Optimizing Power Consumption of Distributed Sensor Networks

Project Report

By

Nikhil Verma

Vaibhav Bhatia

[CMPE 295A]

Project Advisors

Leonard Wesley, Computer Engineering Department

Nadir Mir, Electrical Engineering Department

May 2013

ABSTRACT

Optimizing Power Consumption of Distributed Sensor Networks

By: Nikhil Verma and Vaibhav Bhatia

Distributed sensor networks (DSNs) typically consist of autonomous sensors linked to each other over a geographic area. A factor limiting the service life of a DSN is often each node's energy supply. Batteries typically provide energy source for nodes in a DSN. It is therefore desirable that sensors deployed for a specific application, such as fire-detection and monitoring in forest-like environments. The sensors should have as long as purposeful life at low cost as possible. Accomplishing this requires DSNs to use energy as efficiently, yet effectively, as possible. Currently, the LEACH-C algorithm has achieved the longest useful lifespan of DSNs. However, recent work by Wesley et al., has demonstrated the useful lifespan of DSN can be extended beyond that which LEACH-C can achieve by as much as 17% by taking into account specific application domain and environmental situational knowledge. This was accomplished, in part, by using the Evidential Reasoning (ER) calculus to reason about and optimizes the power consumption of node antennas.

Our work will extend this previous work by enhancing the computational infrastructure, needed to support the carrying out of NS2 simulations, of ER-based approaches to power management in DSNs. Upward of 3K nodes along with additional domain knowledge will be simulated and experiments conducted to further compare and evaluate the merits of an ER approach. The current user interface will be extended to facilitate configuring simulations in an intuitive manner. We believe working on this project will be highly beneficial to society and would be a big milestone in Distributed Sensor Network.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Project goals and objectives

The main goal of the project is to optimize power consumption of sensors in a DSN. The goal of the project is to be achieved through use of two algorithms Leach-C and evidential reasoning. The project will result in simulations showing improvement in power consumption. Low energy adaptive clustering hierarchy (LEACH) is adaptive cluster forming algorithm. It conservers energy based on its adaptive decision making capability. Evidential reasoning on the other hand takes environmental factors into consideration. Our objective is to simulate our design for at least 3000 nodes. We will be forming an interface, where in various environmental situations will be taken into consideration. Our interface will consist of input of various environmental scenarios. Based on this we will generate simulations to show in power consumption. Our project will have a database that will record all the scenarios and database. Also, we will implement an interface that will connect two algorithms.

We have a simulation project. As we don't have any DSN network, we will use NS2 simulator to give us the simulations. Through our project we track power consumption of each node of a DSN network. The simulations will be a result of using LEACH-C and Evidential reasoning algorithm on DSN nodes. The approach used is developing an interface to get simulations. The interface will be made up of components; hence we have a component-based methodology. The main components used a Leach-C algorithm, Evidential reasoning algorithm, environmental simulator and a database. The results will be displayed using NS2 simulator.

The project will be using C++ as the main programming language. But will be altered as per the needs of the project. The network simulator component, NS2 uses TCL script. The evidential reasoning module written in C language will be used to consider environmental conditions and alter working on nodes accordingly. The environmental simulator will take inputs from user and generate real-time data for the simulations.

The problem we face is to optimize power consumption of sensors in DSN. In order to this we aim at working on the basic functionality of sensors. Sensors are meant for computing, sensing and transmitting results. With the use of our algorithms we will regulate the working time of sensors. The sensors will be made to sleep or turn off when not needed.

The power consumption, while computing is being taken care by the central node. We don't want our sensors to be stressed while doing unnecessary sensing; hence through the use of evidential reasoning we calculate when to make our sensors go on sleep mode. The power consumption in transmission is dealt by use of LEACH-C algorithm. It computes an efficient way to transmit, by forming appropriate clusters.

Mainly a comparison of simulations using our components and without them gives the whole idea of our project. Further, our project can be evaluated and validated based on the three main modules: Evidential reasoning module, Environmental simulator module, and Network simulator. After we implement our project, we can evaluate the results by comparing it with raw data. Raw data is the data that doesn't use our method. The comparison between simulations is the best way to evaluate and validate our result. The results of two types of simulation will be included in our final report. Also, a comparison can be made between our own modules by using each module individually and showing how each module adds on to the results.

1.2 Problem and motivation

Distributed sensor networks, consists of small sensors connected over a network managed by a central node. These sensors are used to monitor and track activities in an area. The sensors usually have a limited battery life. Once, the battery dies the sensor is of no use. Manually, it is a difficult task to replace battery over and over in short duration of time. Hence, our project aims to reduce power consumption of these sensors so that they can perform for a long time, without the need for battery replacements.

The main motivation of this project comes from the scenario of DSN deployed in a forest for fire fighting purpose. Sensors are supposed to be cheap and deployed in a forest; hence method to conserve power should be feasible. Mechanically dealing with power consumption problem does not seem a viable option. Thus, our project aims at logically solving the problem through computational and algorithmic changes to the way sensors work.

The constant need of DSN and conserving power of sensor forms the basic need of our project. In a DSN, sensors are working all the time and their power depletes constantly. For sensors to have a long life it is important for them to somehow conserve power. If power is regularly and constantly depleted; the sensors would require quick replacements. It is hard to replace batteries for thousands of sensors in short period of time.

1.3 Project application and impact

Our project aims at saving power in DSN. The main application of our project is to conserve power of sensors in a DSN. We use environmental conditions and data to make our sensors sleep or switch off, when not needed. This saves a lot of power. We are using the evidential reasoning module for specific purpose of fire fighting in a forest. The evidential reasoning module capable of calculating environmental circumstances can further be used for other environment related purposes.

Our project gives sensors a longer life; hence replacement of battery is not needed often. This saves a lot of money and effort. Deploying sensors as per the requirement is a difficult and tiresome job. Our project helps save that time and energy.

Our projective is innovative and thus can server industry as a milestone. Our design serves DSN in a specific situation. It can further be enhanced and used under various circumstances.

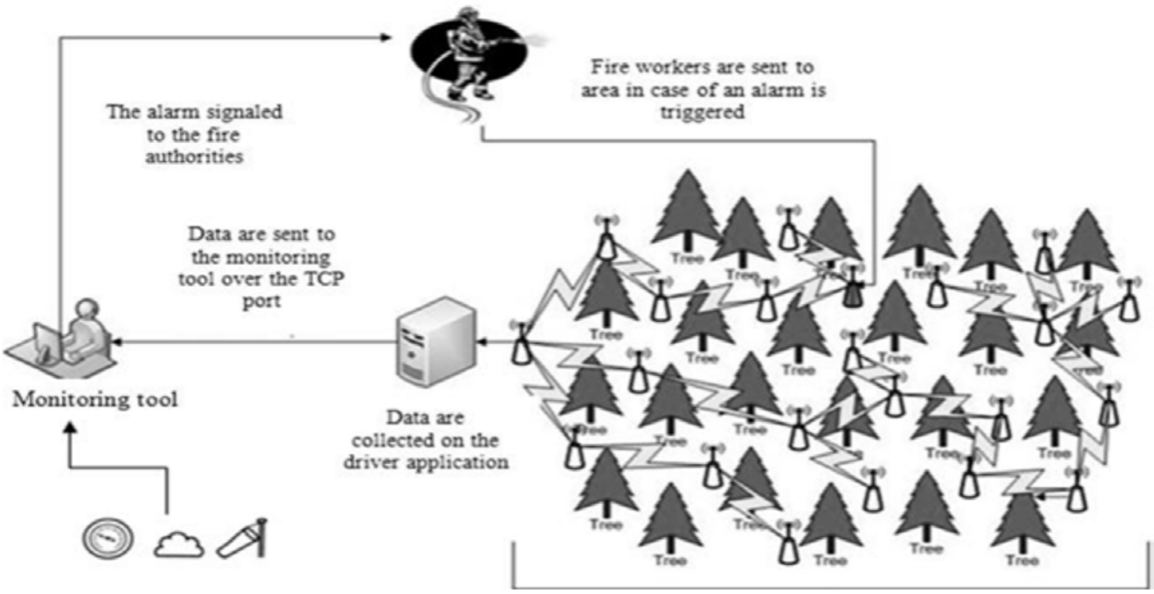

Figure1: Specific Project application: Fire Fighting in Forest

1.4 Project results and deliverables

The project is expected to be a system that will deploy LEACH-C and evidential reasoning algorithm on 3000 nodes of a DSN, to conserve their energy. The result of this application on sensors will be seen in the form of simulations from NS2 module. It will have a user interface in which the user can give environmental data to the system. The system will compute scenario and save power of nodes in a DSN. The project will yield a report of how the system is developed, and how it works. In addition we will have the results of power consumption data from 3000 nodes. The code to do all this will serve as a prototype. We also intend to show a comparison of simulations with and without the use of our algorithm.

1.5 Market Research

The focus of the project is non-power generating sensor nodes for monitoring the environment. The industry monitoring applications are the largest market of the wireless sensor nodes, which have constant power source. The focus of our project is in environment monitoring sensor to be deployed in remote area, deprived of power source.

Remotely deployed wireless sensor market focuses on the permanent or long term sensor deployment, having power generation. The price of sensor increases with power generation method, making its deployment in large scale expensive.

The project focuses on wireless sensor used in environment monitoring applications, having no power source. Short term monitoring applications are available in market such as disaster mitigation activity, where the sensor activity is for the short span of time due to constraint on the sensor's battery capacity. The focus of the project is to enhance the life of sensor and making its deployment affordable.

1.6 Project Report Structure

The report examines the art and technologies for wireless sensor network optimization in the current scenario. Then we present the plan and implementation of our project and the models and software used for the project

Chapter 2: Background and Related Work

In this chapter, we will provide all the information that is required to understand our project completely. This includes all the technologies used, the existing trend in digital sensor networks, its energy consumption algorithms and its literature survey.

2.1 Background and Used Technologies

In this project, we will be working with many technologies throughout the year. These include:

- Scripting language PHP and oTCL
- Programming language C++, HTML
- Distributed Sensor Networks
- A Network Simulator NS2
- Some algorithms and protocols like LEACH-C

Some of these technologies are discussed in the following sections

2.1.1 PHP, oTCL and C++

PHP is a server side scripting language used for web development. In this project we will be designing a better interface for the users to run their simulations. This interface will include all the value columns that are necessary to run a simulation. This includes environmental conditions, number of sensors to be deployed, etc. All these pages will be designed using HTML and CSS. This will form the front end of the website and the backend will be done in PHP.

oTCL and C++ are the two languages which are used to write NS2 simulations. oTCL is the object oriented TCL. This was developed by MIT. oTCL is used to create a network so that user doesn’t have to change a network for every change in simulation scenario [9]. Users are required to write oTCL scripts which includes all the conditions required to run the simulation. These include network topology, environmental conditions and event scheduling information. NS2 contains oTCL interpreter which takes oTCL scripts as an input from user and interprets these scripts without compilation. This makes the execution slow because interpretation is done on runtime but in this case user can make changes to the script easily. oTCL also links various blocks inside NS2 components.

C++ is the other language which is used to write NS2 simulations. It defines each block inside an NS2 component. User can write the detailed protocol description including network component configuration, time scheduling and various environmental events. Before the execution of C++ codes, they need to be compiled to machine code. This makes the execution faster than the execution of oTCL scripts but changing the code is difficult as user have to recompile the code for every change they make. After compilation, oTCL interpreter uses oTCL objects which are created by oTCL linkage based on C++ objects

2.1.2 Distributed Sensor Networks

A distributed sensor network is a wireless network distributed over a geographical area which contains one or more base stations connected to thousands of nodes. These base stations link nodes to the data collection center and nodes sense the environment, collect the related information and send the data to the base station.

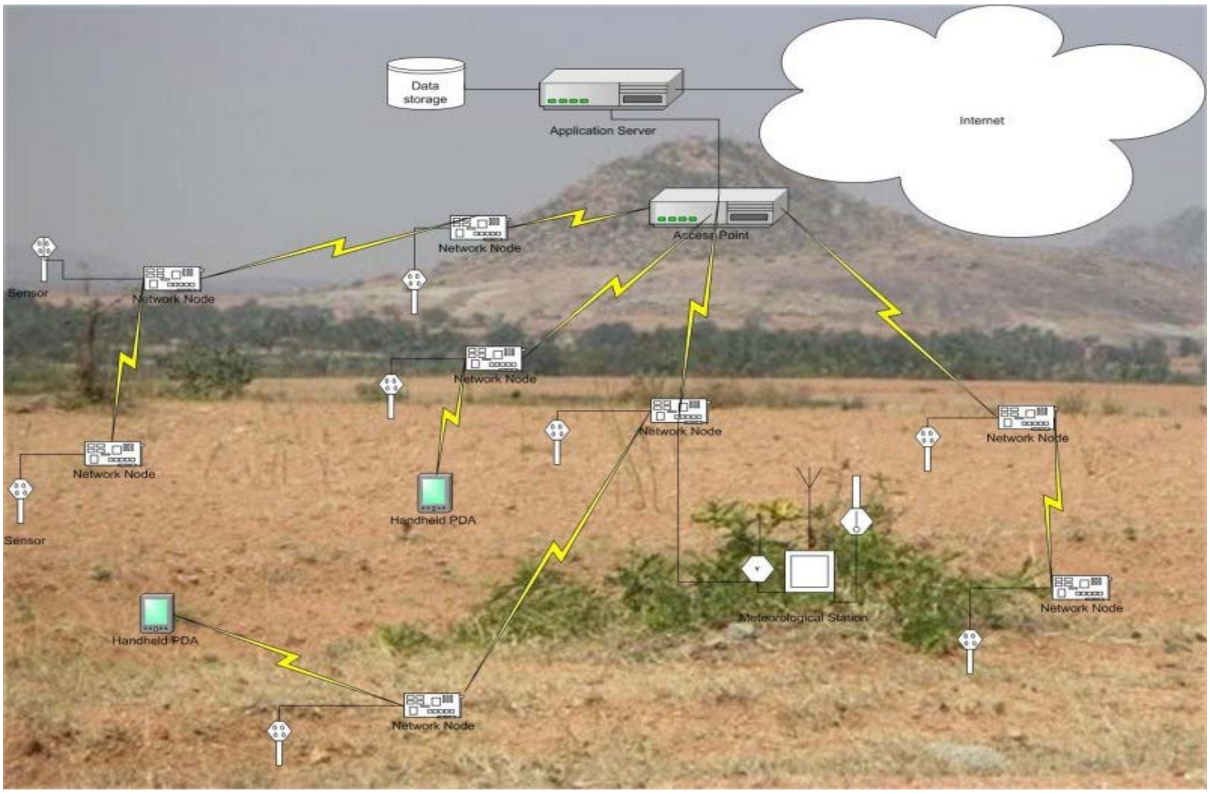

Figure 2: Major Components of DSN [23]

There are many factors which influence the design of DSN, like power consumption, communication, hardware, scalability, transmission media, fault tolerance, production costs, operating environment, network topology and communication. In this project we will only concentrate on three factors i.e. power consumption, communication and operating environment [1]. The main aim is to decrease the power consumption of the nodes by decreasing the communication between the nodes depending upon the operating environmental conditions.

