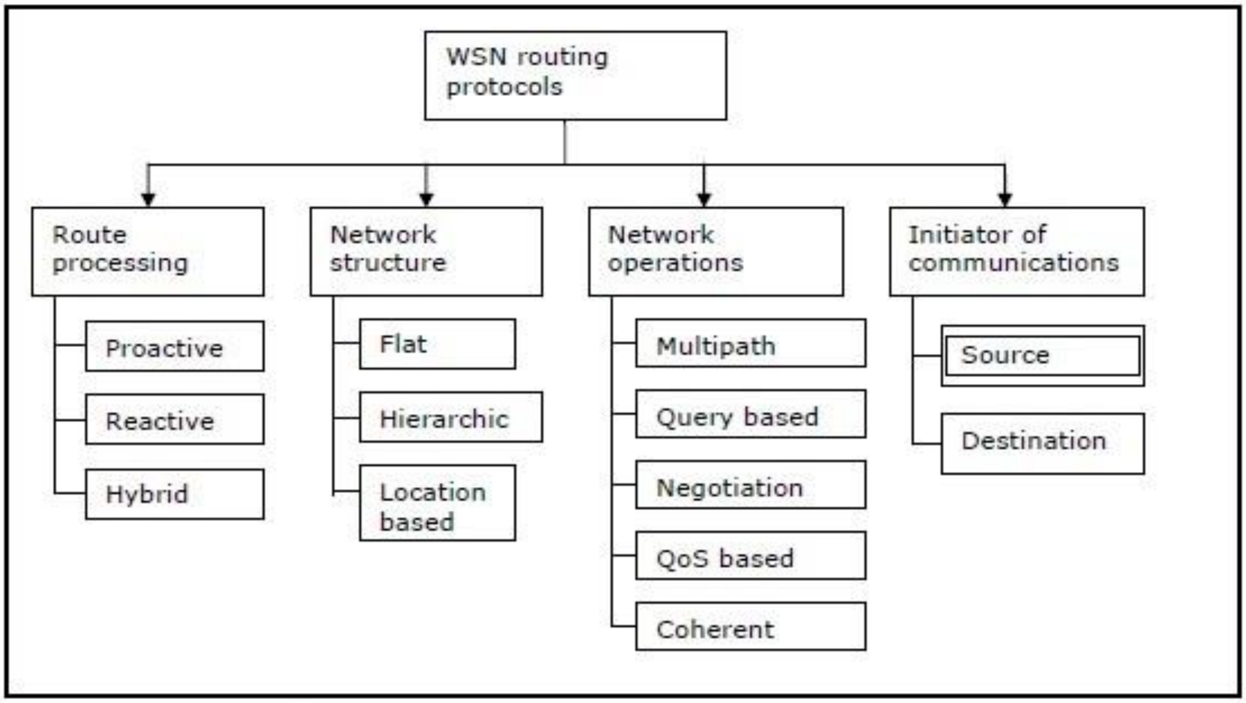


Figure 2.2: Routing protocol Types in WSN

2.5 Routing types in WSN

The two main types of routing: static routing and dynamic routing

The router learns about remote networks from neighbor routers or from an administrator. The router then builds a routing table, the creation of which I will explain in detail, that describes how to find the remote networks. If the network is directly connected then the router already knows how to get to the network. If the networks are not attached, the router must learn how to get to the remote network with either static routing (administrator manually enters the routes in the router's table) or dynamic routing (happens automatically using routing protocols).

The routers then update each other about all the networks they know. If a change occurs e.g a router goes down, the dynamic routing protocols automatically inform all routers about the change. If static routing is used, then the administrator has to update all changes into all routers and therefore no routing protocol is used.

Only dynamic routing uses routing protocols, which enable routers to:

- Dynamically discover and maintain routes
- Calculate routes

- Distribute routing updates to other routers
- Reach agreement with other routers about the network topology

Statically programmed routers are unable to discover routes, or send routing information to other routers. They send data over routes defined by the network Administrator.

A Stub network is so called because it is a dead end in the network. There is only one route in and one route out and, because of this, they can be reached using static routing, thus saving valuable bandwidth.

Dynamic Routing Protocols

There are 3 types of Dynamic routing protocols, these differ mainly in the way that they discover and make calculations about routes:

1. Distance Vector

2. Link State

3. Hybrid

2.6 AODV Routing Protocol

Reactive protocols seek to set up routes on-demand. If a node wants to initiate communication with a node to which it has no route, the routing protocol will try to establish such a route.

The philosophy in AODV, like all reactive protocols, is that topology information is only transmitted by nodes on-demand. When a node wishes to transmit traffic to a host to which it has no route, it will generate a route request (RREQ) message that will be flooded in a limited way to other nodes. This causes control traffic overhead to be dynamic and it will result in an initial delay when initiating such communication. A route is considered found when the RREQ message reaches either the destination itself, or an intermediate node with a valid route entry for the destination. For as long as a route exists between two endpoints, AODV remains passive. When the route becomes invalid or lost, AODV will again issue a request.

AODV avoids the ``counting to infinity" problem from the classical distance vector algorithm by using sequence numbers for every route. The counting to infinity problem is the situation where nodes update each other in a loop. Consider nodes A, B, C and D making up a MANET as illustrated in figure below.

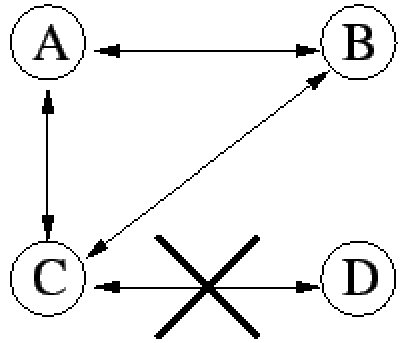


Figure 2.3**:** A scenario that can lead to the ``counting to infinity'' problem.

A is not updated on the fact that its route to D via C is broken. This means that A has a registered route, with a metric of 2, to D. C has registered that the link to D is down, so once node B is updated on the link breakage between C and D, it will calculate the shortest path to D to be via A using a metric of 3. C receives information that B can reach D in 3 hops and updates its metric to 4 hops. A then registers an update in hop-count for its route to D via C and updates the metric to 5. And so they continue to increment the metric in a loop.

The way this is avoided in AODV, for the example described, is by B noticing that A's route to D is old based on a sequence number. B will then discard the route and C will be the node with the most recent routing information by which B will update its routing table.

AODV defines three types of control messages for route maintenance:

RREQ - A route request message is transmitted by a node requiring a route to a node.

As an optimization AODV uses an expanding ring technique when flooding these messages. Every RREQ carries a time to live (TTL) value that states for how many hops this message should be forwarded. This value is set to a predefined value at the first transmission and increased at retransmissions. Retransmissions occur if no replies are received.

Data packets waiting to be transmitted(i.e. the packets that initiated the RREQ) should be buffered locally and transmitted by a FIFO principal when a route is set.

RREP - A route reply message is unicasted back to the originator of a RREQ if the receiver is either the node using the requested address, or it has a valid route to the requested address. The reason one can unicast the message back, is that every route forwarding a RREQ caches a route back to the originator.

RERR - Nodes monitor the link status of next hops in active routes. When a link breakage in an active route is detected, a RERR message is used to notify other nodes of the loss of the link. In order to enable this reporting mechanism, each node keeps a ``precursor list", containing the IP address for each its neighbors that are likely to use it as a next hop towards each destination.

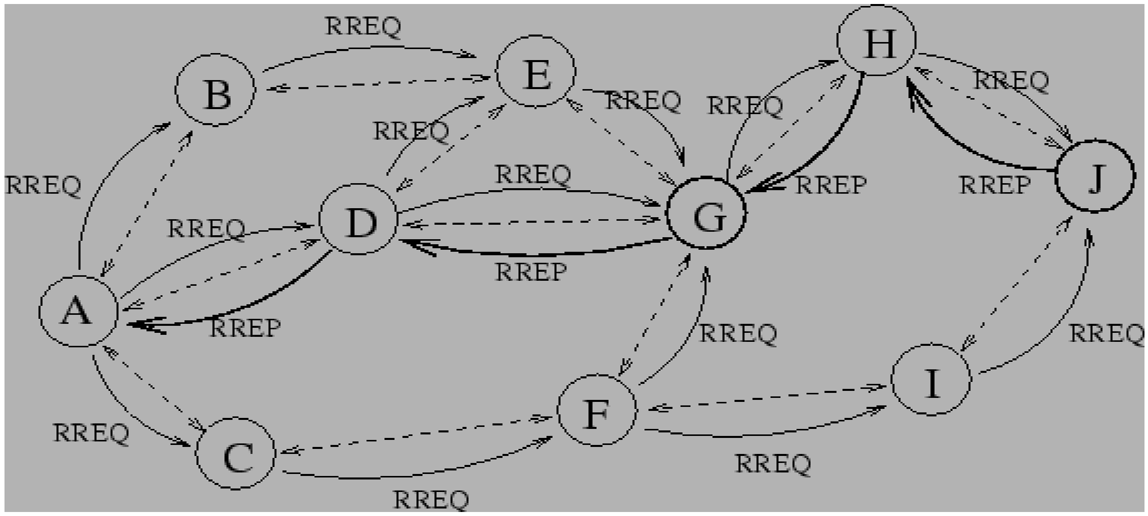


Figure 2.4: This figure illustrates an AODV route lookup session. Node A wishes to initiate traffic to node J for which it has no route. A broadcasts a RREQ which is flooded to all nodes in the network. When this request is forwarded to J from H, J generates a RREP. This RREP is then unicasted back to A using the cached entries in nodes H, G and D.

2.7 AAESS Routing Protocol

The goal of EESAA is to minimize the energy consumption by using the concept of pairing. Sensor nodes of same application and which are at the minimum distance between them will form a pair for data sensing and communication. This protocol will also use the Cluster Heads selection technique, by selecting CHs on basis of the remaining energy of the nodes. Information from the sensor nodes is forwarded to the cluster heads (CHs) and these CHs are responsible to transmit this information to the base station which is placed far away from the field.

According to this scheme, the nodes switch between "Sleep" and "Awake" modes during a particular communication Interval. Firstly node in a pair, switch into Awake mode also called "Active-mode". This happens if its distance from the sink is less than its coupled node. Node which is in "Active-mode" will gather data from surroundings and transmits it to the CHs. During this time period transceiver of the coupled node will remain "off" and switches into "Sleep-mode".

In the next step, nodes in "Active-mode" will switch into "Sleep-mode" and "Sleep-mode" nodes switches into "active-mode". By this process the consumption of energy can be minimized because nodes which are in "Sleep-modes" save their energy by not interacting with the CHs. "Unpaired nodes" remain in "Active-mode" for each round till their energy resources died. This method lessons the energy consumption but is still not up to the mark for the upcoming network requirements. Another thing is that there can be large number of isolated nodes in the cluster which are left out in the coupling process, remain active for whole network life time, hence consuming considerable amount of energy.