The main components of a sensor node are:

- A Sensor
- A Data Computation Unit
- An Energy Source like a battery

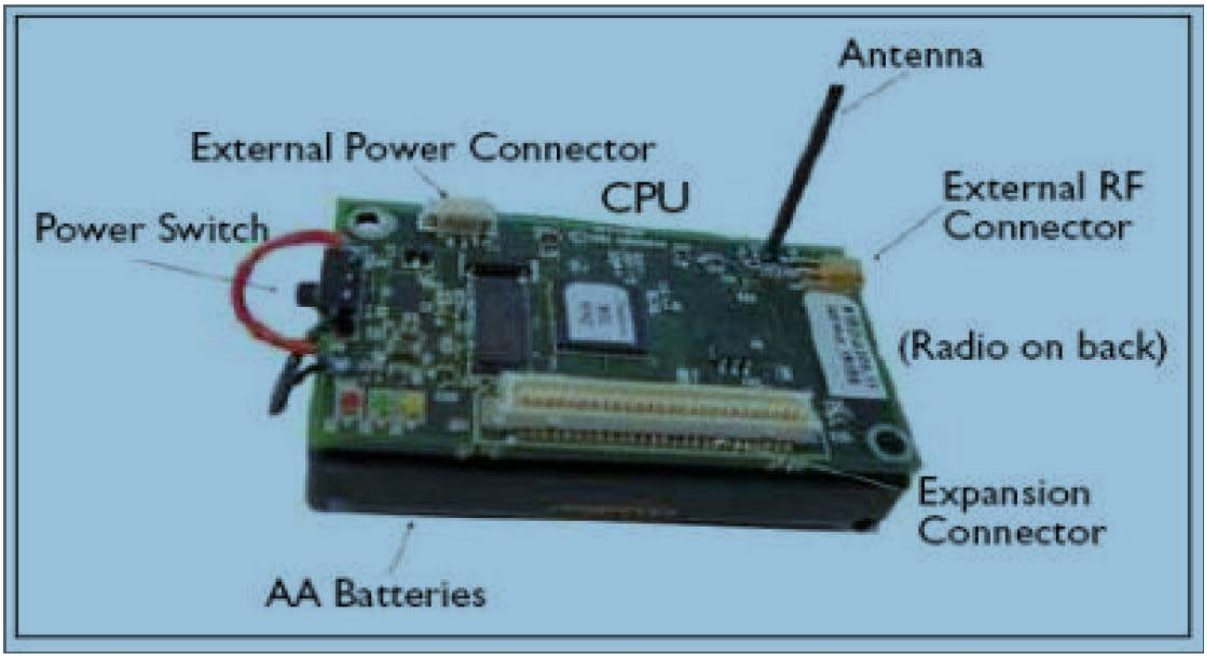

Figure 3: Hardware Components of Sensor [19]

These sensors inside a node are used to sense the environment and collect various data such as temperature, humidity, noise, pressure and wind movement. This node uses a TCP/IP protocol to communicate with other nodes and the base station.

2.1.3 NS 2.34

NS-2 is version 2 of an open-source, object-oriented, discrete event-driven network simulator [15]. It works by using various protocol implementations such as TCP, IP, UDP, FTP and telnet and assists network engineers to analyse traffic behaviour between nodes. This simulator takes oTCL scripts as input containing network configuration objects and scheduling information

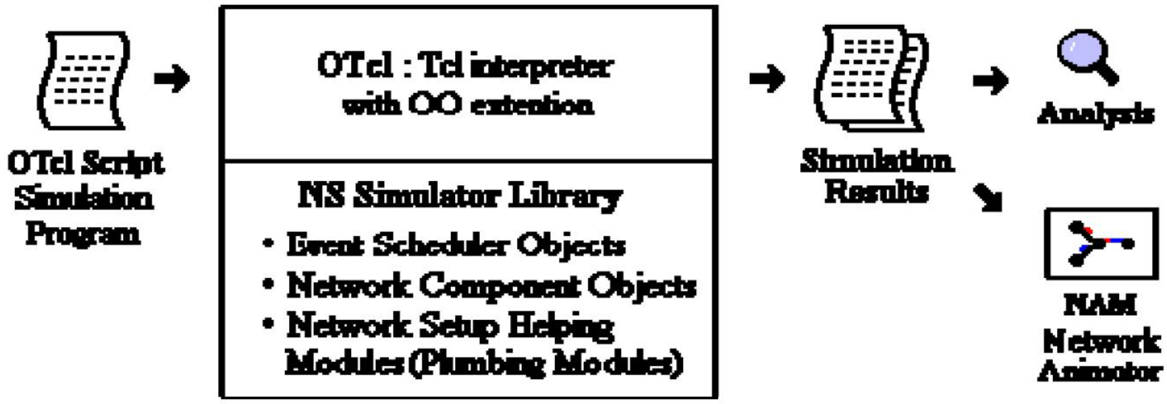

Figure 4: NS2 Architecture [8]

2.1.4 Various Protocols

Network protocol in DSN allows all the sensor nodes to collect information and send it to the base station. For all this work, the network protocol must be fault tolerant with minimum power consumption. So most of the Routing Protocols for DSN are divided into two categories as follow [24]:

- Flat Routing Protocols
- Hierarchical Routing Protocols

Flat routing protocols are ideal for small scale DSNs in which every node can directly communicate with the base station. Some of the flat routing protocols are Sensor Protocols for Information via Negotiation (SPIN), Directed Diffusion (DD), Flooding Algorithm and Sequential Assignment Routing (SAR)[24]. On the other hand, when the number of sensor nodes in a DSN increases, then this flat routing protocol creates a lot of overheads because each node is in direct connection with the base station. In this case, Hierarchical protocols come into action and groups the sensors into clusters with each cluster having one node cluster head. All the nodes in a cluster communicate with the cluster head node that further communicates with the base station. The cluster head is responsible for collecting all the data from the nodes in its cluster. So now the overhead decreases as the number of nodes communicating with the base station decreases. Some of the examples of hierarchical protocols are Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN), Geographic and Energy Aware Routing Protocol (GEAR) and Low Energy Adaptive Clustering Hierarchy (LEACH).

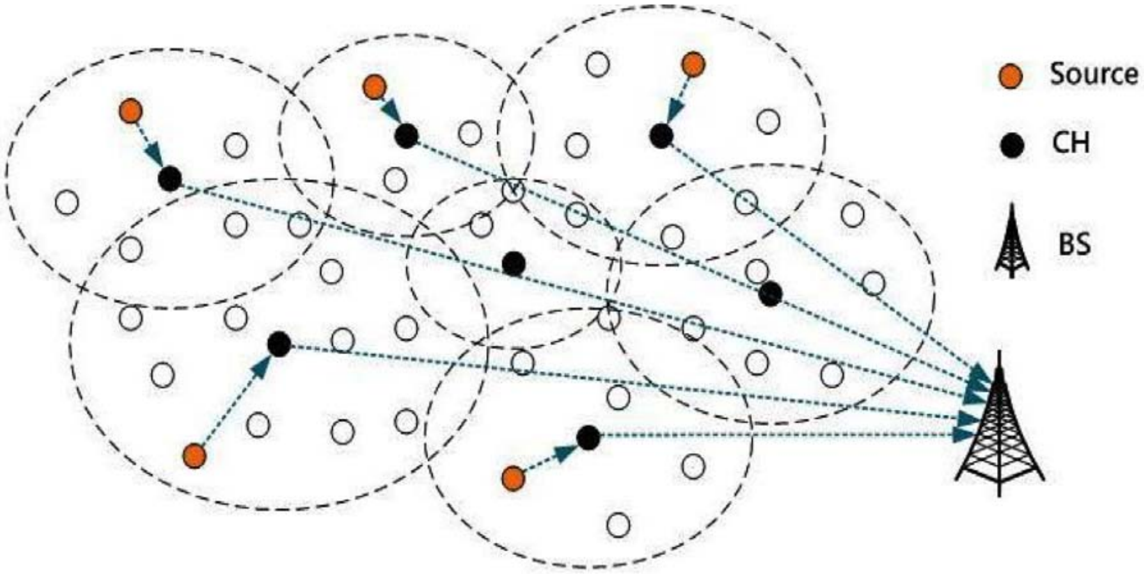

Figure 5: LEACH C Structure [26]

In this project we will be using various results of LEACH-C algorithm as we found this algorithm to be most efficient for experiment setup by analysing the results of experiments done by various researchers and students. LEACH-C is the centralised form of LEACH algorithm with self-organizing clustering function [10].

2.2 State-of-the-Art Technology

The information regarding existing/similar DSN products is given below:

1. Technology: Animals as Sensors [18]

   Class: Energy Conservation

   Design: To cover the larger area, animals are used as mobile sensors.

2. Technology: Geographic Adaptive Fidelity [25]

   Class: Environmental Data/Energy Conversation

Design: Identify the nodes which are relatively close and among them shut all the nodes except one node.

3. Technology: Geographic Information System (GIS) [14]

   Class: Environmental Data

   Design: using this technology to keep check on areas of more fire risks. Our Environmental Simulator is similar to this GIS.

4. Technology: Voltree [3]

   Class: Energy Conversation

   Design: Sensors get their power from Bio-energy

5. Technology: Waspmote [22]

   Class: Environmental Data

   Design: This is a sensor platform that can be used for variety of detections. This is a low power wireless network.

2.3 Literature Survey

During the entire phase of the project we read many articles, books and research papers. We learned a lot regarding DSN and its related issues and solutions. Some of the material provided a deep knowledge of DSN and are discussed below:

A Survey on clustering algorithms for wireless sensor networks by Abbasi and Younis [4]: This provides a good overview of fifteen clustering algorithms and protocols. It starts by describing objectives like fault tolerance, minimal cluster count, maximal network longevity, increased connectivity and reduced delay. After describing these factors, it then describes all the fifteen algorithms based on these factors. Algorithms are CLUBS, GS3, Linked Cluster Algorithm (LCA), Adaptive Clustering, Random Competition Based Clustering (RCC), Hierarchical Control Clustering, Energy Efficient Hierarchical Clustering (EEHC), Fast local Clustering Service (FLOC), Hybrid Energy-Efficient Distributed Clustering (HEED), Low Energy Adaptive Clustering Hierarchy (LEACH) and Distributed Weight-Based Energy-Efficient Distributed Clustering (DWEHC).

Chapter 7 of Ad Hoc Mobile Wireless Networks book gives details about the energy management in networks [21]. This chapter gives the complete description of power management in distributed sensors, its importance and various schemes that are required for power management in networks. These schemes include system power management schemes, battery management schemes and transmission power management schemes. This project is primarily concerned with transmission power management schemes.

A survey on Wireless Sensors Networks by Akyildizet. al.[1] contains a lot of basic information regarding wireless networks containing sensors. It gives all the information including their implementation and various issues related to these networks. It also explains various layers of the protocol that is used for communication between these sensors.

Beyond these reading materials, there were some articles which described MAC layer protocols such as S-MAC, T-MAC, U-MAC, DEE-MAC, Z-MAC, etc. [17]. Many research papers on power optimization of DSN are also read to have a complete overview of the factors effecting the power management in DSN.

Table 1: Summary of Articles

Article Name	**Type of Protocol/Algorithm**	**Name of protocol(s)**

MR-LEACH: Multi-hop Routing with LEACH[8]	Multi-hop and clustering-based communication protocol	MR-LEACH (Multi-hop Routing with Low Energy Adaptive Clustering Hierarchy)
Distributed Clustering in Ad-Hoc Sensor networks: A Hybrid Energy-Efficient Approach (HEED) [27]	Clustering Algorithm	HEED (Hybrid Energy-Efficient Approach)
A Low Energy Intelligent Clustering Protocol for Wireless Sensor Networks [16]	Clustering Algorithm	LEICP (Low Energy Intelligent Clustering Protocol)
Energy-Efficient Communication Protocol for Wireless Microsensor Networks [10]	Clustering-based communication protocol	LEACH (Low Energy Adaptive Clustering Hierarchy)
An Application Specific Protocol Architecture for Wireless Microsensor Networks [11]	Clustering-based communication protocol	LEACH-C (Low Energy Adaptive Clustering Hierarchy - centralized)
An Efficient TDMA Scheme with Dynamic Slot Assignment in Clustered Wireless Sensor Networks [9]	Timeslot allocation algorithm	(unnamed)
Power-Efficient Data Dissemination in Wireless Sensor Networks [6]	Event-based communication	TD-DES (Topology-Divided Dynamic Event Scheduling)

Dozer: Ultra-Low Power Data Gathering in Sensor Networks [5]	Communication protocol	Dozer
LEACH-HPR: An Energy Efficient Routing Algorithm for Heterogeneous WSN [12]	Cluster head election protocol	LEACH-HPR (low Energy Adaptive Clustering Hierarchy – Head Election)
An Energy Efficient Wireless Communication Mechanism for Sensor Node Cluster Heads [17]	Scheduling Algorithm	
An Energy-Efficient Clustering for Normal Distributed Sensor Networks [20]	Cluster Formation Algorithm	DAEEC (Density-Aware Energy Efficient Clustering)
Optimizing Energy-Efficient Reconstruction of Mobile Point Source in Sensor Networks [7]	Reconstruction algorithm	LPSS (Low power Self Scheduling Protocol)
An Energy-Aware Routing Protocol in Wireless Sensor Networks [13]	Routing Protocol	Energy Aware Routing Protocol (EAP)

Chapter 3: Project Plan and Schedule

All the details like lists of tasks, schedules and deliverables are presented in this chapter. This detailed plan also includes cost analysis which gives the estimation of benefits and cost of the project. In this chapter, details of budget and resources we have are also included.