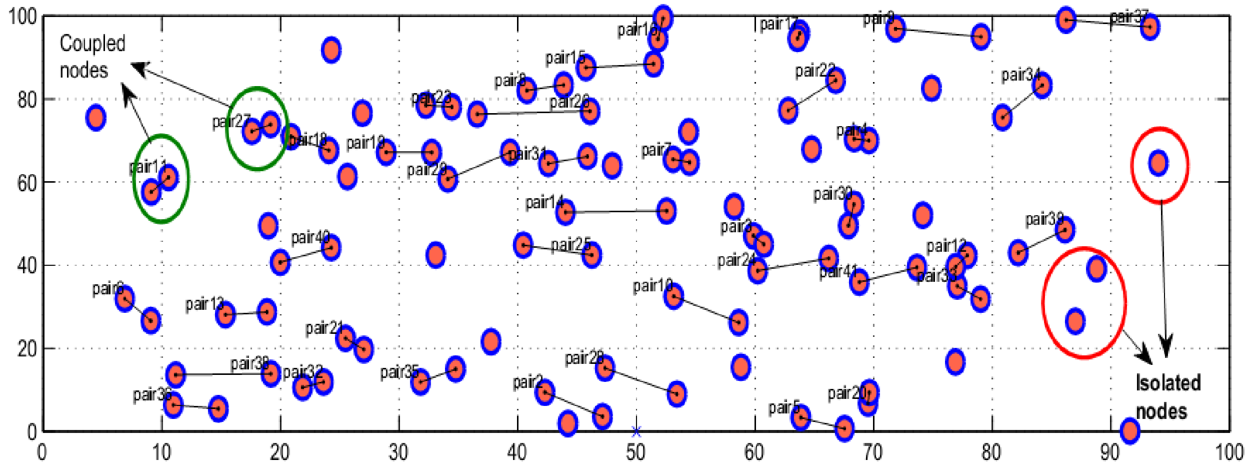


Figure 2.5

- Left circles – coupled or paired nodes
- Right circles – unpaired or isolated nodes

- **<u>CH selection mechanism:</u>**
    - Initially, in first round CHs are selected by same mechanism described by LEACH
    - CHs selection after first round is based on remaining energy of each node
    - Nodes in Active-mode take participation in CH election process
    - During start of round, nodes also transmit their energy information to CH
    - CH computes the remaining energy of every node in cluster and its distance from each node and select CH for the next upcoming round

- **<u>Transmission phase:</u>**
    - When node has been selected as CH, it broadcasts an advertisement message to whole network.
    - Only Active-mode nodes hear the broadcast advertisements from different CHs
    - They select their CHs on the basis of Received Signal Strength Indication (RSSI) of advertisements
    - Nodes in Active-mode, transmit their sensed data to CH during their assigned TDMA slots
    - Nodes in Sleep-mode do not take participation in data transmission and save their energy by turning their transceiver off

- **Node mode setup algorithm**

```
END OF ROUND
 if(node == coupled) then
      if (node_mode==active && CCH_FLAG==1) then
           node_mode=active
     else if (node_mode==active && CCH_FLAG==0) then
              node_mode=sleep
          else if (node_mode==sleep && CCH_FLAG==1) then
                 node_mode=sleep
                     else if (node_mode==sleep && CCH_FLAG==0) then
                        node_mode=active
end if
        else if (node==coupled&&node_neighbor==dead)then
              node_mode=active
        else
          node_mode=active
             end if
```

Chapter 3: Design and Implementation

3.1 Feasibility Study

"FEASIBILTY STUDY" is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. The feasibility of the project is analyzed in this phase. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed environment must be feasible. For feasibility analysis, some understanding of the major requirements for system is essential.

The key objective of the feasibility study is to weigh up two types of feasibility. They are:

- Operational feasibility
- Technical feasibility

3.1.1 Operational feasibility

Operational feasibility is necessary as it ensures that the project developed is a successful one. As the execution process of the proposed work is very much user friendly, the operational feasibility of the project is high.

3.1.2 Technical feasibility

Technical feasibility analysis makes a comparison between the level of technology available and that is needed for the project development of the project. The level of technology consists of the factors like software tools, machine environment, and platform developed and so on.

3.2 Requirement Analysis:

To create an environment where sensors are randomly deployed and sensors in the environment can detect the moving object in wireless sensor network.

3.2.1 Software Interface

Describes the connections between this product and other specific software components, including data bases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system into and going out and describe the purpose of each. Describe the services needed and the nature of the communications.

- Windows with windows 7 or 8 as OS.
- MATLAB version 7.10.0(R2010a).
- My SQL

3.2.2 Hardware requirements

Describes the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of data and control interactions between the software and hardware, communication protocols to be used.

Processor	: Pentium IV or any higher version
Memory	: 1GB RAM (Min)
Hard Disk	: 10GB (Min)
Input Device	: Keyboard, mouse and other peripherals

3.3 Assumptions

The following assumptions have been made in the development of our project:

- The sensor nodes are randomly placed in the field and are immobile thereafter.
- A battery is sole energy source of the sensor nodes.
- There is only one sink in the network, which is always active and has an infinite power supply.

3.4 Project Application Overview

Before any analysis can be performed, there are several steps that have to be taken to accurately generate the desired results. This section provides an overview of the various applications used.

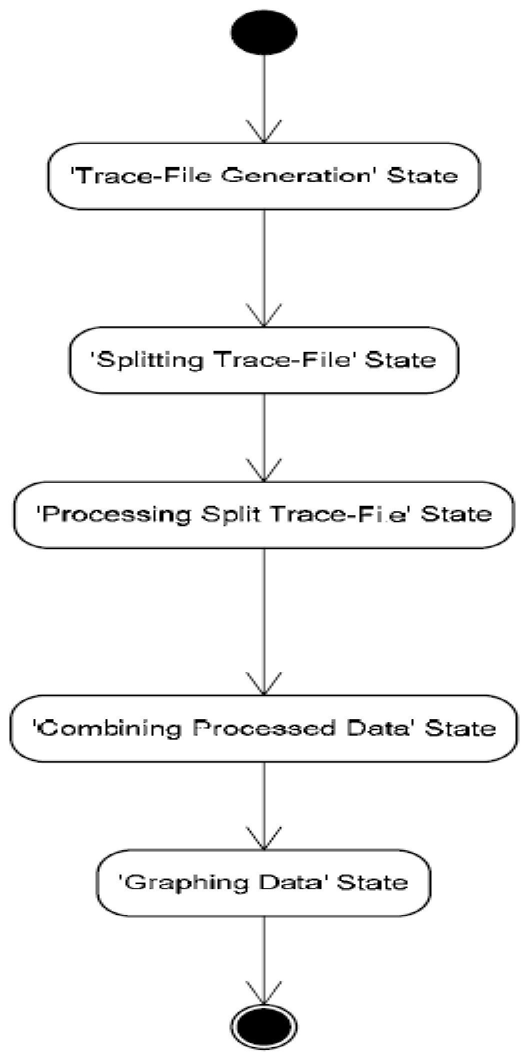


Figure 3.1: Steps undertaken before analysis

3.5 Block Diagram

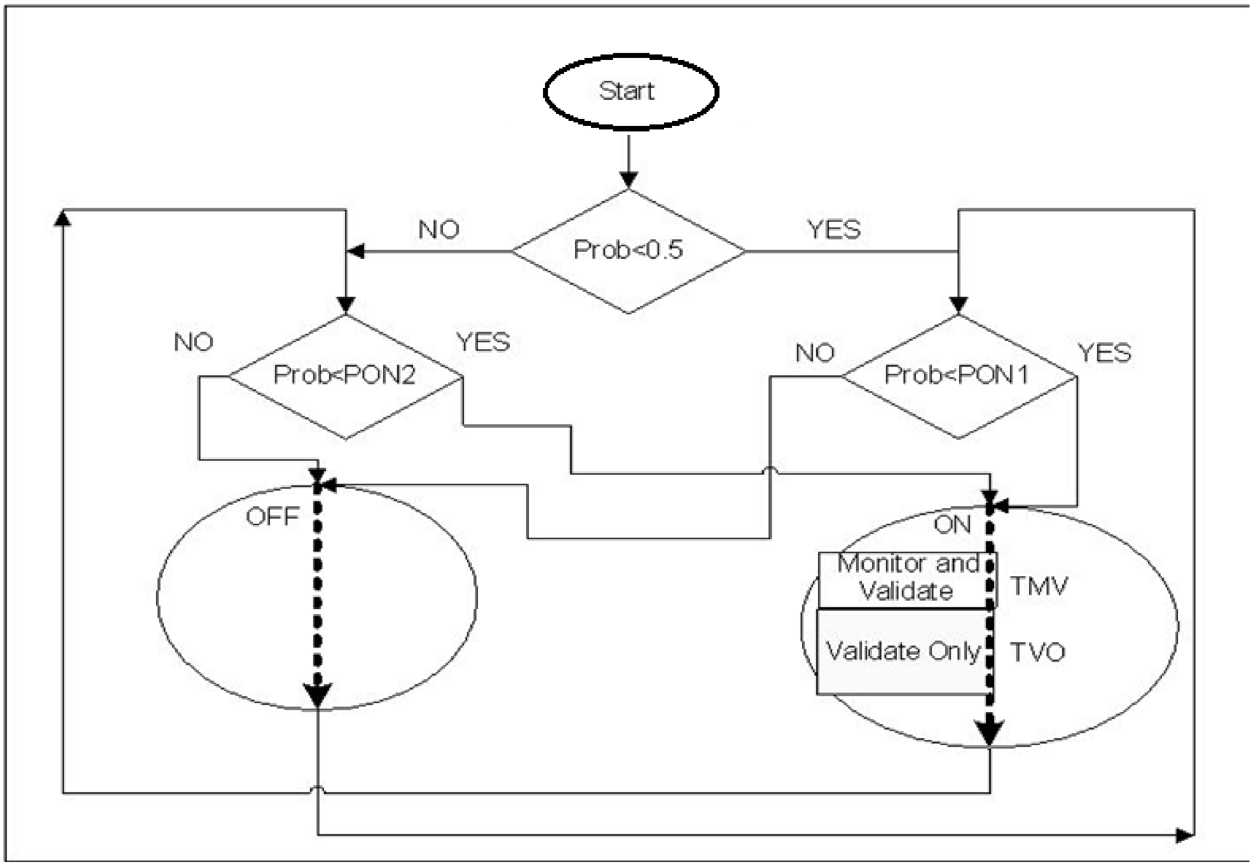


Figure 3.2: Block Diagram

In the above figure, block diagram is representing the steps of implementation with methodologies used during implementation.