3.1 Project Tasks and Schedule

3.1.1 Project Tasks

The table below shows the list of deliverables with start and end dates and the resources involved for that particular deliverable.

Table 2: List of deliverables

Task	**Start Date**	**End Date**	**Primary Resources**
Kick - off Meeting	2/5/2013	2/5/2013	All
Initial research	2/6/2013	2/10/2013	All
Project Abstract – Draft	2/11/2013	2/16/2013	All
Project Abstract – Approved	2/17/2013	2/20/2013	All
Project Abstract Presentation	2/22/2013	2/28/2013	All
Project Journal 1	3/1/2013	3/6/2013	All
Project Proposal – Draft	3/5/2013	3/7/2013	All
Introduction	3/8/2013	3/11/2013	All
Background and Related Work	3/12/2013	3/16/2013	All

Literature Survey	3/17/2013	3/26/2013	All
Project Description	3/23/2013	3/30/2013	All
Project Journal 2	3/31/2013	4/4/2013	All
Project Plan and Schedule	4/2/2013	4/9/2013	All
Project Proposal – Approved	4/9/2013	4/12/2013	All
Project Proposal Presentation	4/13/2013	4/18/2013	All
Project Report – Draft	4/17/2013	4/19/2013	All
Introduction	4/19/2013	4/21/2013	All
Background and Related Work	4/19/2013	4/21/2013	All
Project Plan and Schedule	4/19/2013	4/21/2013	All
Requirement and Analysis	4/20/2013	4/26/2013	All
Domain and Business	4/20/2013	4/22/2013	All
Customer-oriented	4/20/2013	4/22/2013	All
System Function	4/20/2013	4/22/2013	All
System Behaviour	4/22/2013	4/26/2013	All
Performance and Non-Functional	4/22/2013	4/26/2013	All
Context and Interface	4/22/2013	4/26/2013	All

Technology and Resource	4/22/2013	4/26/2013	All
System Design	4/25/2013	5/1/2013	All
System Architecture	4/25/2013	5/28/2013	All
Data and Database	4/25/2013	5/28/2013	All
Interface and Connectivity	4/25/2013	5/28/2013	All
User Interface	4/25/2013	5/28/2013	All
Component API and Logic	4/25/2013	5/28/2013	All
Design Problems, Solutions and Patterns	5/27/2013	5/1/2013	All
Used Tools and Standards	5/27/2013	5/1/2013	All
Testing and Experiment Plan	5/27/2013	5/1/2013	All
Project Journal 3	4/28/2013	5/2/2013	All
Project Report – Approved	5/2/2013	5/6/2013	All
Research and Experimentation	5/24/2013	7/4/2013	All
Implementation	7/3/2013	10/13/2013	All
Evidential Reasoning	10/2/2013	10/16/2013	All
Environmental Simulation	10/2/2013	10/16/2013	All
Integration	10/17/2013	10/19/2013	All

Testing	10/20/2013	12/1/2013	All
Create test cases	10/20/2013	12/1/2013	All
Functional Testing	10/20/2013	12/1/2013	All
Performance Testing	10/20/2013	12/1/2013	All
Final Project Report – Draft	12/1/2013	12/3/2013	All
Final Project Report – Approved	12/3/2013	12/6/2013	All
Final Project Demo	12/5/2013	12/9/2013	All

3.1.2 Project Schedule

Project schedule includes the detailed plan of the various stages of the project. This is shown by Gantt charts. The Gantt charts, one showing the work for first semester and other showing the work for whole year are given below:

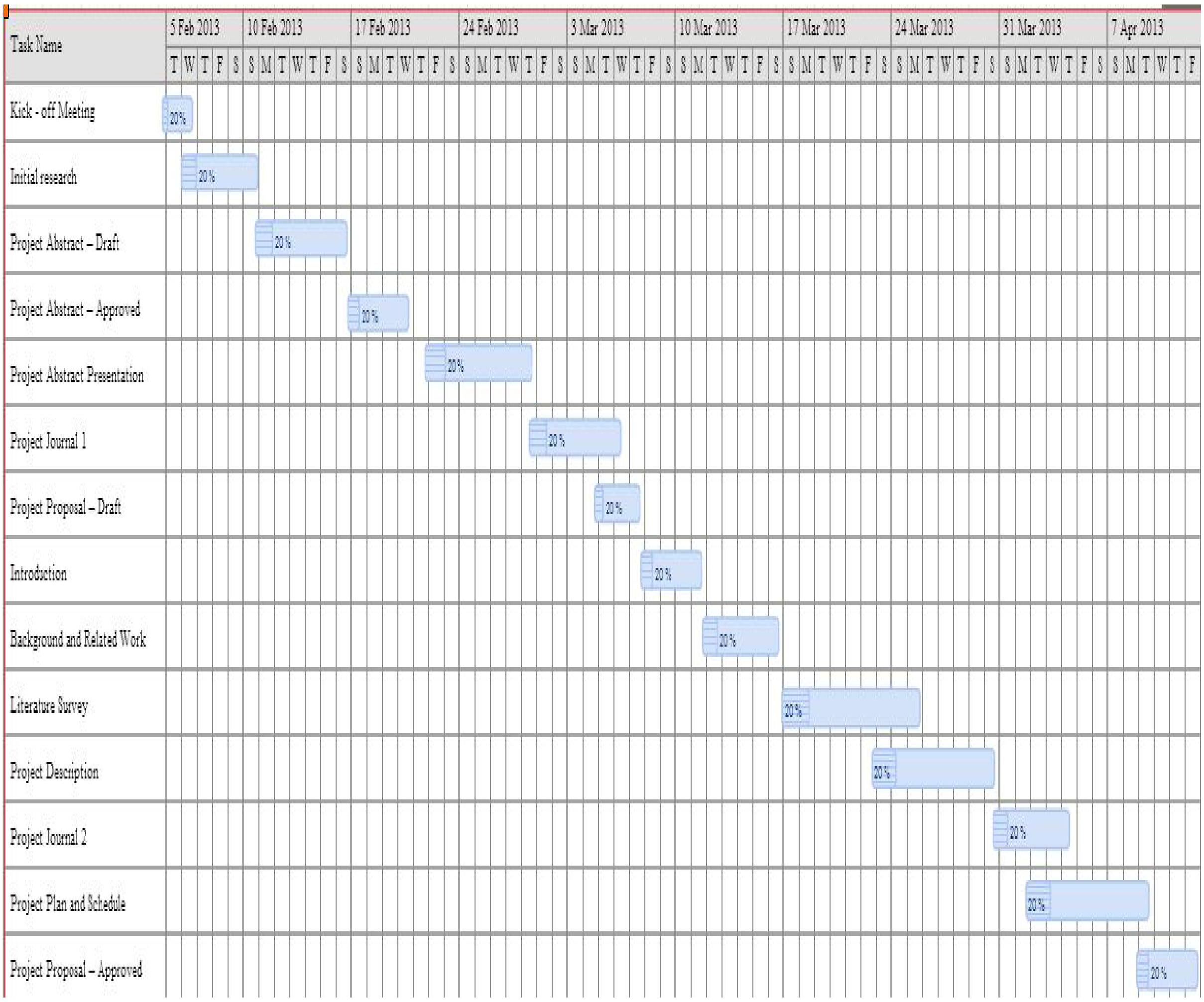

Figure 6 (a): Gantt chart 1

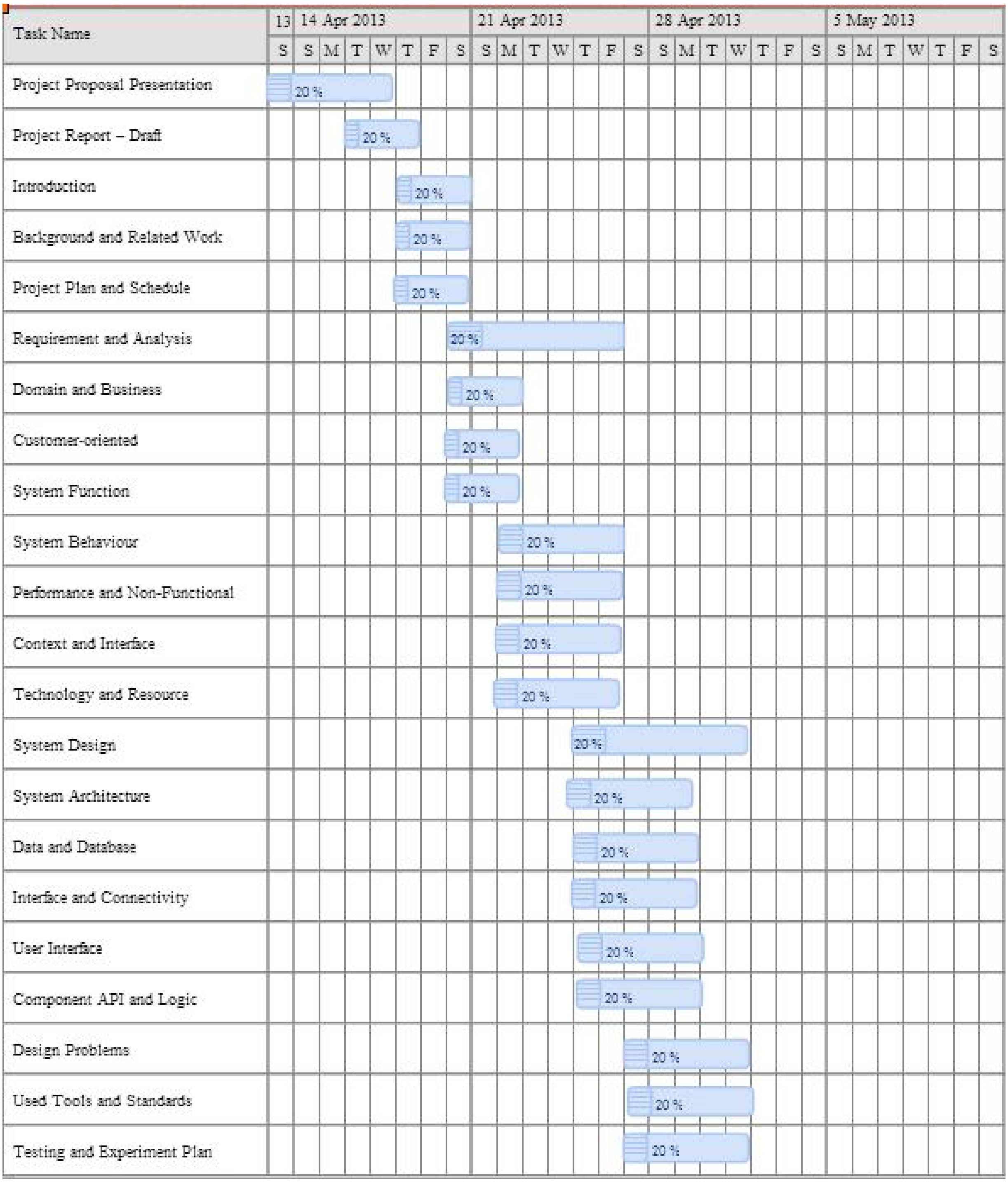

Figure 6(b): Gantt chart 2

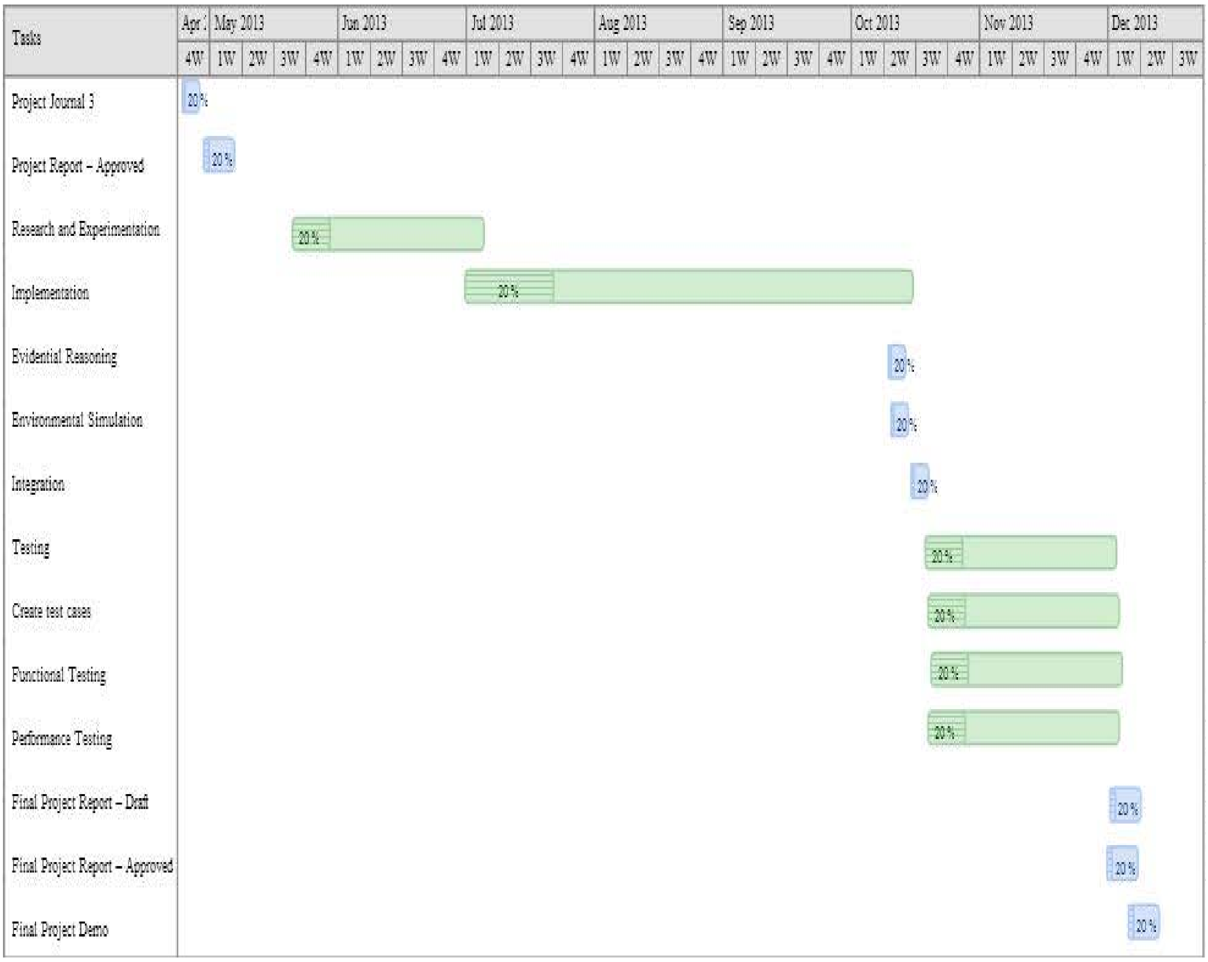

Figure 6 (c): Gantt chart 3

3.2 Project Budget, Resources and Cost Analysis

For any good project, there has to be a detailed schedule, budget and cost analysis and resources information that are available to us.