3.6 Activity Diagram

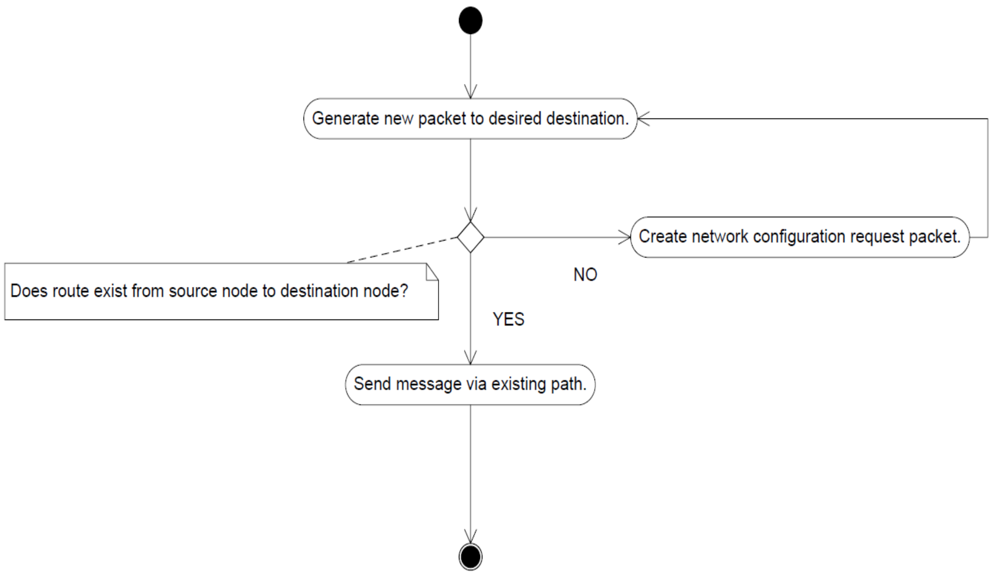


Figure 3.3: AODV Routing Protocol

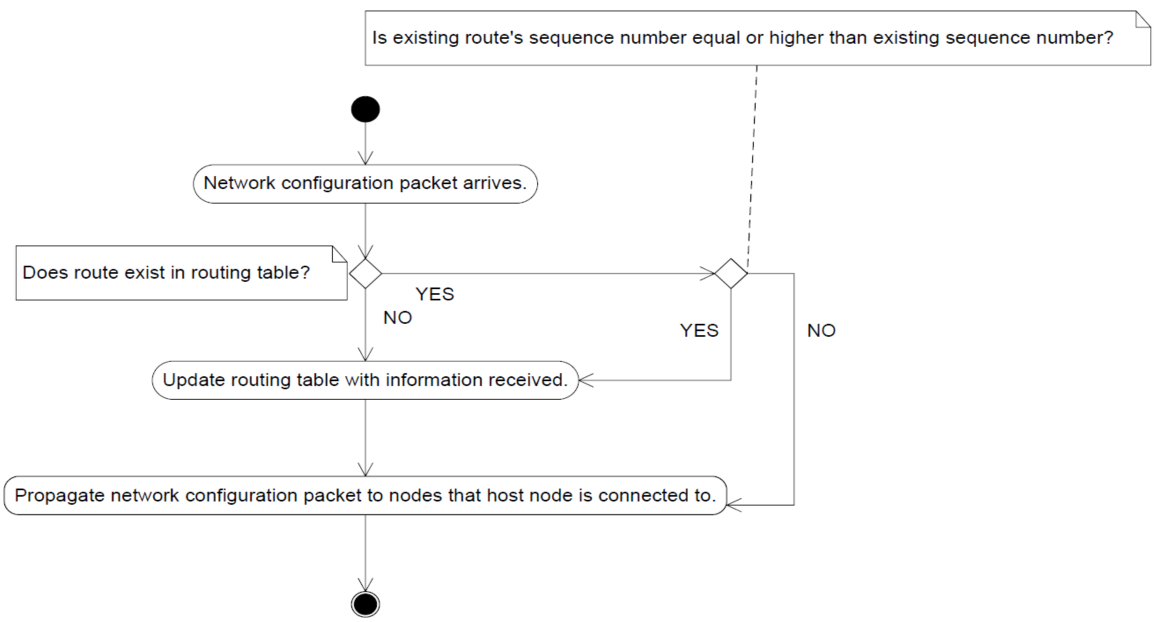


Figure 3.4: AODV network configuration packet

3.5 Collaboration Diagram

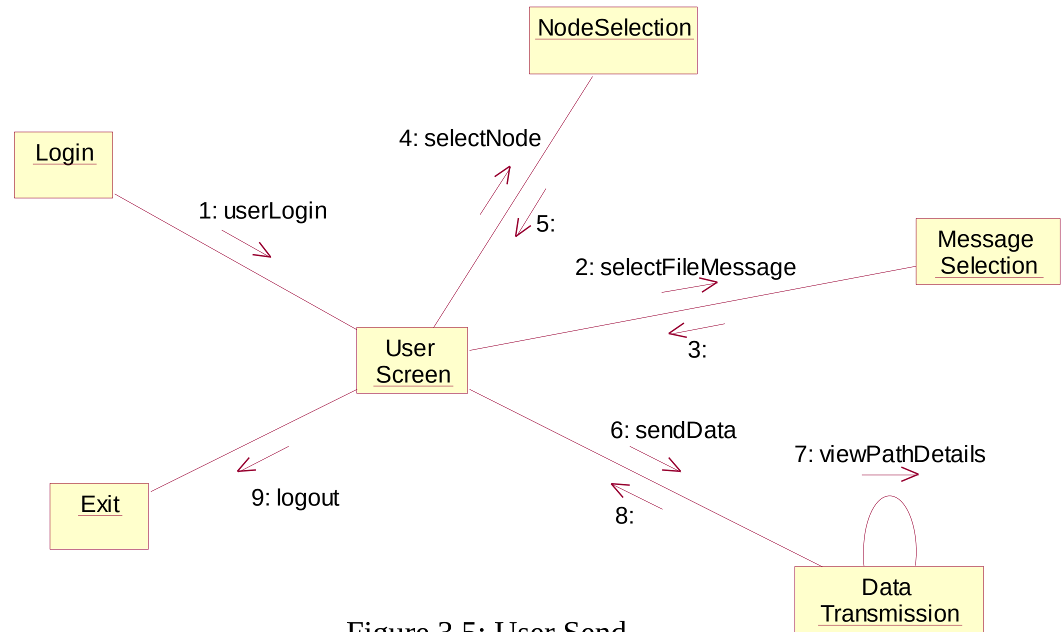


Figure 3.5: User Send

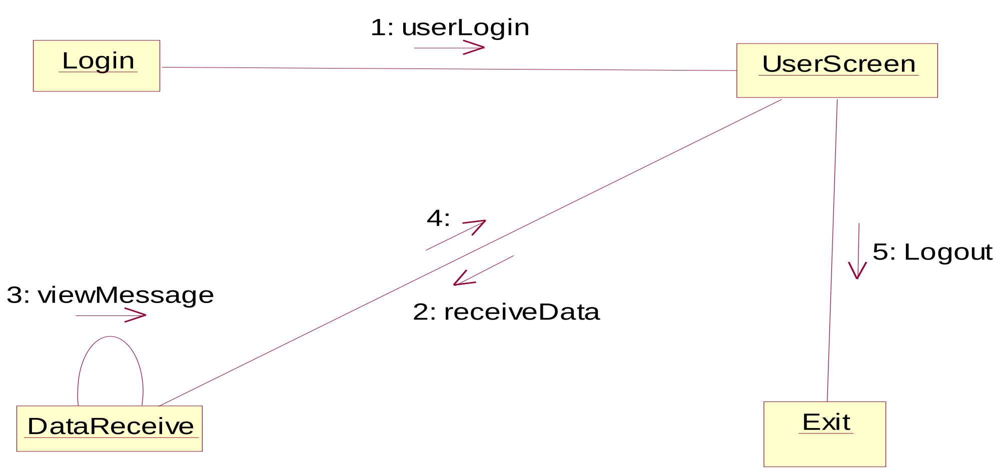
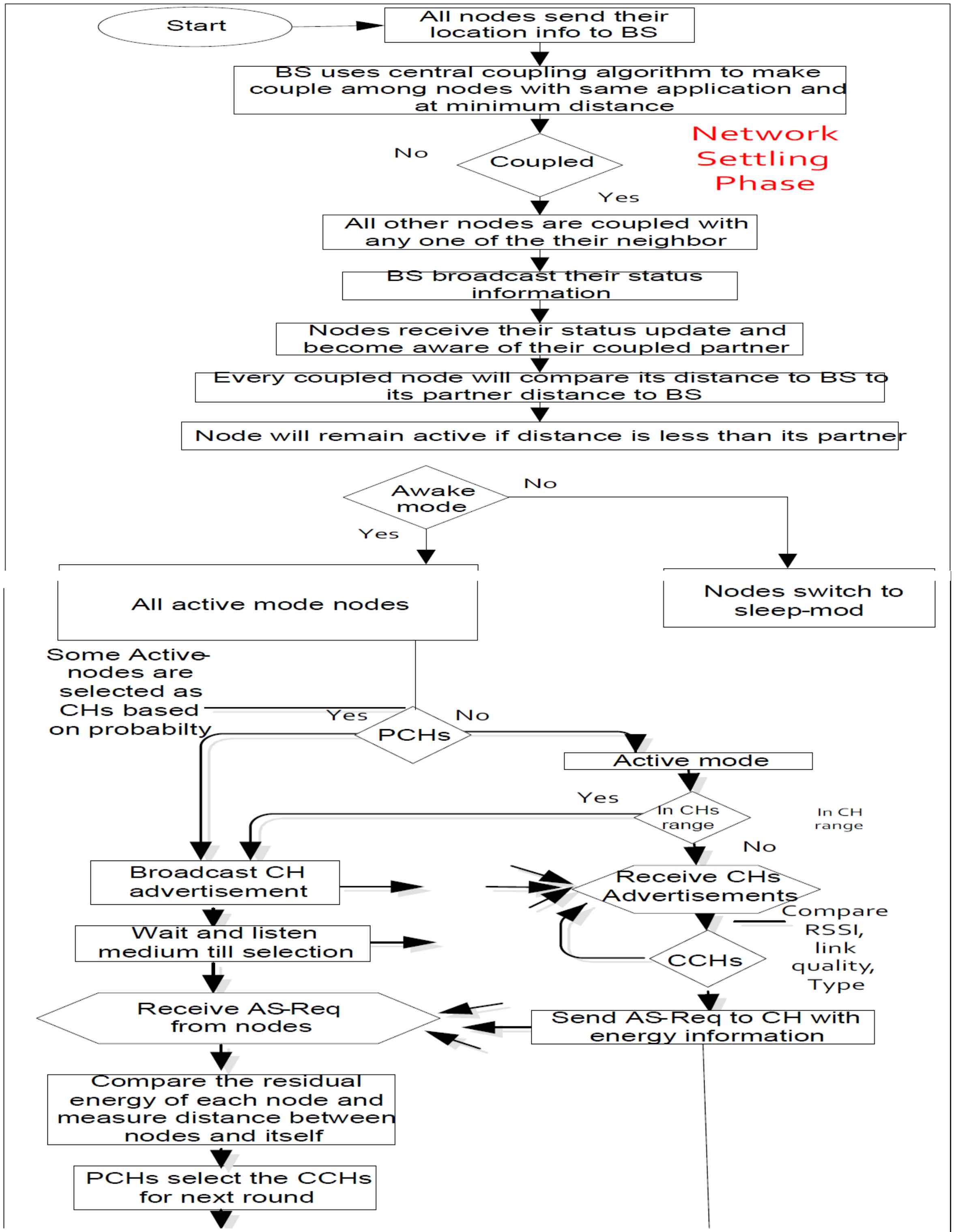


Figure 3.6: User Receive

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

3.6 EESAA Flow chart



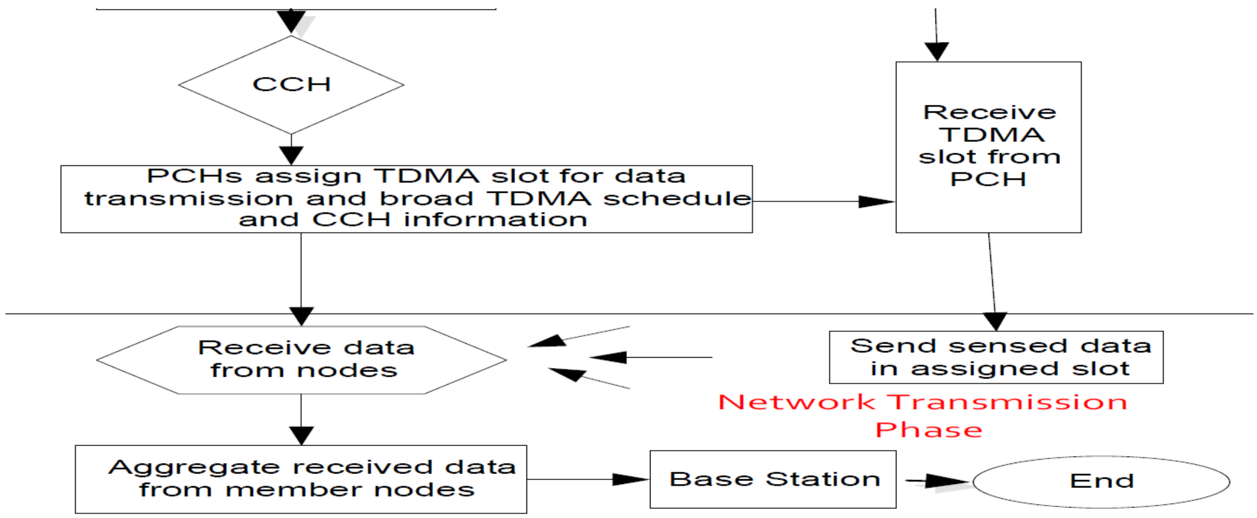


Figure 3.7

A flowchart is a formalized graphic representation of a logic sequence, work or manufacturing process, organization chart, or similar formalized structure. The purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process.

3.7 Flow Chart Of AODV

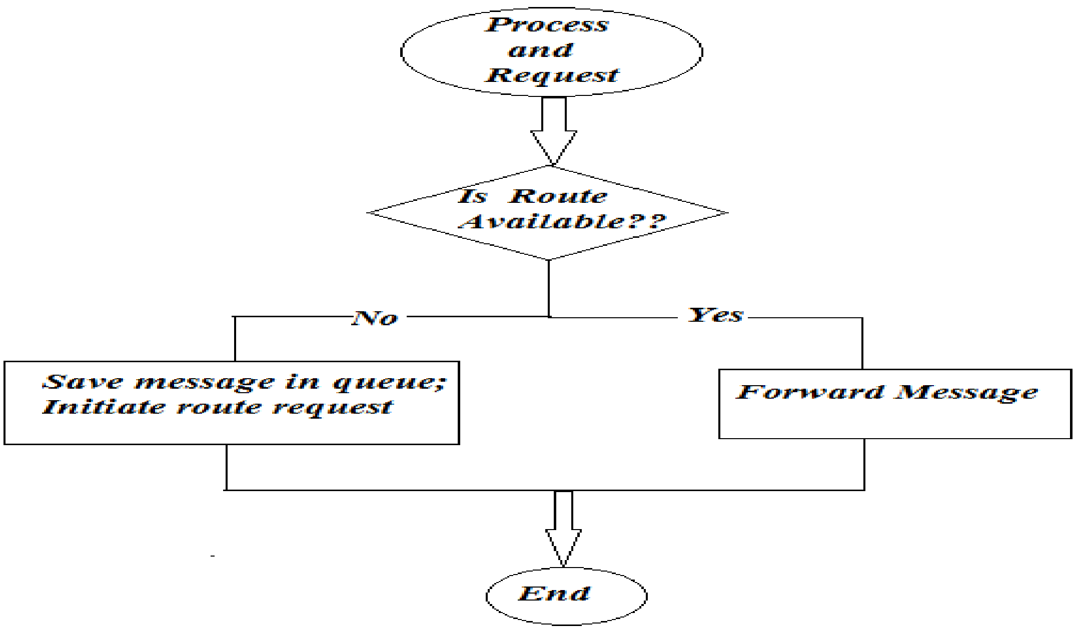


Figure 3.8: Flowchart Of Routing Protocol

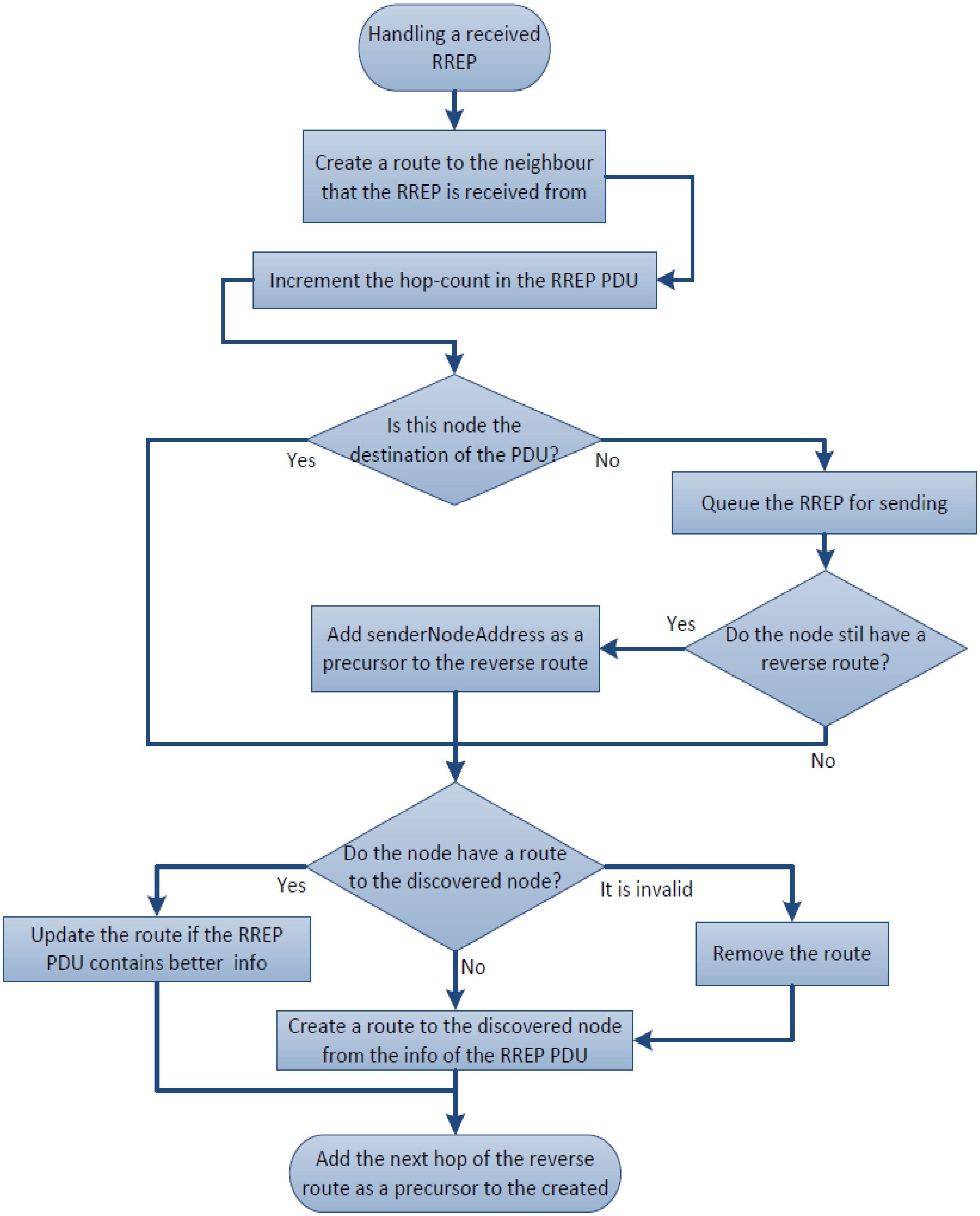


Figure 3.9: Flowchart showing how a received route reply is handled in AODV.