3.2.1 Project Budget

The table below shows the budget of the materials those are required during the development of the project.

Table 3: Budget for the material that is required

Category	**Item**	**Cost per unit**	**Quantity**	**Total Cost**
Hardware	Laptop	$1650.00	2	$3300.00
Software	Linux	$0.00	2	$0.00
	NS2	$0.00	2	$0.00
	oTCL	$0.00	2	$0.00
	C++	$0.00	2	$0.00
	IDE	$0.00	2	$0.00
	Office Software	$0.00	2	$0.00
	Google Docs	$0.00	2	$0.00
Other	Printouts	$0.10	1100	$110.00
	Reports	$5.50	10	$55.00
	Poster Board	$6.00	1	$6.00
	Internet Charges	$48	9	$432.00
Total Cost				$3903.00

3.2.2 Project Resources

Resources those are required for the project has been divided into two categories i.e. Human Resources and Material Resources. Both of these resources are described in the following sections.

3.2.2.1 Human Resources

The table below shows the available human resources that will be there for the development of the project

Table 4: Available Human resources

Project Topic	Optimizing Power Consumption of Distributed Sensor Networks
Project Advisor	Dr. Leonard Wesley
Cmpe Team	Nikhil Verma (Spring 2013 and Fall 2013)
	Vaibhav Bhatia (Spring 2013 and Fall 2013)
SE Team	Harvey Chan (Spring 2013)
	Tejaswini Karra (Spring 2013)
Project Period	2/5/2013 – 12/9/2013

3.2.2.2 Material Resources

This includes all the materials that are required to complete the project including laptops, software and other materials. We will be using our own laptops and open source software. Therefore not much cost is incurred for these resources. Other resources such as printouts, Poster Boards, Reports, etc. require some cost which is given in section 4.2.1.

3.2.3 Cost Analysis

For this master's project, we will not evaluate the cost paid to the engineers because we are implementing the project on our own. So we will be calculating the average number of hours required to finish each phase. This is just an estimation that we are doing. Actual hours may vary according to the amount of work done for each phase.

Table 5: Labour cost

Project Phase	**# of Resources**	**Duration (weeks)**	**Hours spent per week**	**Total Effort**
Planning	2	4	12	96
Design	2	5	13	130
Development	2	12	13	312
Testing	2	7	14	196
Validation	2	5	11	110
Total		33 weeks		844 hours

Chapter 4: Software System Requirements and Analysis

All the information regarding system requirements i.e. hardware and software including various components in the system that is required for this project are discussed in this chapter.

4.1 Domain and Business Requirements

The diagram below is the activity diagram showing the work flow for this project

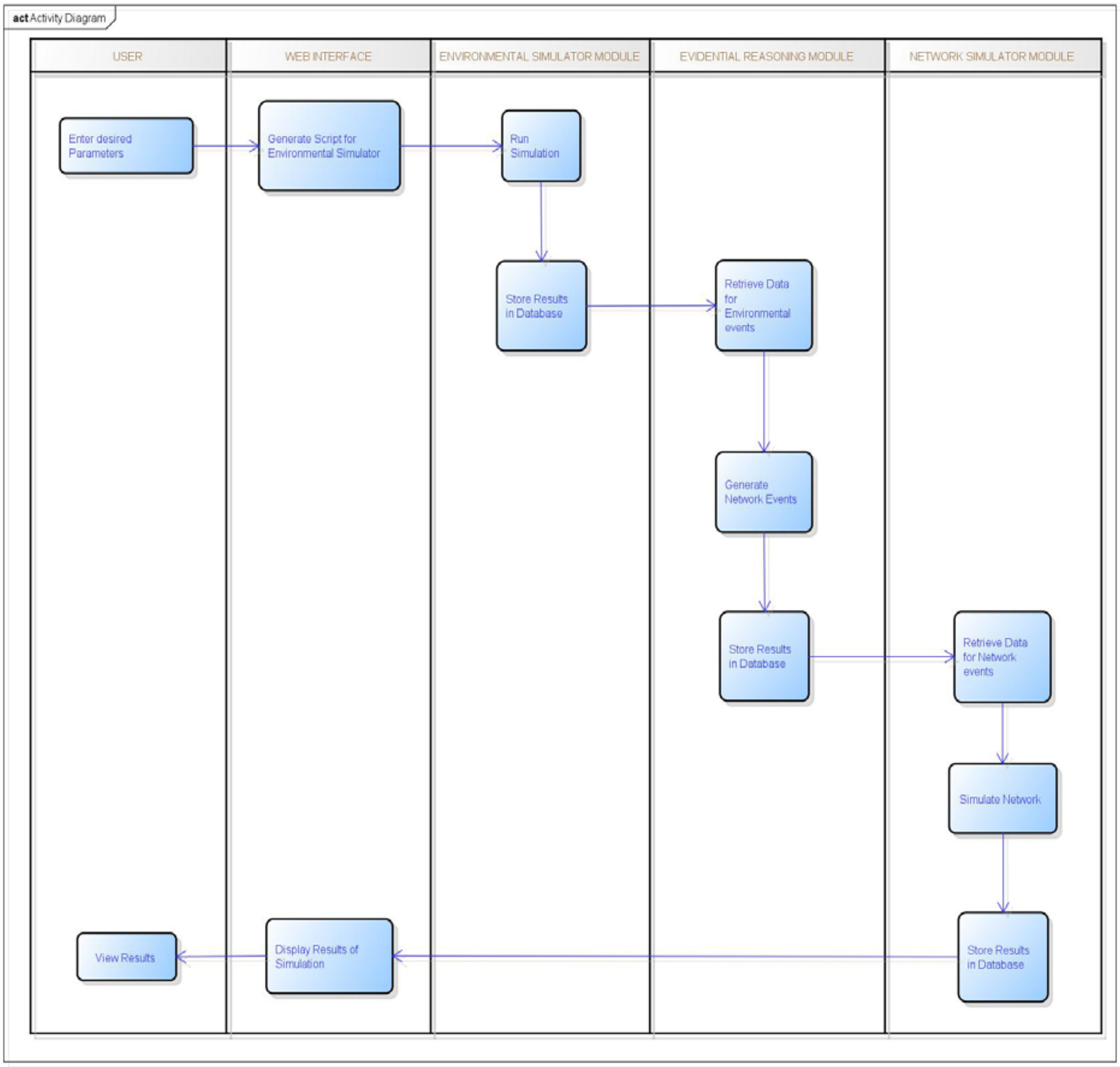

Figure 7: Activity Diagram

The diagram given below is the Data Flow diagram illustrating the flow of data between various entities.

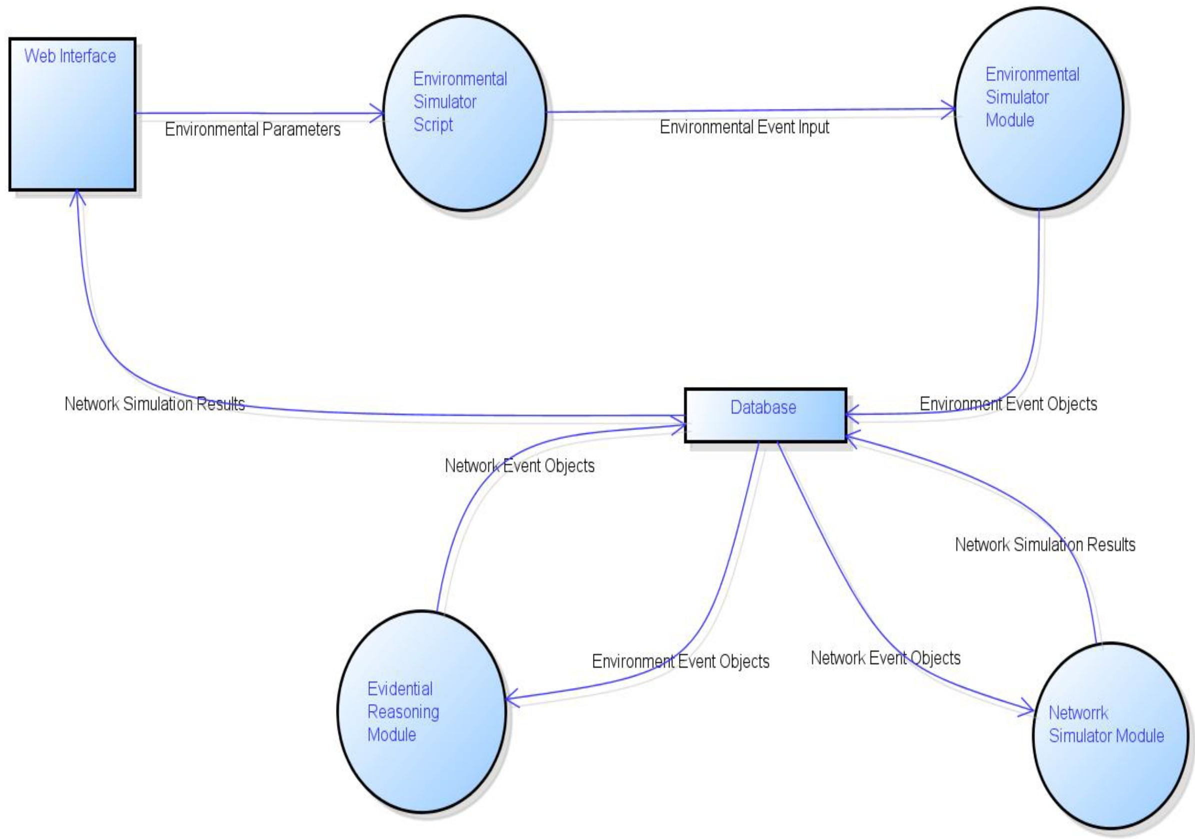

Figure 8: Data Flow Diagram

4.2 Customer-Oriented Requirements

The user of simulation software must be able to use the software easily. An Interface must be there for the user to modify the simulation parameters according to their requirement. Some of the parameters that we will keep for users to change anytime they wish to are:

- Number of nodes in DSN
- Simulation Time
- Environmental Conditions for Experiment

For this purpose, a proper GUI must be there for the user allowing him to make changes in an easy way through drop down menus, text boxes, etc. This will reduce the set-up time which was more if the users were forced to make changes in the scripts which are produced to run the simulation. The results should also be shown by proper graphs by comparing the experiment results with the benchmark.

4.3 Software System Functional Requirements

The current simulation frame determines the environmental state of simulation, specified by the configuration file generated by user using GUI at the start of simulation.

The simulation software constantly updates the state of environment for the current simulation frame, throughout the simulation process and stores the state in the database. Future analysis can be done on the simulation frames stored in the database. The visual representation of the environment can be represented easily at any simulation point using the simulation frames, also the environment states of the current simulation are updated in all the data structures and the databases for processing the next simulation frame.

The Evidential Reasoning(ER) module inputs the frame from the result of the environmental simulator processing module through message passing framework. The next output of ER module is produced using the current network state and environment simulator output. The changes in the environment state output and the network state from network simulator determines whether a certain node is in on or off state.

Network simulator receives the input from the ER module, containing the instruction for the activation and the deactivation of the nodes. The data received by the network simulator is used intelligently to power on or off nodes according to the algorithm used in ER module.

The energy data is tabulated after processing the frames, based on user defined setting for granularity and length of simulation and stored in the text file. This file contains the amount of energy in each node at each frame. The graph and visual representation of the information stored in the file is created for the better understanding and analyzing the throughput of the sensor network.

ER. LEACH-C are the efficient routing protocols for communication in DNS, the future feature being consideration is the integration LEACH –C with this protocols in the simulation process. Integration of the LEACH-C in the ER algorithm decreases the power consumption.

The level of integration with LEACH-C and ER can be modified by the user, between the different mode i.e. fully integrated, semi integrated or no integration mode. Automated simulation scripts can be generated by the user through web interface for network simulator.

4.4 Software System Behaviour Requirements

The diagram given below is The State Diagram for Environmental Simulator module.

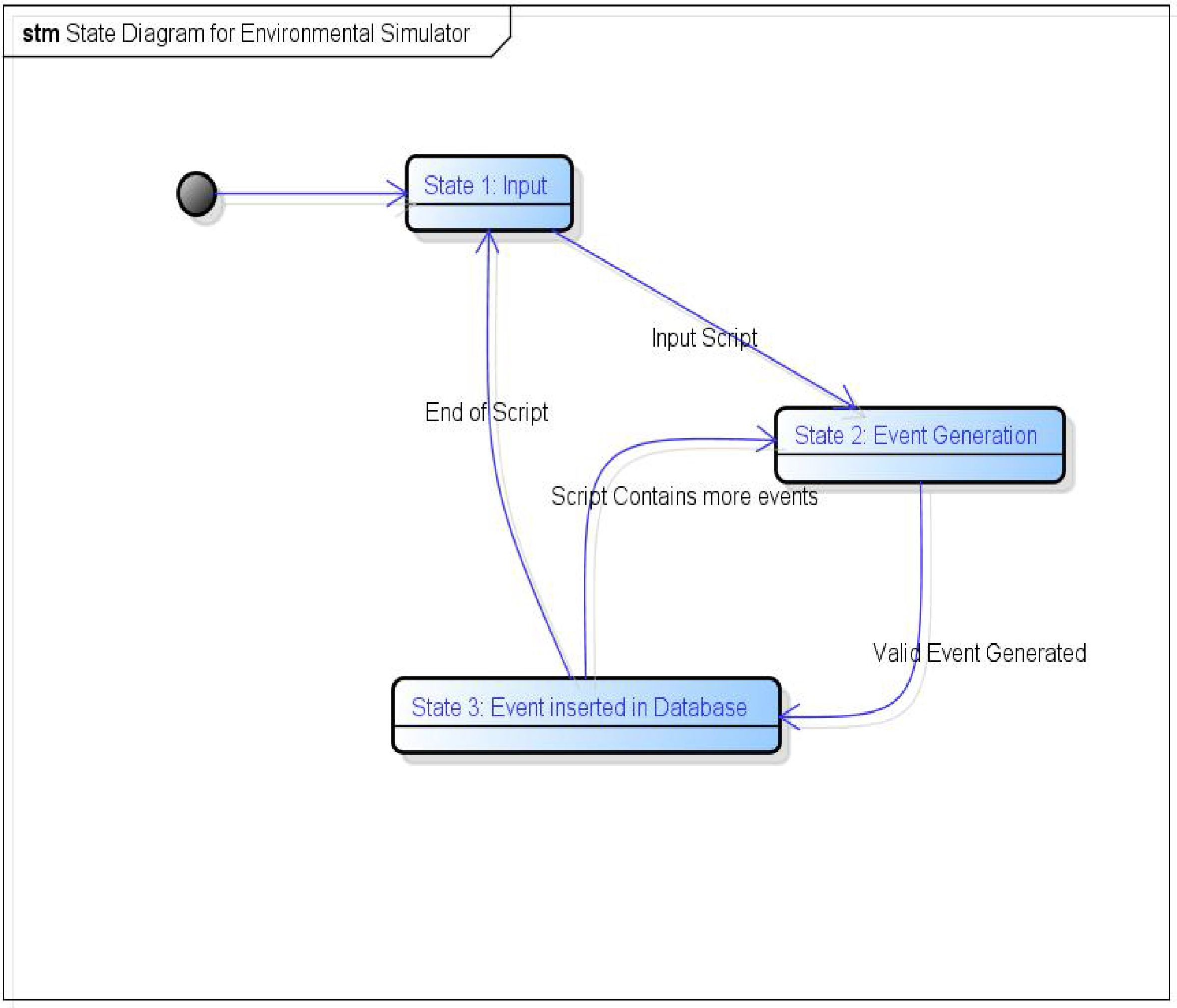

Figure 9: State Diagram for Environmental Simulator module

The environmental simulator is the first module in the simulation process. It gets its input from web interface in the form of script containing the environmental conditions and generates the environmental events according to the input script. These events are then stored in the database which are later used by Evidential Reasoning module.

The second diagram given below is the State Diagram for Evidential Reasoning module.

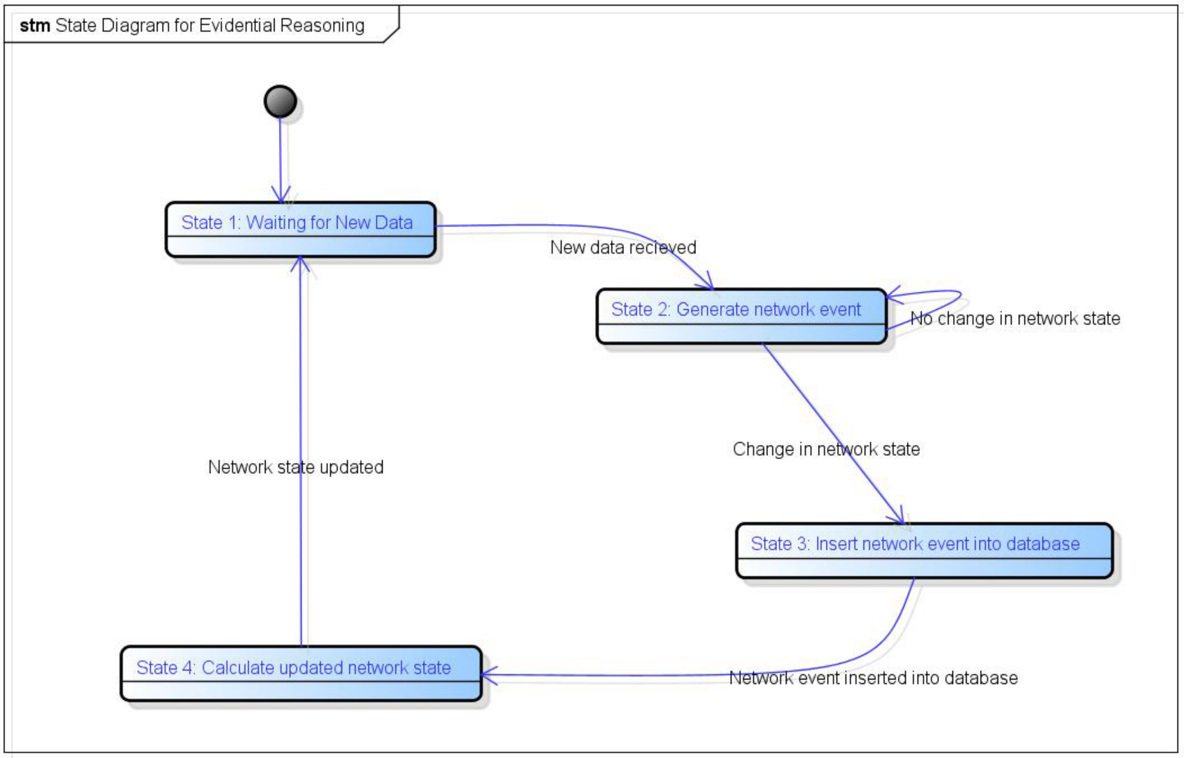

Figure 10: State Diagram for Evidential Reasoning module

The evidential reasoning is the second module in the simulation process after the environmental simulator. This module gets its input from the database in the form of environment event objects. Based on these environment events such as fire, rain, windy, etc. various network events are created which decide the operation of nodes. These network events are stored in the database which is further used by network simulator.

The diagram below is the state diagram for the third module i.e. Network Simulator module.

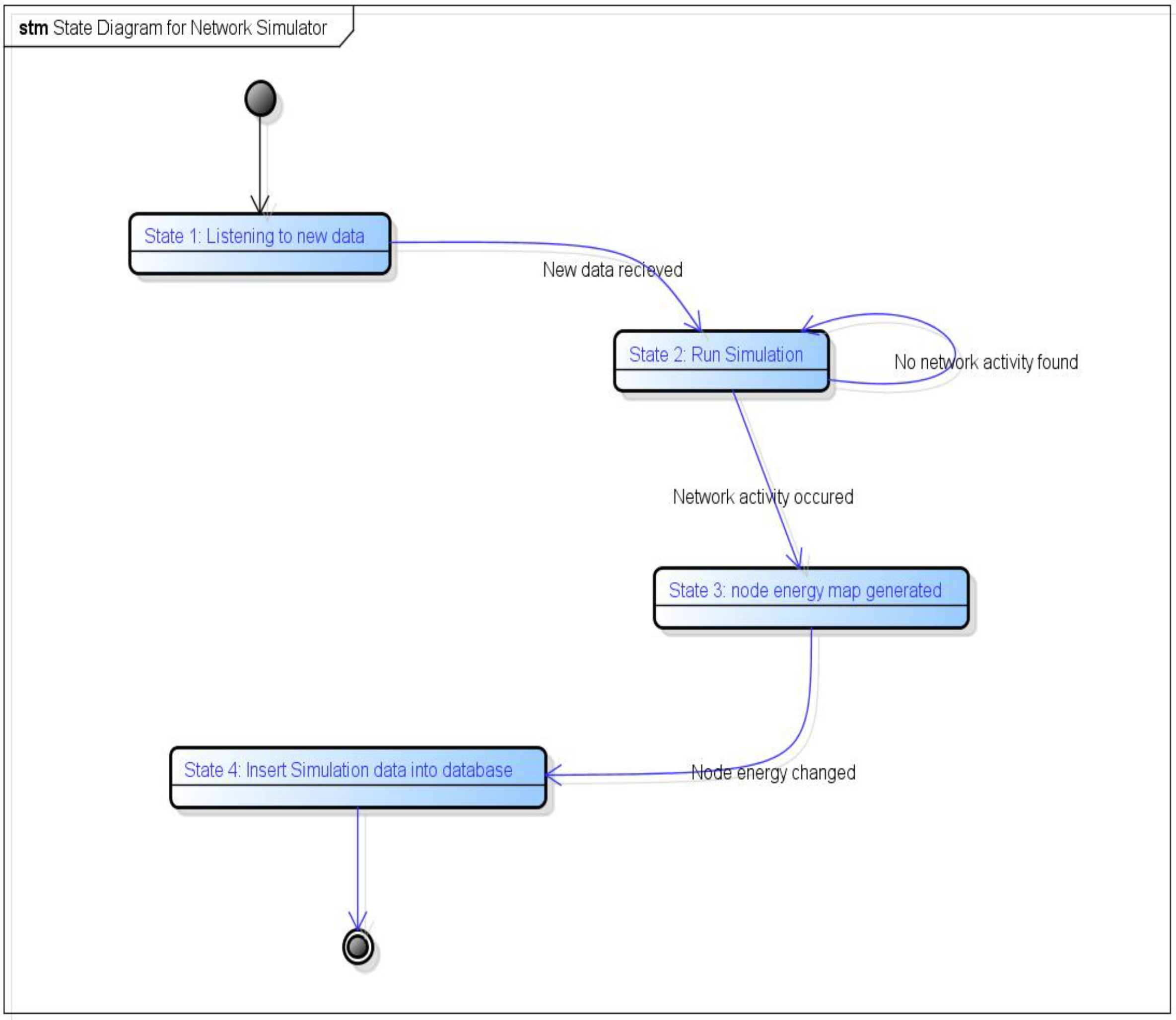

Figure 11: State diagram for network simulator module

The network simulator module is the last module in the simulation process. This module gets its input from the database in the form of network event objects and runs a simulation. After simulation, the results are stored in the database and a new energy map is displayed to the user.

4.5 Performance and Non-Functional Requirements

- Data about battery power consumption of various nodes and their runtime will be provided by simulation.

- No need to run the previous simulations again and again as simulation will have the ability to compare different simulations with runtime.
- Comparison of LEACH-C and our ER approach will also be done by simulation by comparing their performance.
- During simulation, any error that will be detected will be shown to the user in some meaningful terms.
- The software for simulation will be easily accessible online.

4.6 Context and Interface Requirements

The diagram below is the Use Case diagram for the system.

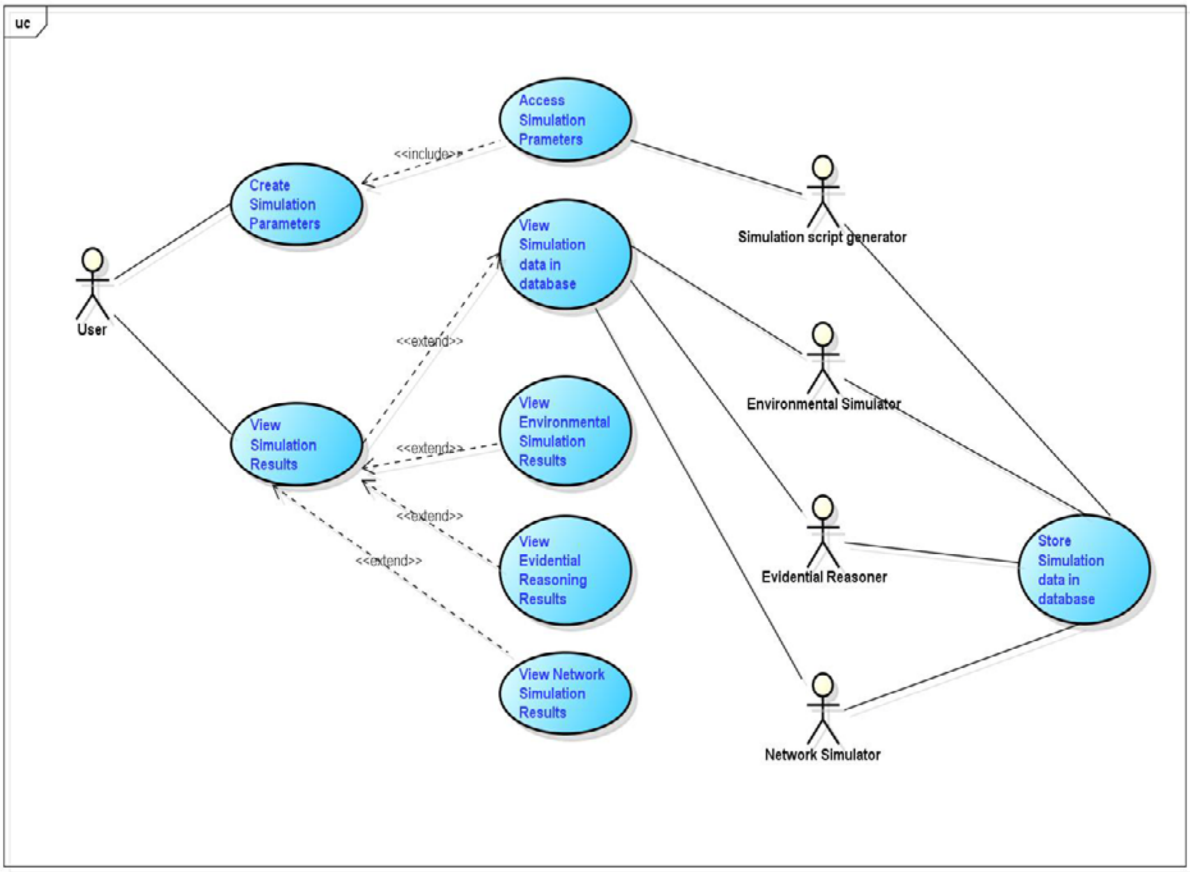

Figure 12: Use Case Diagram

A web interface will be provided to the user to run the simulation. A user will be asked for various parameters which will be entered through web interface. A user will also have an option to save the results in the database. This web interface hides the actual simulation from the user by providing a GUI which shows the final results.