3.8 Data Flow Diagram:

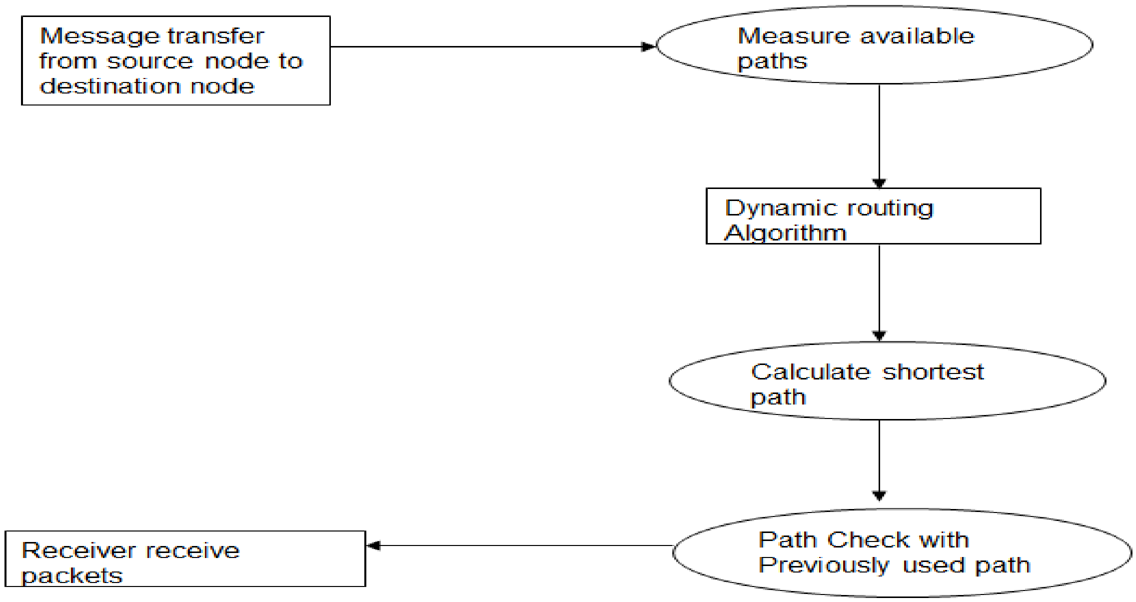


Figure 3.10: Data Flow Diagram

A data flow diagram is a graphical representation of the "flow" of data through an information system, modelling its process aspects. DFD can also be used for the visualization of data processing. Above diagram is showing what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

3.9 Simulation

Simulation of the routing protocols namely AODV and EESAA are being carried out in MATLAB. Simulation part contain two matlab files namely aodv.m and eesaa.m.

aodv.m

```
AODV Routing.

x=1:20;
s1=x(1);
d1=x(20);
clc;

A=randint(20);

% Making matrix all diagonals=0 and A(i,j)=A(j,i),i.e. A(1,4)=a(4,1),
% A(6,7)=A(7,6)

for i=1:20
    for j=1:20
        if i==j
          A(i,j)=0;
        else
          A(j,i)=A(i,j);
        end
    end
end

t=1:20;
disp(t);



 disp(A);

 status(1)='!';

% dist(1)=0;

dist(2)=0;
```

```
next(1)=0;

for i=2:20

    status(i)='?';

    dist(i)=A(i,1);

    next(i)=1;

  disp(['i== ' num2str(i) ' A(i,1)=' num2str(A(i,1)) ' dist(i)=' num2str(dist(i))]);

end

flag=0;
    for i=2:20
       if A(1,i)==1
           disp([' node 1 sends RREQ to node ' num2str(i)])
               if i==20 && A(1,i)==1
                      flag=1;
               end
       end
end

while(1)

   if flag==1
           break;
   end

   temp=0;
   for i=1:20
       if status(i)=='?'
           min=dist(i);

           vert=i;

           break;
       end
   end

   for i=1:20

       if min>dist(i) && status(i)=='?'
           min=dist(i);
           vert=i;
```

```
        end
    end
    status(vert)='!';

    for i=1:20
        if status()=='!'
            temp=temp+1;
        end
    end

    if temp==20
        break;
    end
end

i=20;
count=1;
route(count)=20;

while next(i) ~=1
    disp([' Node ' num2str(i) 'sends RREP message to node ' num2str(next(i))])
    i=next(i);
    %disp(i);

    count=count+1;

    route(count)=i;

    route(count)=i;

end

disp([ ' Node ' num2str(i) 'sends RREP to node 1'])

disp(' Node 1 to send message to node 20 since node 20 has send back the RREP ')

for i=count: -1:1

    disp([ ' Sends message to node ' num2str(route(i))])

end
```

b) eesaa.m

```
% INITIALIZATION
xm=100;
ym=100;
range=10;
sink.x=50;
sink.y=0;

%number of nodes
n = 100;
p=0.1;

%initial energy of network
Eo=0.5;

ETX=50*0.000000001;
ERX=50*0.000000001;

Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;


%Data Aggregation Energy
EDA=5*0.000000001;
a=1;

%number of rounds
        rmax=1;
        cluster_range=5;

                do=sqrt(Efs/Emp);
                figure(1);
                for i=1:1:n

           S(i).xd=rand(1,1)*xm;
           XR(i)=S(i).xd;
           S(i).yd=rand(1,1)*ym;
           YR(i)=S(i).yd;
           S(i).G=0;
           %S(i).E=Eo*(1+rand*a); %%%
         hetrogenous energy for normal nodes with different staritng nodes %

        % for homogenous energy distribution
          S(i).E=Eo
```

```
%initially there are no cluster heads only nodes
%defines type of node %
    S(i).type='N';    %%% defines type of node %%%%
    S(i).neighbour_flag=0;
    S(i).checked=0;
    S(i).id=i;
    S(i).CH_FLAG=1;
plot(S(i).xd,S(i).yd,'o','LineWidth',2,
'MarkerEdgeColor','b','MarkerFaceColor',[1 0.4 .3],'MarkerSize',10)
    grid on;
    hold on;
end

% BTS x coordinate
S(n+1).xd=sink.x;

% BTS y cooordinate
S(n+1).yd=sink.y ;

plot(S(n+1).xd,S(n+1).yd,'h','LineWidth',2,
'MarkerEdgeColor','b','MarkerFaceColor','g','MarkerSize',12)

countCHs=0;
cluster=1;
flag_first_dead=0;
flag_teenth_dead=0;
flag_all_dead=0;

dead=0;
first_dead=0;
teenth_dead=0;
all_dead=0;

allive=n;
%counter for bit transmitted to Bases Station and to
%Cluster Heads
packets_TO_BS=0;
packets_TO_CH=0;
temp_neighbour_distance=0;
x=1;
y=1;
figure(1);
```

```
%FORMATION OF PAIRS AMOUNG NODES %

for i=1:1:n
    neig_distance=Inf;
    if(S(i).E>0 && S(i).neighbour_flag==0)
        ID=0;
        for h=1:1:n
            if(S(h).E>0 && S(h).neighbour_flag==0)
                if(h~=i)
                    temp_neighbour_distance= sqrt( (S(i).xd-(S(h).xd) )^2 + (S(i).yd-(S(h).yd) )^2 );
                    if(temp_neighbour_distance<=range)
                        if(temp_neighbour_distance<neig_distance)
                            neig_distance=temp_neighbour_distance;
                            ID=h;
                        end
                    end
                end
            end
        end
        if (ID>0)
            S(i).neighbour=ID;
            S(i).mode='A';
            S(i).neighbour_flag=1;
            S(ID).neighbour_flag=1;
            S(ID).neighbour=i;
            S(ID).mode='S';
            P.pair(x)=ID;
            P.axis(x)=S(i).xd;
            x=x+1;
                              hold on;
        else
            S(i).neighbour=ID;
            S(i).mode='A';
            up.unpairs(y)=i;
            y=y+1;
                    plot(S(i).xd,S(i).yd,'o','LineWidth',2,        'MarkerEdgeColor','k','MarkerFaceColor',[1
0.4 .3],'MarkerSize',10)
        end
    end

end

hold on;

for i=1:1:n
    for j=1:1:n
```

```
        distance(i,j)=((S(i).xd-(S(j).xd))^2 + (S(i).yd-(S(j).yd) )^2 )^(1\2);
    end
end
figure(1);
for r=0:1:rmax
    r


    alive=n;
    dead=0;
    figure(1);
    for i=1:1:n

        if (S(i).E<=0)
            plot(S(i).xd,S(i).yd,'^','LineWidth',2,
'MarkerEdgeColor','r','MarkerFaceColor','r','MarkerSize',10)

dead=dead+1;
            if (dead==1)
                if(flag_first_dead==0)
                    first_dead=r
                    flag_first_dead=1;
                end
            end

            if(dead==0.1*n)
                if(flag_teenth_dead==0)
                    teenth_dead=r;
                    flag_teenth_dead=1;
                end
            end
            if(dead==n)
                if(flag_all_dead==0)
                    all_dead=r;
                    flag_all_dead=1;
                end
            end
        end

        if S(i).E>0
            S(i).type='N';

           plot(S(i).xd,S(i).yd,'o','LineWidth',2,
'MarkerEdgeColor','b','MarkerFaceColor',[1 0.4 .3],'MarkerSize',10)
            grid on;
            hold on;
```

```
            hold on;
        end

    end
    plot(S(n+1).xd,S(n+1).yd,'h','LineWidth',2,
'MarkerEdgeColor','b','MarkerFaceColor','g','MarkerSize',12)


    alive=alive-dead;
    STATISTICS.DEAD(r+1)= dead;
    STATISTICS.ALLIVE(r+1)=allive-dead;

    packets_TO_nextCH=0;
    countCHs=0;
    cluster=1;
    for i=1:1:n
        if(S(i).E>0)
            if(S(i).CH_FLAG==1 && S(i).mode=='A')
                if(cluster <= p*alive)
                    countCHs=countCHs+1;
                    plot(S(i).xd,S(i).yd,'o','LineWidth',2,
                        'MarkerEdgeColor',[0.8 1 0.02],'MarkerFaceColor','r','MarkerSize',10)
                    grid on;
                    hold on;


                    S(i).type='C';
                    S(i).G=round(1/p)-1;
                    C(cluster).xd=S(i).xd;
                    C(cluster).yd=S(i).yd;
                    dis=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );

                    C(cluster).distance=dis;
                    C(cluster).id=i;
                    X(cluster)=S(i).xd;
                    Y(cluster)=S(i).yd;
                    C(cluster).residual=0;
                    cluster=cluster+1;
                    S(i).CH_FLAG=0;
                    hold on;
                    packets_TO_BS=packets_TO_BS+1;
                    PACKETS_TO_BS(r+1)=packets_TO_BS;
```

```
            if (dis>do)
               S(i).E=S(i).E- ( (ETX+EDA)*(4000) + Emp*4000*(dis*dis*dis*dis ));
            end
            if (dis<=do)
               S(i).E=S(i).E- ( (ETX+EDA)*(4000)  + Efs*4000*(dis * dis ));
            end
            X10=[S(n+1).xd,S(i).xd]; Y10=[S(n+1).yd,S(i).yd];
            plot(X10,Y10,'-.k');

            hold on;
         end
      end
   end
end
STATISTICS.COUNTCHS(r+1)=countCHs;

%     for c=1:1:cluster-1
%        b=C(c).id;
%        dist=C(c).distance;
%        if (dist <= cluster_range)
%           packets_TO_BS=packets_TO_BS+1;
%           PACKETS_TO_BS(r+1)=packets_TO_BS;
%
%           dist=C(c).distance;
%           if (dist>do)
%              S(b).E=S(i).E- ( (ETX+EDA)*(4000) + Emp*4000*(dist*dist*dist*dist ));
%           end
%           if (dist<=do)
%              S(b).E=S(b).E- ( (ETX+EDA)*(4000)  + Efs*4000*(dist * dist ));
%           end
%           X10=[S(n+1).xd,C(c).xd]; Y10=[S(n+1).yd,C(c).yd];
%           plot(X10,Y10,'r');
%
%           hold on;
%
%
%        else
%           temp=4000;
%           for h=1:1:cluster-1
%
%              if (C(h).id~=b && C(h).distance < dist)
%
%                 if (distance(S(b).id,S(h).id)< temp)
%                    temp=distance(S(b).id,S(h).id);
%                    ca=h;
%
```

```
%                    end
%
%                end
%                min_dis=temp;
%            end
%            S(b).E=S(b).E- ( (ETX+EDA)*(4000) + Emp*4000*( min_dis* min_dis ));
%            packets_TO_nextCH=packets_TO_nextCH+1;
%            PACKETS_TO_nextCH(r+1)=packets_TO_nextCH;
%            X1=[C(c).xd,C(ca).xd]; Y1=[C(ca).yd,C(ca).yd];
%            plot(X1,Y1,'r');
%        end
%
%    end


for i=1:1:n
    resi=0;
    if ( S(i).type=='N' && S(i).E>0 )
        if (S(i).mode=='A')
            if(cluster-1>=1)
                min_dis=Inf;
                min_dis_cluster=0;
                for c=1:1:cluster-1
                    temp=min(min_dis,sqrt( (S(i).xd-C(c).xd)^2 + (S(i).yd-C(c).yd)^2 ) );
                    if ( temp<min_dis )
                        min_dis=temp;
                        min_dis_cluster=c;

                    end
                end
                min_dis;
                if (min_dis>do)
                    S(i).E=S(i).E- ( ETX*(4000) + Emp*4000*( min_dis *min_dis * min_dis *
min_dis));
                end
                if (min_dis<=do)
                    S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*( min_dis * min_dis));
                end
                resi=C(min_dis_cluster).residual;
                C(min_dis_cluster).nodeE(i+1)=S(i).E;
                if (S(i).E > resi)
                    C(min_dis_cluster).residual=S(i).E;
                    %                    N_CH(min_dis_cluster)=i;
                    N_CH=i;
                end
                S(N_CH).CH_FLAG=1;
```

```
            packets_TO_CH=packets_TO_CH+1;
            PACKETS_TO_BS(r+1)=packets_TO_CH;
            X10=[S(i).xd,C(min_dis_cluster).xd]; Y10=[S(i).yd,C(min_dis_cluster).yd];
            plot(X10,Y10,'r');
          end

      end
   end
   S(i).CH_FLAG=0;
end
%    for i=1:1:cluster-1;
%
%        ch_id=N_CH(i);
%        S(ch_id).CH_FLAG=1;
%
%    end

STATISTICS.PACKETS_TO_CH(r+1)=packets_TO_CH;
STATISTICS.PACKETS_TO_BS(r+1)=packets_TO_BS;
x=0;
%    for i=1:1:cluster-1
%        id=i;
%        if(S(i).E > 0 && S(i).mode=='A')
%            temp=S(i).E;
%            for h=1:1:n
%                if (h~=i && S(i).mode=='A')
%                if(S(h).E > 0 && S(h).checked==0)
%
%                   if (S(h).E <temp)
%                       temp=S(h).E;
%                       id=h;
%
%                   end
%                    end
%
%                end
%            end
%                S(id).CH_FLAG=1;
%                S(id).checked=1;
%                nid=S(id).neighbour;
%                S(nid).mode='A';
%
%        end
%
%    end
an=0;
```

```
    for i=1:1:n
        if (S(i).E>0)
        NID=S(i).neighbour;
        if(S(i).mode=='A' && S(i).CH_FLAG==0)
            S(i).mode='S';
        elseif(S(i).mode=='S' )
                S(i).mode='A';
        elseif (S(i).mode=='S' && S(NID).CH_FLAG==1)
                S(i).mode='S';
            an=an+1;
        end
        if(S(i).neighbour==0)
            S(i).mode='A';
            an=an+1;
        end

        if (S(i).neighbour~=0 && S(NID).E<=0)
            S(i).mode='A';
            an=an+1;
        end
        S(i).checked=0;

        end
    end
    hold off;

end

% first_dead
% teenth_dead
% all_dead
% STATISTICS.DEAD(r+1)
% STATISTICS.ALLIVE(r+1)
% STATISTICS.PACKETS_TO_CH(r+1)
% STATISTICS.PACKETS_TO_BS(r+1)
% STATISTICS.COUNTCHS(r+1)
% r=0:rmax;
% subplot(2,2,1);
% plot(STATISTICS.DEAD);
% subplot(2,2,2);
% plot(STATISTICS.ALLIVE);
% subplot(2,2,3);
% plot(STATISTICS.PACKETS_TO_BS);
% subplot(2,2,4);
% plot(STATISTICS.COUNTCHS);
```

3.10 Output

a) aodv.m

Command Window

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	0	0	1	0	0
0	0	0	1	0	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	0	1	1	1	1	0	0	1	1	1	0	1
1	1	0	0	1	1	0	1	1	0	1	0	1	0	1	1	1	0	0	1
1	0	0	1	0	0	1	1	1	1	0	0	1	1	1	1	0	1	0	0
1	1	1	1	0	0	1	0	1	1	1	1	0	0	0	1	1	1	1	1
1	1	1	0	1	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1
1	0	1	1	1	0	1	0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	0	1	1	1	1	0	0	1	0	0	0	1	1	0	1	0	0	1
1	0	1	0	1	1	1	1	1	0	0	0	0	0	1	0	0	1	0	0
0	1	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	1	1	0
1	1	1	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	0
0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	1	0	1	0	1
0	1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
0	1	0	1	1	0	0	0	1	1	0	1	1	0	0	0	1	0	0	0
0	1	1	1	1	1	0	1	0	0	0	1	1	0	0	0	0	0	1	1
0	1	1	1	0	1	0	1	1	0	0	1	0	0	1	0	0	0	1	0
1	0	1	0	1	1	1	0	0	1	1	1	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	1	1	0	1
0	0	1	1	0	1	1	0	1	0	0	0	1	0	0	1	0	1	1	0

Figure 3.11

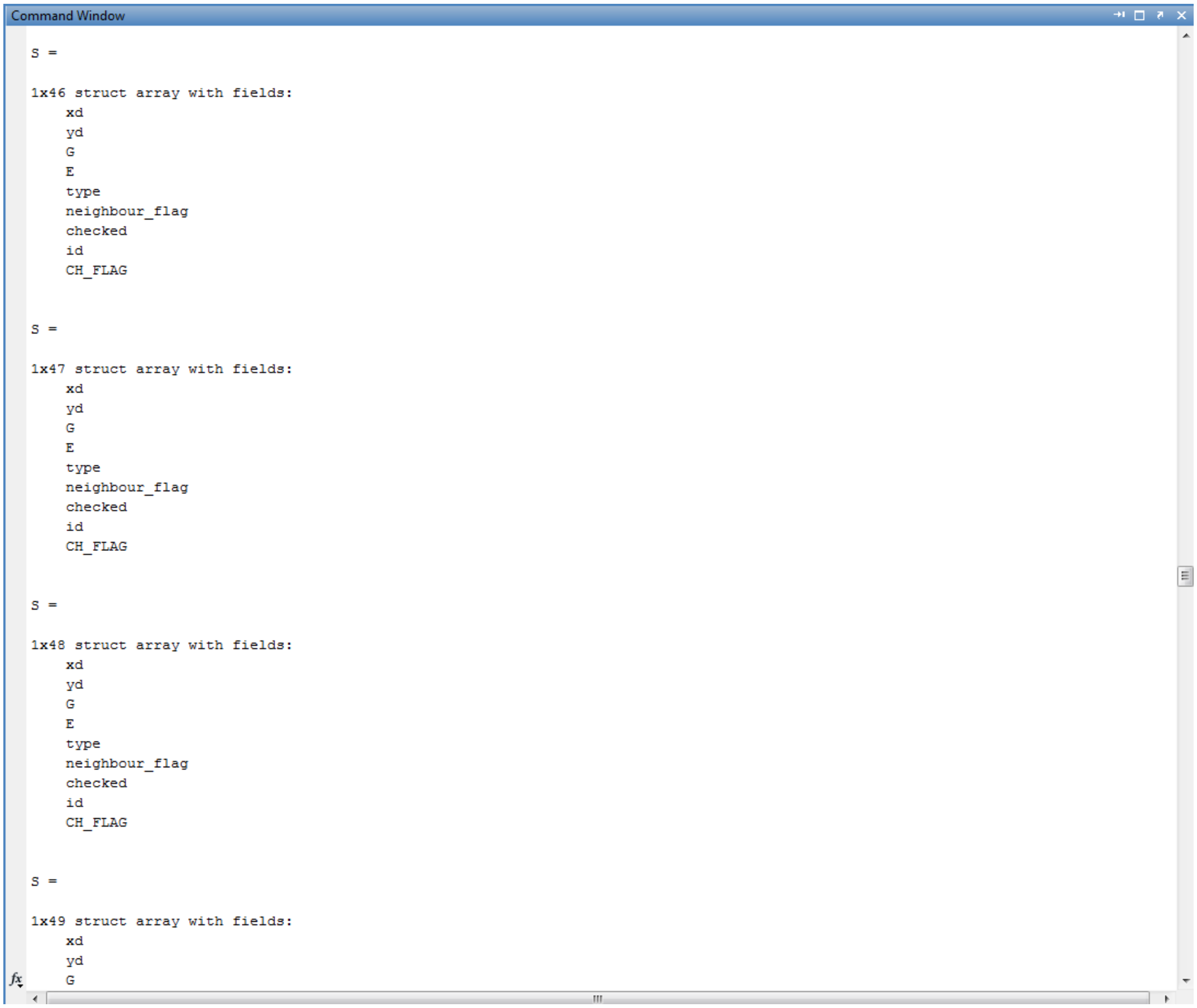
This figure shows the output of aodv protocol. In this output, matrix is formed of 20*20.
Here all the diagonal elements are zero.
All other elements are assigned either 1 or 0 with the probability of being 0 is .5.

```
i== 2 A(i,1)=0 dist(i)=0
i== 3 A(i,1)=0 dist(i)=0
i== 4 A(i,1)=1 dist(i)=1
i== 5 A(i,1)=1 dist(i)=1
i== 6 A(i,1)=1 dist(i)=1
i== 7 A(i,1)=1 dist(i)=1
i== 8 A(i,1)=1 dist(i)=1
i== 9 A(i,1)=0 dist(i)=0
i== 10 A(i,1)=1 dist(i)=1
i== 11 A(i,1)=0 dist(i)=0
i== 12 A(i,1)=1 dist(i)=1
i== 13 A(i,1)=0 dist(i)=0
i== 14 A(i,1)=0 dist(i)=0
i== 15 A(i,1)=0 dist(i)=0
i== 16 A(i,1)=0 dist(i)=0
i== 17 A(i,1)=0 dist(i)=0
i== 18 A(i,1)=1 dist(i)=1
i== 19 A(i,1)=0 dist(i)=0
i== 20 A(i,1)=0 dist(i)=0
 node 1 sends RREQ to node 4
 node 1 sends RREQ to node 5
 node 1 sends RREQ to node 6
 node 1 sends RREQ to node 7
 node 1 sends RREQ to node 8
 node 1 sends RREQ to node 10
 node 1 sends RREQ to node 12
 node 1 sends RREQ to node 18
 Node 20sends RREP to node 1
 Node 1 to send message to node 20 since node 20 has send back the RREP
 Sends message to node 20
>>
```