4.7 Technology and Resource Requirements

4.7.1 Hardware

The hardware requirements for the simulation suite are given below:

Processor	Intel Dual Core 2.6 GHz
Memory	4096 MB SDRAM
Storage	210 GB Hard Disk
Network Connection	110 Mbit NIC
OS	Red Hat Enterprise Linux 3ES
Kernel	2.4.21-32.0.1.EI.smp

Table 6: Hardware Requirements

4.7.2 Simulation Software

The software which we are using is NS2.34 and MIT LEACH-C simulation modules for network simulator. We will be developing Environmental Simulator and Evidential Reasoning modules which will be integrated to form the final system.

4.7.3 Programming Languages Used

The programming languages which we will be using are given below:

Language	**Purpose**
PHP/HTML 5/CSS	Frontend (Website)
JavaScript	To show the results
Tcl	Scripts for NS
C++	NS modules

Table 7: Programming Languages

4.7.4 Server Software Requirements

The Server requirements are given below:

Server	**Purpose**
MySQL	Storing various intermediate and final results
Apache Server	To run the web interface
Linux	To run the NS package

Table 8: Server Requirements

Chapter 5: Preliminary Software System Design

This chapter gives the software system architecture, software database and data design, system interface and connectivity design, system user interface design, system component API and logic design, and Software design problems, solutions and Patterns. It contains numerous diagrams to help understand the basic architecture of our software system design.

5.1. Software System Architecture Design

This chapter provides the architecture design, which depicts the components, their relation and connectivity. Below, the figure shows high-level system architecture. The user interacts with the web interface and gives the desired environmental parameters. The web interface then generates the ES script and passes on to the environmental simulator. Environmental simulator then creates objects for stores in ES event data structure database. The outputs are then passed to evidential reasoning module. Evidential reasoning module uses ES event Data interpreter to interface ES data structure with its program. The result is generation of NS2 events. These events trigger the NS2 simulator, which then generates simulations to be stored in database. Also, the users view these simulations on the web interface.

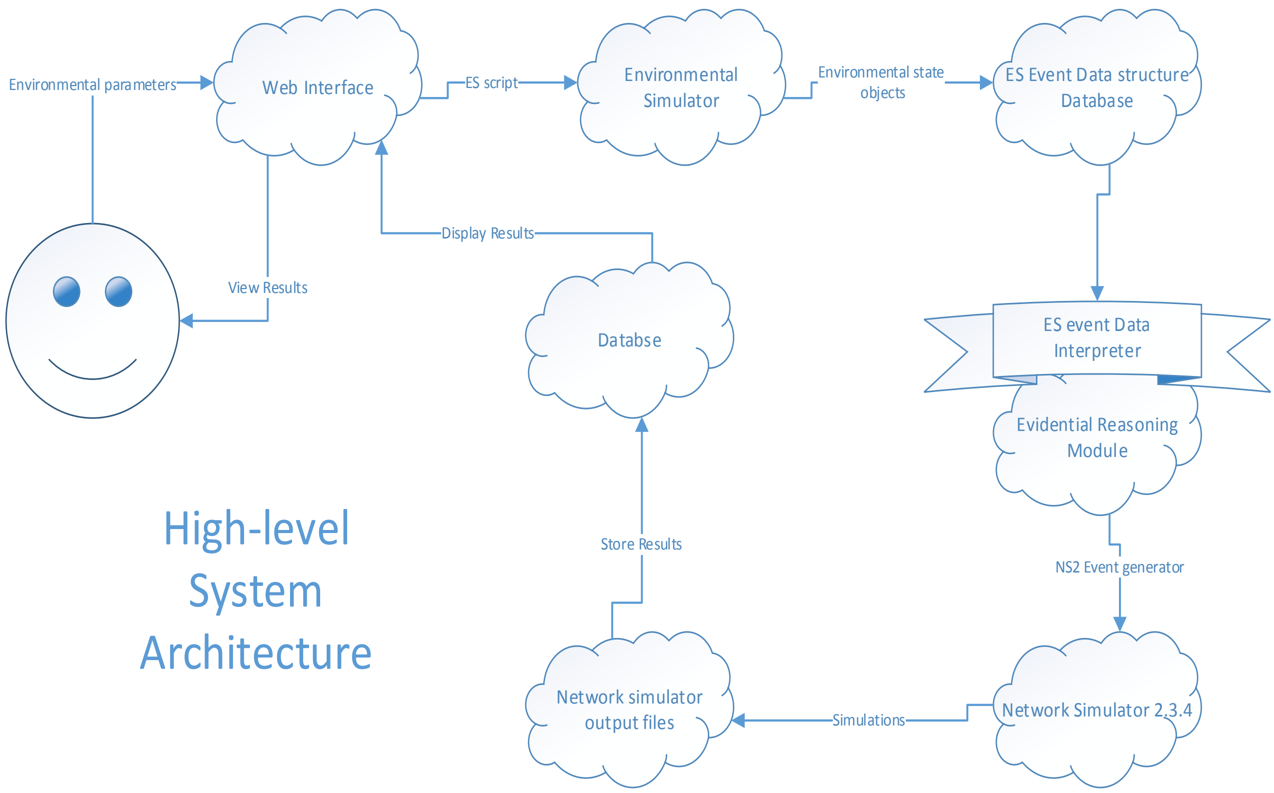

Figure 13: High-level system architecture

5.2. System Data and Database Design

We gather data from our four main components; web interface, environment simulator, Evidential reasoning, and Network simulator. For each of these components, tables were created. Below figure shows an entity relationship diagram for data and database of our system.

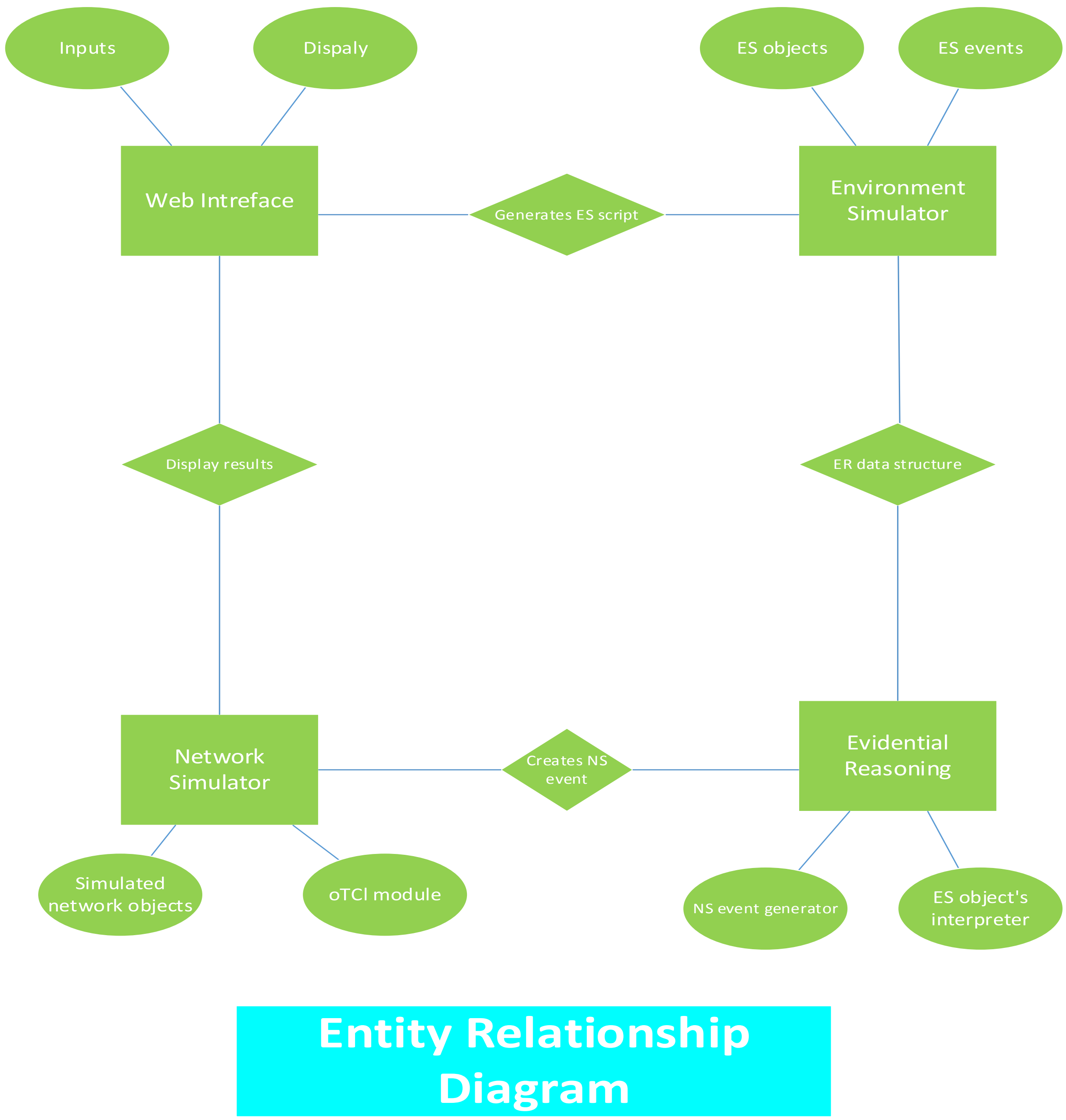

Figure 14: Entity Relationship Diagram

5.3. System Interface and Connectivity Design

This section describes how a user communicates with the system through web interface. The system web interface home page includes project description and authentication. After authentication the user is guided to either creating a new event or viewing old results. The Create new event page, accepts the attribute shown in the next figure. As the project is specific to scenario of fire-fighting in a forest location; system accepts parameters accordingly.