Figure 3.12

This figure shows the output of aodv protocol.
In this output figure, it is shown that source checks which path to follow and accordingly sends data to the destination via the best path. The nodes which earlier received the RREQ will neglect if it again receives and node which will receive RREQ for the first time will forward the data to its next neighbors. This will continue until destination receives RREQ and then it will send back RREP to source via the best path. Upon receiving RREP , source will send the data to destination via the route it receives the RREP.

b) eesaa.m



```
Command Window

s =

1x46 struct array with fields:
    xd
    yd
    G
    E
    type
    neighbour_flag
    checked
    id
    CH_FLAG


s =

1x47 struct array with fields:
    xd
    yd
    G
    E
    type
    neighbour_flag
    checked
    id
    CH_FLAG


s =

1x48 struct array with fields:
    xd
    yd
    G
    E
    type
    neighbour_flag
    checked
    id
    CH_FLAG


s =

1x49 struct array with fields:
    xd
    yd
    G
```

Figure 3.13

The above figure shows the output of EESAA routing protocol.
In the above figure, types of nodes and their properties are being displayed.

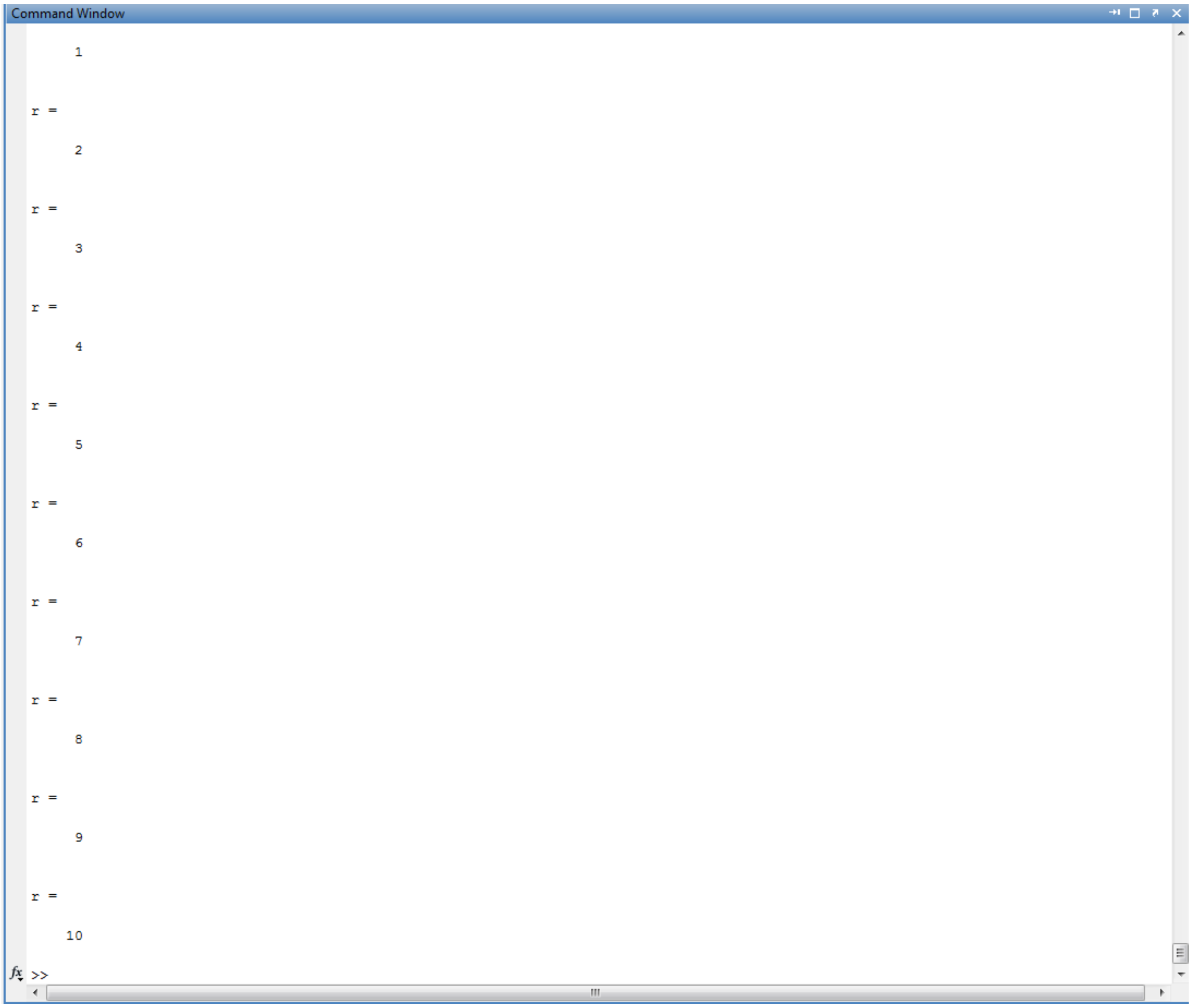


Figure 3.14

The above figure shows the number of rounds being carried out.