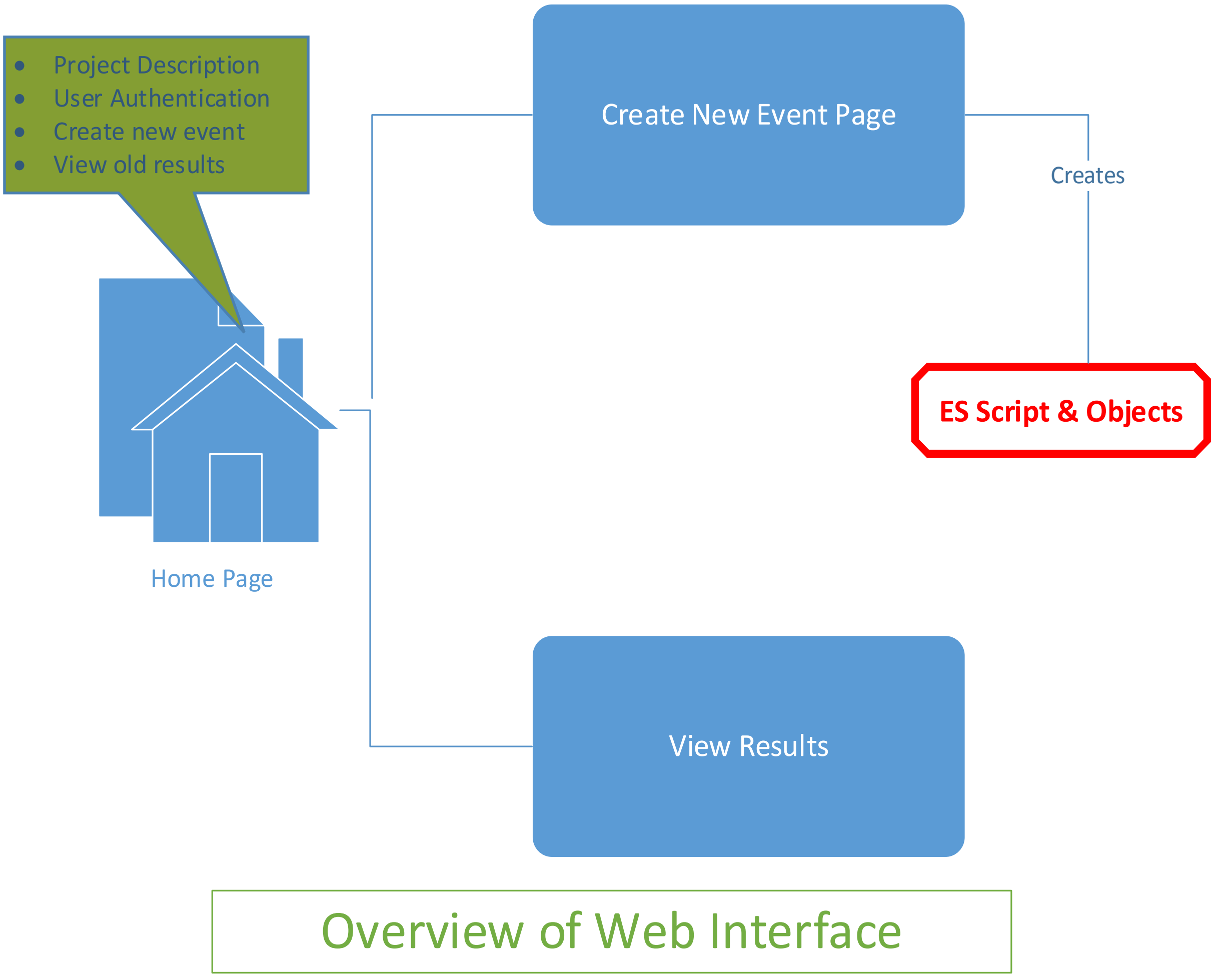

Figure 15: Overview of Web Interface

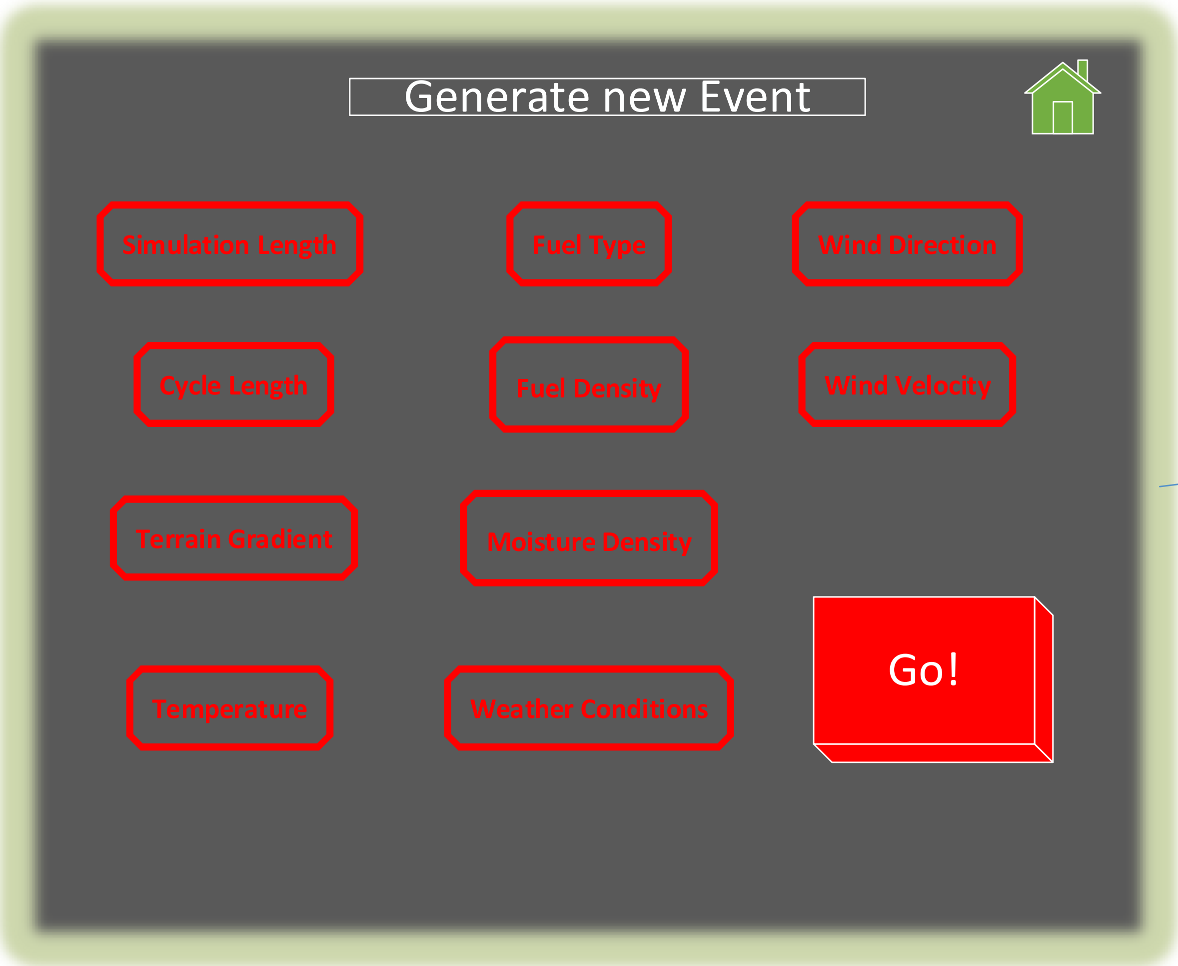

Figure 16: Overview of create new event page

Next we have the system interface of all the major components of our system. Users interact with the web interface. Environmental simulator consists of an interpreter and state object factory. The objects from ES is send to Evidential Reasoning module. ER module interprets these object and performs calculus on the given objects. It then generates the event for NS2 module. Network simulator gets the TCL script and data from Leach-C module. It then generates the resulting simulations.

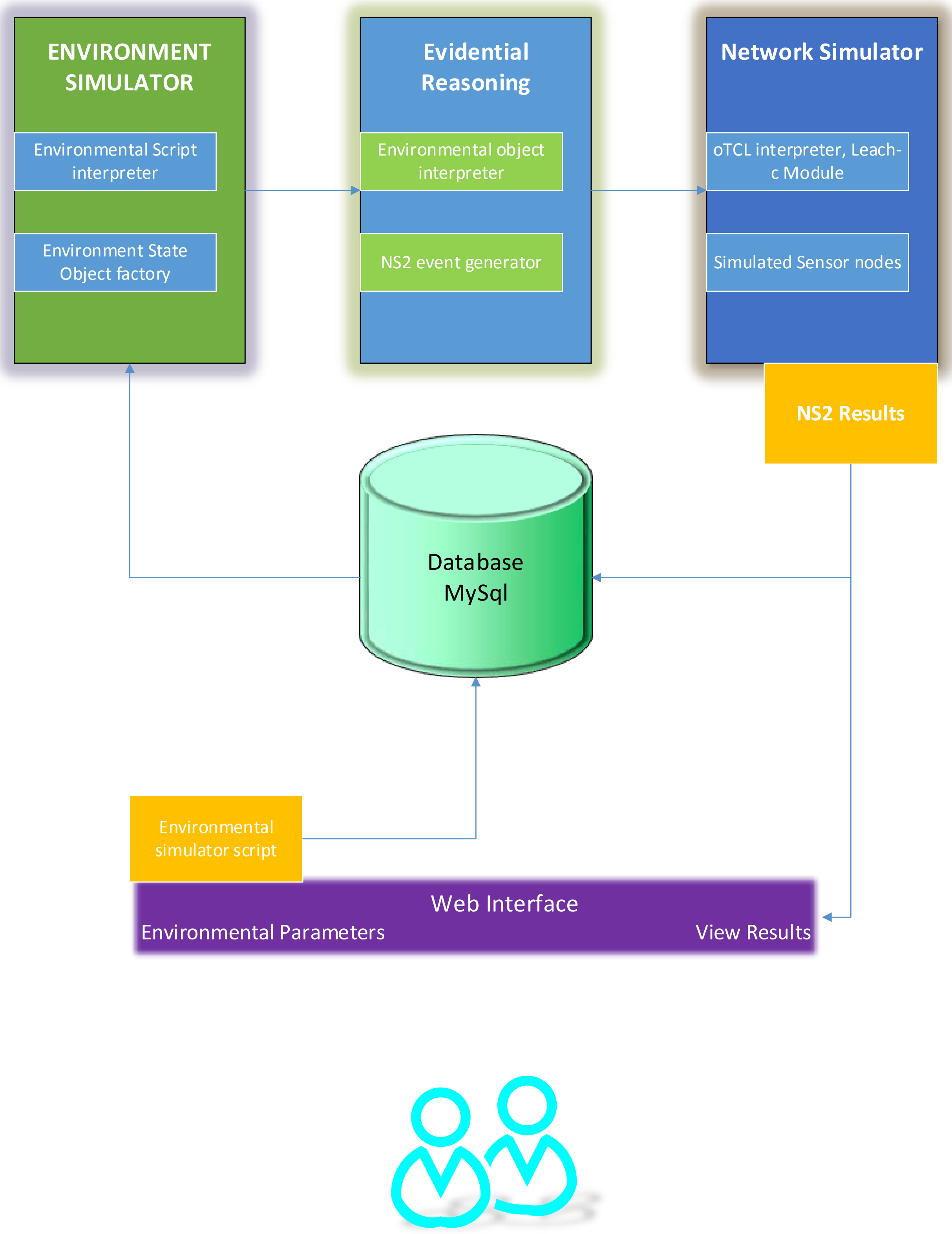

Figure 17: Components connectivity

5.4. System User Interface Design

This section gives an overview of system-and-user interface design, along with operation flow.

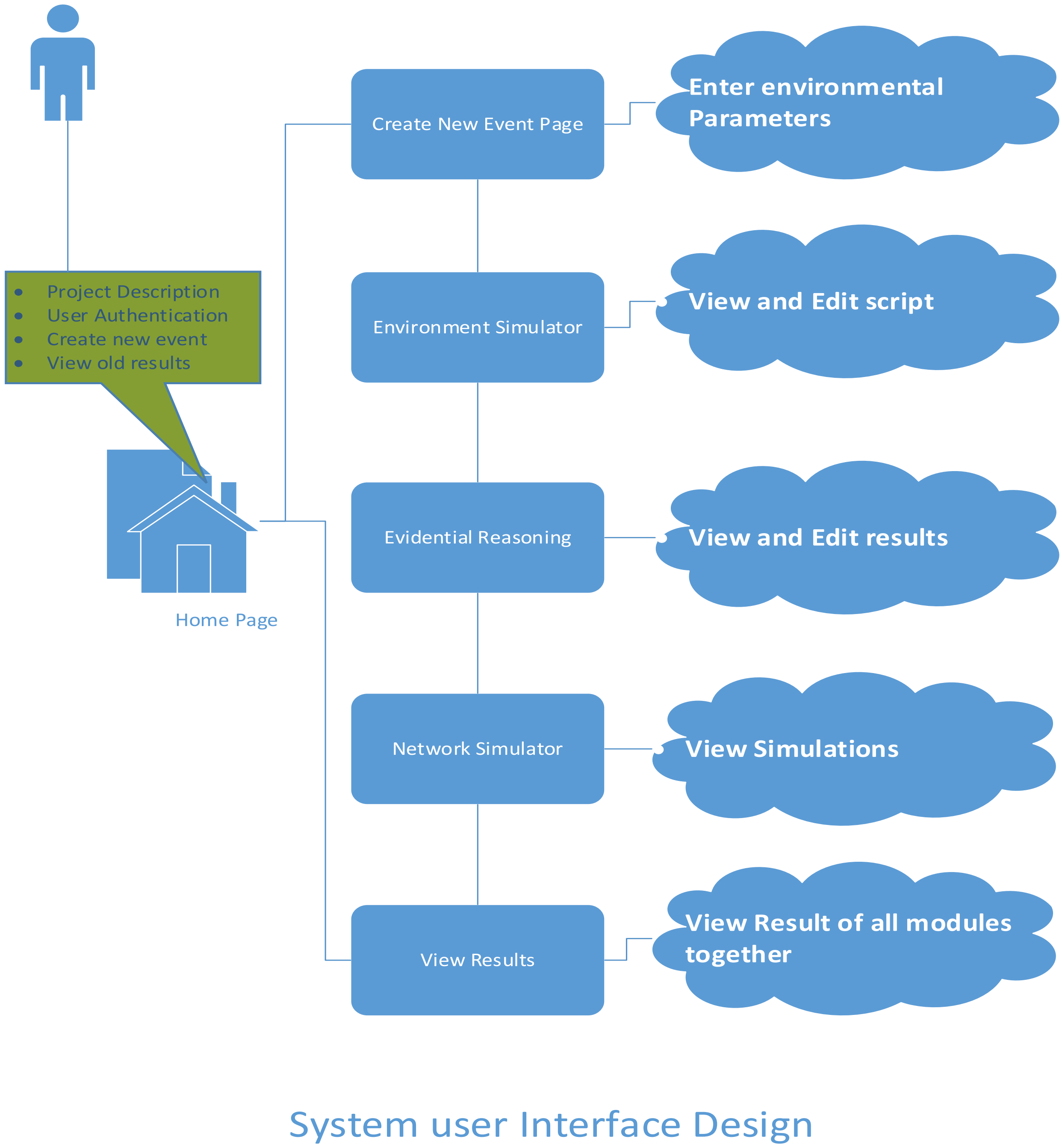

Figure 18: System-user Interface Design

The above figure shows the system-user interface design. The user interacts with the system using web interface. After user authenticates, he is directed to create new event as explained in chapter 5.2. In creating new event, the user inputs environmental parameters. Then the process goes to environment simulator's page. Here the user can see script being processed and the working of environment simulator. After its processing, the user is navigated to evidential reasoning module. Here the user sees the working of evidential reasoning module. After its processing, the user is navigated to network simulator. Here the user sees the result obtained from simulations. Finally, results from all the modules are shown on final page. This helps user understand the entire process.

5.5. System Component API and Logic design

This section discusses the components API and logic design through a UML sequence diagram. The diagram shows all major components and how system serves the user.

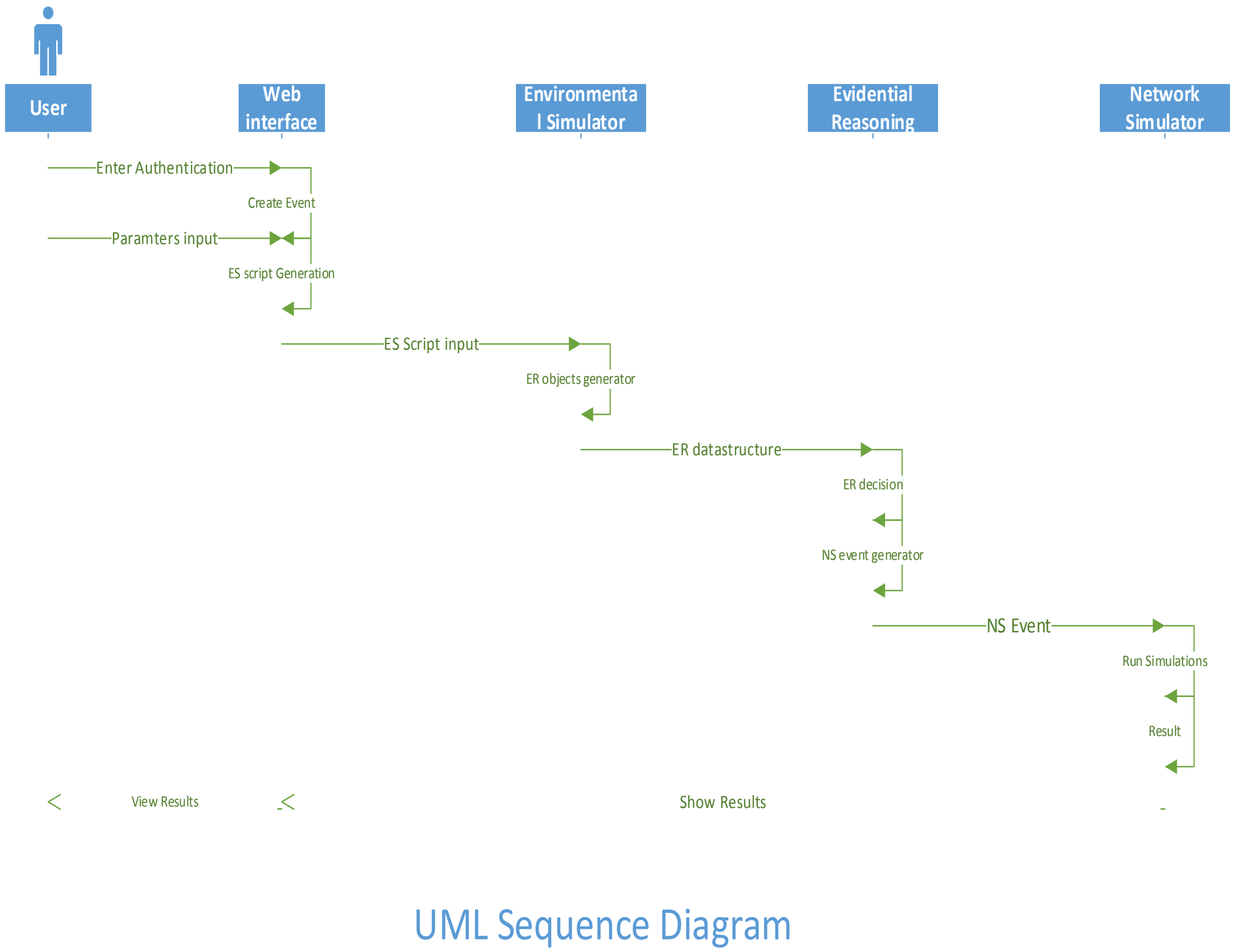

Figure 19: UML sequence Diagram for logic design

The above diagram shows that a user uses the system through a web interface. User enters its authentication. Web interface goes to create even page, then user enters parameters. The ES script is generated and passed to ES module. The ES module creates object and sends to ER module. ER module processes the inputs and makes decision. The network event is then sent to Network simulator. The simulator then generates simulations which are viewed by the user.