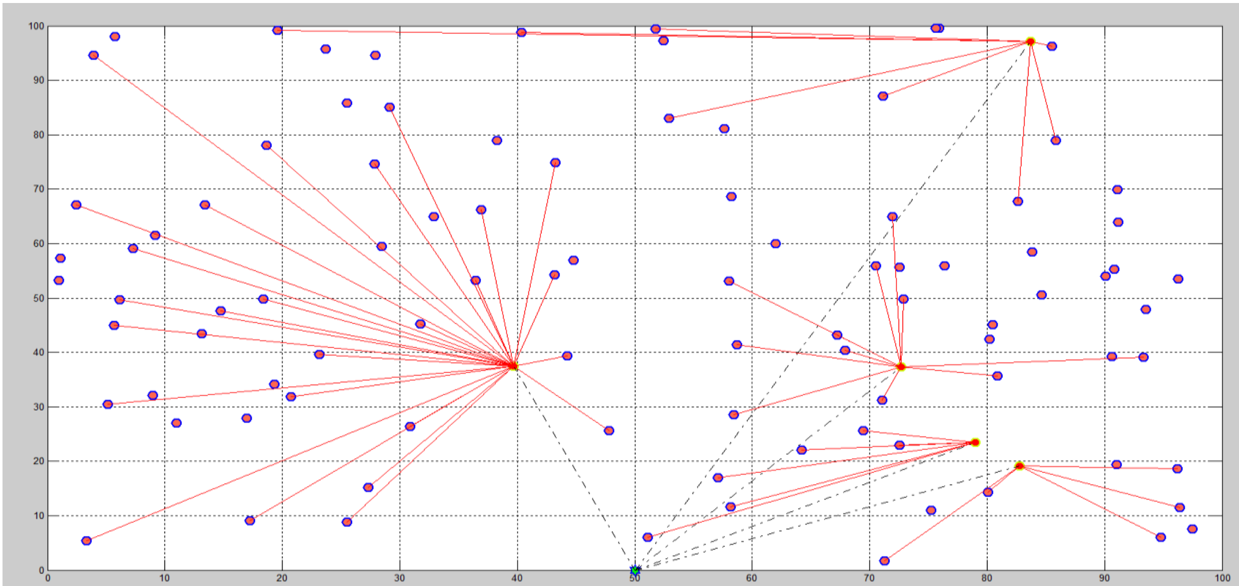


Figure 3.15

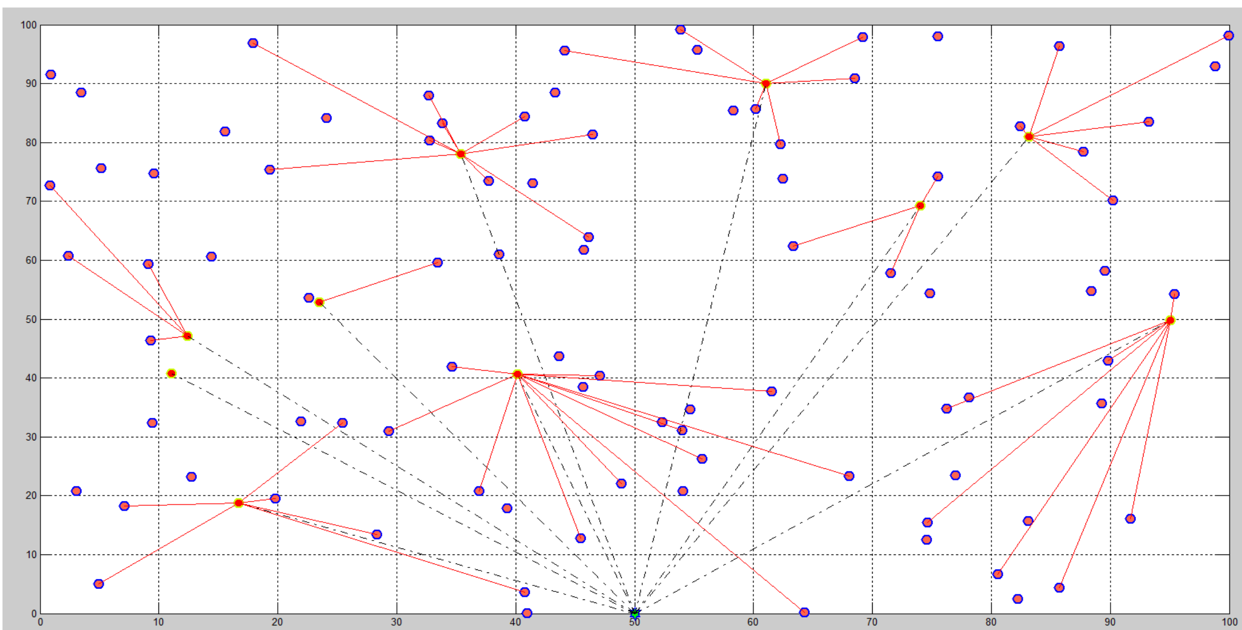


Figure 3.16

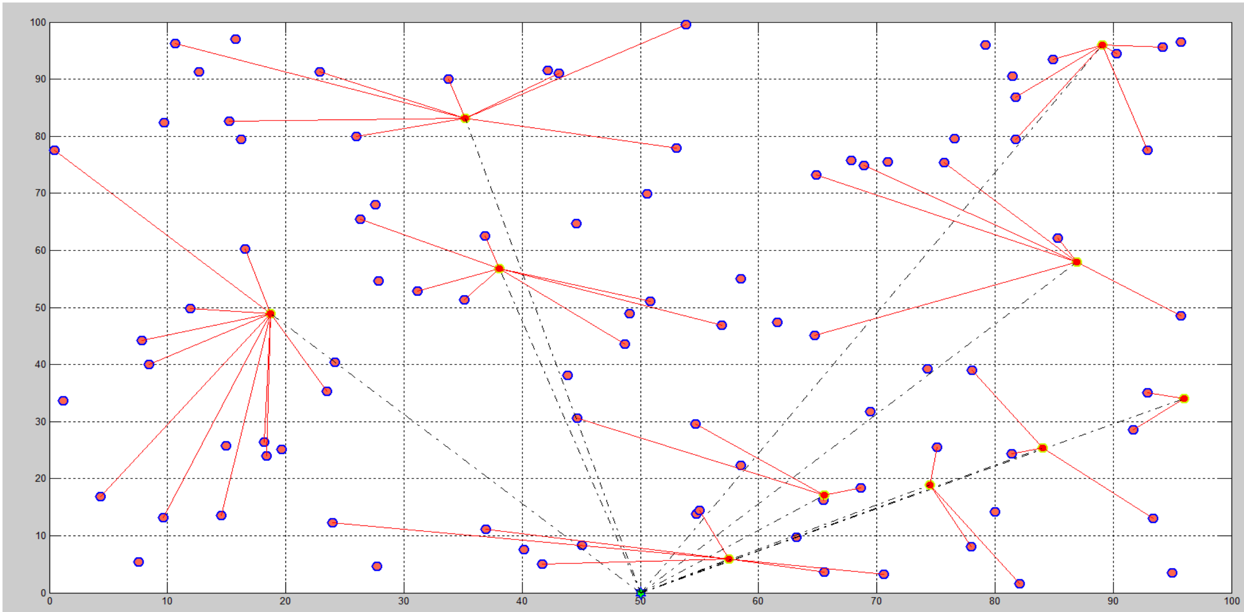


Figure 3.17

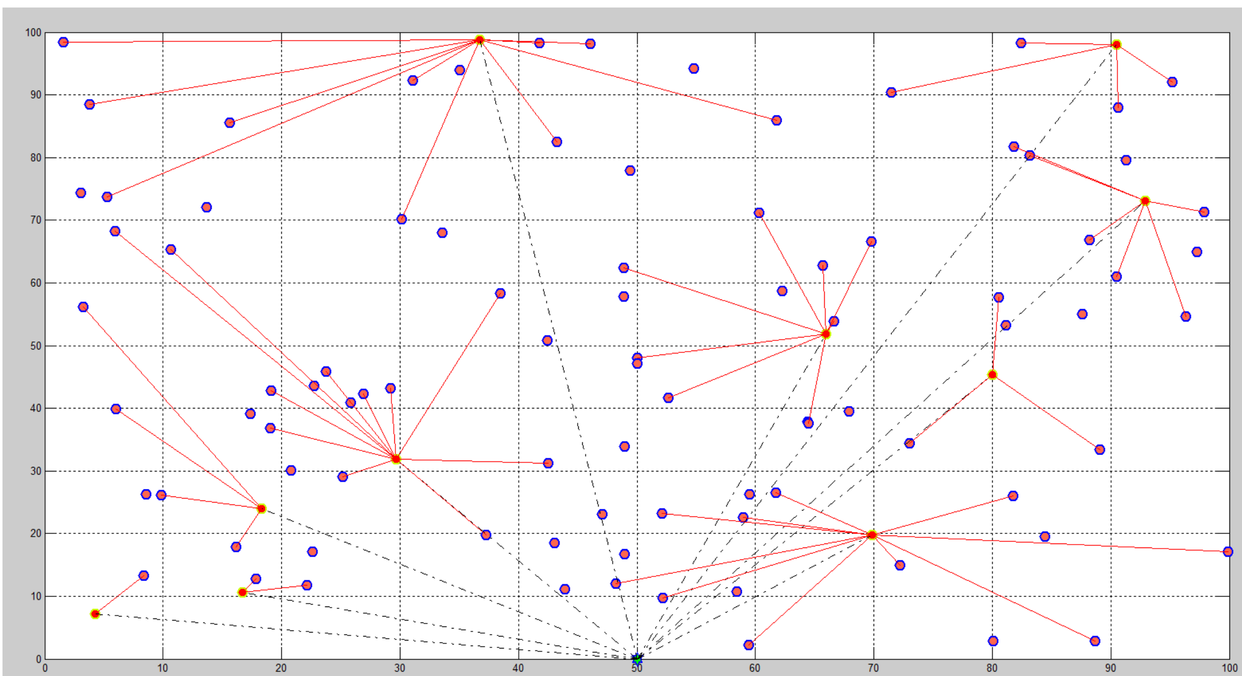


Figure 3.17

The above figures are the various outputs of EESAA routing protocol. In all the figures, pairing is followed. A pairing of nodes is done and CHs selection is done between them alternatively. The node other than CH, goes to sleep when its paired node becomes CH and wait for the next round to become the next CH. In this way they minimize the energy conservation and data is passed efficiently.

Chapter 4: Conclusion

Conclusion:

In existing system, when a source wants to send data to destination, it does not know the best path and sends data to all the nodes, leading to more energy consumption than required. The traditional routing algorithms make use of a centralized approach which are basically cover a single path. They even consume more bandwidth than required while sending data. Existing systems are not so reliable. Sometimes it is possible that data may crash in somewhere between the source and destination. Existing systems are not so secure. Taking into consideration that WSN applications need to support critical infrastructures (i.e. military, healthcare, environmental, etc.), security becomes an issue.

The two routing protocols namely AODV and EESAA routing protocols have been developed to overcome existing drawbacks. . AODV being a reactive protocol it still uses characteristics of a proactive protocol. Routes in AODV are discovered and established and maintained only when and as long as needed. To ensure loop freedom sequence numbers, which are created and updated by each node itself, are used. These allow also the nodes to select the most recent route to a given destination node. Whereas in EESAA, minimize the energy consumption by using the concept of pairing. The algorithm makes many improvements on the way of dividing clusters, strategy of electing the cluster head and construction method of data relay path, the two aspects of inter-cluster energy balance and energy balance among the cluster are taken into account at the same time.

References:-

- Perkins, C.; Belding-Royer, E.; Das, S. (July 2003). Ad hoc On-Demand Distance Vector (AODV) Routing. IETF. RFC 3561. Retrieved 2010-06-18.
- Energy Efficient Sleep Awake Aware (EESAA) Intelligent Sensor Network Routing ProtocolT. Shah, N. Javaid, T. N. Qureshi ,COMSATS Institute of Information Technology, 44000, Islamabad, Pakistan.
- Perkins, C.; Belding-Royer, E.; Das, S. (July 2003). Ad hoc On-Demand Distance Vector (AODV) Routing. IETF. RFC 3561. Retrieved 2010-06-18.
- S.-J. Lee and M. Gerla, “Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks,” Proc. ICC 2001, vol. 10, pp. 3201–3205, June 2001.
- W. Lou and Y. Fang, ""A Multipath Routing Approach for Secure Data Delivery,"" Proc. MILCOM 2001, vol. 2, pp. 1467–1473, Oct. 2001.
- FatemehDeldar and Mohammad HossienYaghmaee, “Energy Efficient Prediction-based Clustering Algorithm for Target Tracking in Wireless Sensor Networks”, International Conference on Intelligent Networking and Collaborative Systems,2010.
- SudakshinaDasgupta, Paramartha Dutta, “An Energy Efficient Genetic Approach for Clustering of Wireless Sensor Network”, International Journal of Information Engineering, Vol. 2, Iss. 2, PP. 54-58, Jun. 2012.
- Sunil Kumar Kashyap, K. R. Kambhatla, Bin Zan, Fei Hu and Yang Xiao, “An Energy-Aware and Intelligent Cluster-based Event Detection Scheme in Wireless Sensor Networks”, Internal Journal of Sensor Networks.
- Chuan-Ming Liu and Chuan-Hsiu Lee, “Distributed Algorithms for Energy-Efficient Cluster-head Election in Wireless Mobile Sensor Network”,Conf. on Wireless Networks (ICWN'05).
- Yun Xu1, Wei Bao and Hongli Xu, “An Algorithm for Continuous Object Tracking in WSNs.”,International Conference on Research Challenges in Computer Science,2009.
- Dan Liu and Nihong Wang Yi, “A Dynamic Cluster Based Object Tracking Algorithm in WSN”, Second WRI Global Congress on Intelligent Systems,2010.
- FatemehDeldar and Mohammad HossienYaghmaee, “Designing a Prediction-based Clustering Algorithm for Target Tracking in Wireless Sensor Networks”, International Symposium on Computer Networks and Distributed Systems,February 23-24,2011.

- Shaohua Liang, Md. ZakirulAlamBhuiyan and Guojun Wang, “Auction-Based Adaptive Sensor Activation Algorithm for Target Tracking in WSNs”, International Joint Conference of IEEE TrustCom-11/FCST-11,2011
- E. Cortese, F. Ouarta, and G. Vitaglione, “Scalability and Performance of the JADE Message Transport System,” *Proc. Of the AAMAS Workshop on AgentCities*, Bologna, Italy, July 2002
- Event Detection in Wireless Sensor Networks
  By - A V U Phani Kumar, AdiMallikarjuna Reddy V and D. Janakiram
- Dargie, W. and Poellabauer, C., "Fundamentals of wireless sensor networks: theory and practice", John Wiley and Sons, 2010 ISBN 978-0-470-99765-9, pp. 168–183, 191–192.
- E. Chen, *Jade and JXTA Extensions for Implementing a Better Distributed Systen*,Master’s Thesis, School of Engineering Science, Simon Fraser University, Canada, 2005.
- http://www.webopedia.com/TERM/H/hybrid_routing_protocol.html
- http://www.techopedia.com/definition/26311/hybrid-routing-protocol-hrp
- http://www.sciencedirect.com/science/article/pii/S1084804509001027
- Outlier Detection Techniques for Wireless Sensor Networks: A Survey
  By- Yang Zhang, Nirvana Meratnia, and Paul Havinga
- Coverage Problem in Wireless Sensor Network
  By- GaoJun Fan and ShiYaoJin
- http://www.ics.uci.edu/~dsm/ics280sensor/readings/intro/chong.pdf
- http://www.rimtengg.com/iscet/proceedings/pdfs/misc/176.pdf
- http://wiki.media-ulture.org.au/index.php/Main_Page
- http://www.sensor-networks.org/