5.6. Software Design Problems, Solutions, and Patterns

The main problem as seen from developer point of view is how to interface the components. The four components are entirely different from each other. For them to communicate with each other a set of protocols need to be defined. Firstly, the web interface takes inputs from a user and the script for Environment simulator needs to be generated from this input. The solution to this problem is that the programming for ES script should be done in the web interface itself. The script then sent to ES needs to be interpreted.

Next problem is the creation of objects and data structure to be used by evidential reasoning module. Evidential reasoning module is coded in C++; hence, the objects will be created by ES and invoked by ER. Next, the ER creates network event for NS2 module. This needs to be in sync with Leach-C algorithm, as the data from ER and Leach-C both need to be run by NS2.

Another major problem is the time that NS2 module will take to run simulations. As the number of nodes increase, the NS2 module can take forever to produce results. The solution to this problem is use better system configuration which will be discussed in the next section.

Chapter 6: Used Tools and Standards

6.1 Used Tools

The optimization of the team’s time for any project greatly depends on the tools used. For this project successful completion we have used numerous tools and will continue to do so.

Google Docs, WordPress, Dropbox and GitHub are some of the applications which we have used for collaboration of the project documents, design and communication for the project deliverables. Google Docs helped in creation of the documentation and collaborating at the same time. Schedule creation, Gantt charts and diagram can also be made in Google docs and collaborated. The sharing of the documents amongst us is done using the Dropbox, while the code is shared through the GitHub. The project information organization communication with advisor and the team mates is done through WordPress. StarUML and Microsoft office helped in finishing the design and documents.

Research was another aspect of project where tools played a major role. In deciding the approach for the execution of our project we gathered information from article databases and the websites. Google Scholar, ACM and IEEE are some the sources whose access was found from the SJSU’s library.

The simulator and the code for the project are executed using RedHat Linux server. The network simulation according to our requirement is done using NS version 2.34. The start environment is controlled using NS, it includes node count, the node location and the energy required at the time of execution. oTCL scripts and C++ modules languages are used to simulate the communication with each other. The dynamic aspect of the simulation is written in C++ and the static code is written using oTCL. Network Animator, NAM is fed the result of the NS2 simulation using NS2. NS2 helps un visualizing the simulation, by capturing the simulation as video with the ability to review the simulation process. Eclipse and textpad are the text editors and IDE for coding and MySQL and PS/SQL are used ad relational databases.

6.2 Adopted Standards

The uniformity in everyone’s work and creation of recognizable and the understandable document and the result by the individual is achieved through the Standards. They provide the guidelines for plan, management, designing and the documents format as well as the data. The standards are integral part for high quality results and they are required to be followed in the project.

Waterfall software development lifecycle model is followed for the project development, as the set of standard described are comprehensive and can be followed for SDLC. The requirement and design specification documents are documented using IEEE Requirement Description and IEEE design specification document. Testing of the project is done on IEEE standards described in the IEEE standard 820. Class websites defines the class standards and our project documentation follows them.

Chapter 7: Testing Plan and Experiment

This chapter includes all the testing plans and experiment that we have planned to perform to complete this project.

7.1 Testing Plan and Experiment Scope

Testing is the most important phase in our project. This is because of the fact that we have to check for performance gain that we will get by using evidential reasoning. Testing will be done at each and every phase of the simulation and data will be checked at every entry to the database. The major component where the testing is most critical is network simulator module because this is the module in which all the energy calculations will take place and fortunately we have the results from the researchers working on LEACH-C approach so we can keep these results as a baseline to compare our results and check that our network simulator is calculating the correct results.

7.2 Testing Plan and Experiment Approaches

7.2.1 Web Interface

This web interface is the platform from where the user can interact with the system. This will be developed using HTML 5/PHP and JavaScript. For testing of this phase we will be using unit codes which are written using jasmine testing framework. As this phase is not that complicated and is the smallest part of the project so it does not require any rigorous testing.

7.2.2 Environmental Simulator Script Generator

This module generates the scripts based on the values given by user through web interface. These scripts act a input to environmental simulator which uses these scripts to generate environmental events. These scripts are based on simple if-else logic which is tested by environmental simulator itself by throwing an exception if some error is detected during the simulation face.

7.2.3 Environmental Simulator

In this module, environmental events are generated for the evidential reasoning module. We will be using unit testing to perform testing in this phase. Framework like CppTest or the built in test suite of Visual Studio will be used to perform the testing.

7.2.4 Evidential Reasoning Module

This module contains the most complicated code in our project as this module controls the power state of the sensors. We will be performing rigorous testing on this phase very carefully. This testing will be done based on the hand-calculated solutions which have been provided to us by Dr. Leonard Wesley. These solutions will be used to test each and every portion of the module.

7.2.5 Network Simulator

This module is the main component of our project as all the calculations related to power consumptions will be done by this module. As we mentioned earlier in this chapter, we will be using the data available by other researchers and will keep that data as a baseline to do the comparison with our results. This seems to be the safest method of testing as there is a huge amount of data which is available to us by other researchers.

Chapter 8: Project Summary

This concludes the first part of our project i.e. CMPE 295 A. In this we have completed our project design and planned the implementation and testing which will be done in the next part. The document gives a complete description of the project with all the requirements and budget that is required for the successful implementation of the project.

This project is all related to the power consumption of sensors in DSNs which becomes the major factor in rural area where there is no fixed source of power. So for this we will be using Evidential Reasoning to control the functioning of the sensors. The project is a good academic project as we will learn a lot of programming and networking concepts.

The main deliverable for this project will be a good user interface and the visualizations for the simulation results showing the comparison of our approach with LEACH-C. the user interface will be designed in such a manner that anyone can input their conditions without knowing any programming language or technical stuff and can see the results and understand them.

References

[1] Akyildiz, I.F.; Weilian Su; Sankarasubramaniam, Y.; Cayirci, E.; , “A Survey on sensor networks,” Communication Magazine, IEEE, vol.40, no.8, pp. 102-114, Aug 2002. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1024422&isnumber=22021.

[2] Anatasi, Giuseppe, Marco Conti, Mario Di Francesco, and Andrea Passarella. “Energy Conservation in Wireless Sensor Networks: A survey (2008). Web. 29 March 2013. www.elsevier.com/locate/adhoc.

[3] “Bioenergy Harvester.” Voltree Power. Web. 29 March 2013. http://voltreepower.com/bioHarvester.html.

[4] Boyinbode, O,; Le, H.; Mbogho, A.; Takizawa, M.; Poliah, R.; , “A Survey on Clustering Algorithms for Wireless Sensor networks,” Network Based Information Systems (NBiS), 2010 13th International Conference on , vol., no., pp.1-10,26-29 April 2011 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5772405&isnumber=5772305 .

[5] Burri, N.; von Rickenbach, P.; Wattenhofer, R.; , “Dozer: Ultra-Low Power Data Gathering in Sensor Networks,” Information Processing in Sensor Networks, 2007. ISPN 2007. 6th International Symposium on, vol., no., pp.450-459, 25-27 April 2007

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4379705&isnumber=4379652.

[6] Cetintemel, Ugur; Flinders, Andrew; Sun, Ye; “Power-Efficient Data Dissemination in Wireless Sensor Networks,” MobiDE’03, September 19, 2003. ACM, 2003.

[7] Chen, Zhi-Gang; Zhao, Ming; Guo, Ying; Deng, Xiao-Heng; “Optimizing Energy-Efficient Reconstruction of Mobile Source in Sensor Networks,” International Journal of Distributed Sensor Networks, 5: 645-659. Taylor & Francis Group, LLC. 2009.

[8] Farooq, M.O.; Dogar, A.B.; Shah, G.A.; , “MR-LEACH: Multi-hop Routing with Low Energy Adaptive Clustering Hierarchy,” Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on , vol., pp.262-268, 18-25 July 2010 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5558064&isnumber=5558025.

[9] Hashmi, S.U.; Sarker, J.H.; Mouftah, H.T.; Georganas, N.D.; “An Efficient TDMA Scheme with Dynamic Slot Assignment in Clustered Wireless Sensor Networks,” Global Telecommunications Conference (GLOBECOMM 2010), 2010 IEEE, vol., no., pp.1-6,6-10 Dec. 2010 <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5683483&isnumber=5683069>

[10] Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H; "Energy-efficient Communication protocol for wireless Microsensor networks, "System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, vol., no., pp. 10 pp. vol.2, 4-7 Jan. 2000<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=926982&isnumber=20043>

[11] Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H; "An Application specific Protocol Architecture for wireless Microsensor networks," Wireless Communication, IEEE Transactions on, vol. 1, no. 4, pp. 660-670 Oct 2002 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1045297&isnumber=20043.

[12] Li Han; , "LEACH-HPR: An Energy efficient routing algorithm for heterogeneous WSN," Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on , vol.2, no., pp.507-511, 29-31 Oct 2010 http://ieeexplore.ieee.org/stamp.stamp.jsp?tp=&arnumber=1045297&isnumber=22403.

[13] Liu Ming;Cao, Jiannong, Chen, Guihai; Wang, Xiaomin,; "An Energy-Aware Routing Protocol in Wireless Sensor Networks," Open Access Sensors. 13 january 2009.

[14] Mladineo, Nenad, and S.Knezic. "Optimization of forest fire sensor network using GIS technology." Information Technology Interfaces, 2000. ITI 2000. Proceedings of the

22nd International Conference on IEEE, 2000

[15] Neglia, Giovanni, and Mouhamad Ibrahim. "Introduction to Network Simulator." INRIA Sophia-Antipolis – France. Web 29 March 2013. http://www.sop.inria.fr/members/giovanni.neglia/ns_course/slides/lec1.pdf.

[16] Qiao Li; Lingguo Cui; Baihai Zhang; Zhun Fan; , "A low energy intelligent clustering protocol for wireless sensor networks," Industrial Technology (ICIT), 2010 IEEE International Conference on , vol., no., pp.1675-1682, 14-17 March 2010 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5472629&isnumber=5472440.

[17] Regini, E.; Rosing, T.S.; , "An Energy efficient wireless communication mechanism for sensor node cluster heads," Intelligent Sensors, Sensor Networks and Information processing (ISSNIP), 2009 5th international conferenceon, vol. , no. , pp.403-408, 7-10 Dec. 2009

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5166998&isnumber=5166954.

[18] Sahin, YasarGuneri. "Animals are mobile biological sensors for forest fire detection." Sensors 7.12(2007):3084-3099.

[19] Stavrou, Eliana. " Wireless Sensor networks Pt 1:Introduction." Web Hosting Articles,

Devshed, 8th June 2005. Web. 30th March 2013.

http://webhosting.devshed.com/c/a/Web-Hosting-Articles/Wireless-Sensor-Networks-pt-1-Introduction/1/.

[20] Su, Xin; Choi, Dongmin; Moh, Sangman; Chung, Ilyong; “An Energy Efficient Clustering for Normal Distributed Sensor Networks.” Ministry of Education, Science Technology (MEST). 2009.

[21] Toh, C.-K. Ad Hoc Mobile Wireless Networks: Protocols and Systems. Upper Saddle River, NJ: Prentice hall PTR, 2002. Print.

[22] “Waspmote – Wireless Sensor Networks 802.15.4 ZigBee Mote.” Voltree Power. Web. 01 April 2013. http://www.libelium.com/products/waspmote.

[23] “Wireless Sensor Networks.” Wireless Sensor Networks. North Carolina State University, Web. 29th March 2013. http://www.ece.ncsu.edu/wireless/wsn.html.

[24] Wu Xinhua; Wang Sheng; , “Performance Comparison of LEACH and LEACH-C Protocols by NS2.” Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on, vol., no., pp. 254 258, 10-12 Aug 2010

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5571691&isnumber=5570815.

[25] Xu, Ya, John heidemann, and Deborah Estrin. “Geography-informed energy conservation for ad hoc routing.” Proceedings of the 7th annual international conference on mobile computing and networking.ACM, 2001.

[26] Yang, Cheng-Lung, WernhuarTarng, Keun-Rong Hsieh, and Mingteh Chen. “A Security mechanism for Clustered Wireless Sensor Networks Based on Elliptic Curve Cryptography.” ENewsletter Systems, Man and cybernetics Society. IEEE, Dec. 2010. Web. 28th March 2013.

http://www.mysmc.org/news/back/2010_12/main_article3.html.

[27] Younis, O.; Fahmy, S.; , “Distributed Clustering in Ad Hoc Sensor Networks: a hybrid, energy efficient approach,” INFOCOM 2004. 23rd Annual Joint Conference of the IEEE Computer and Communication Societies, vol.1, no., pp. 4 vol. (xxxv+2866), 7 11 March 2004

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6144227&isnumber=6144197.