



SOFTWARE REQUIREMENTS SPECIFICATION

**GREENHOUSE MONITORING with WIRELESS
SENSOR NETWORK**



**F. Aybike Avşaroğlu
Esra Gariboğlu
Osman Kaya
Önder Kalacı**

2010

Table of Contents

Table of Contents

1. Introduction

1.1. Problem Definition and Purpose of the Project

1.2. Scope

1.3. Definitions and Abbreviations

1.4. References

2. Overall Description:

2.1. Product Perspective

2.2. Product Functions

2.3. User Characteristics

2.4. Constraints

2.5. Assumptions and Dependencies

2.6. Apportioning Requirements

3. System Requirements

3.1. Interface Requirements

3.2. Functional Requirements

3.2.1. Programming the Sensor Nodes

3.2.1.1. Introduction/Purpose

3.2.1.2. Stimulus/Response sequence

3.2.1.2.1. Stimulus/Response Sequence for Sensor Nodes in Star Topology Based Protocol

3.2.1.2.2. Stimulus/Response sequence for Sensor Nodes in LEACH Protocol

3.2.1.3. Associated functional requirements

3.2.1.3.1. Functional requirement 1

3.2.1.3.2. Functional requirement 2

3.2.1.3.3. Functional requirement 3

3.2.2. Programming the Gateway

3.2.2.1. Introduction/Purpose of feature

3.2.2.2. Stimulus/Response sequence

3.2.2.2.1. Stimulus/Response sequence for Star Topology Based Protocol of the Gateway

3.2.2.2.2. Stimulus/Response sequence for LEACH Protocol of the Gateway

3.2.2.3. Associated functional requirements

3.2.2.3.1. Functional requirement of Programming the Gateway -1

3.2.2.3.2. Functional requirement of Programming the Gateway -2

3.2.3. Developing Web Application

3.2.3.1. Introduction/Purpose of feature

3.2.3.2. Stimulus/Response sequence

3.2.3.3. Associated functional requirements

3.2.3.3.1. Functional requirement of Web Application - 1

3.3.2.3.2. Functional requirement of Web Application - 2

3.5. Software System Attributes / Non-Functional Requirements

3.5.1. Reliability

3.5.2. Availability

3.5.3. Security

3.5.4. Maintainability

3.5.5. Portability

3.5.6. Performance

4. Object Data Model

4.1. Data Description

4.1.1. Data Objects

4.1.2. Complete Data Model and Relationships

4.1.3. Data Dictionary

5. Behavioral Models and Description

5.2. State Transition Diagrams

5.1. Description for Software Behavior

6. Planning

6.1. Team Structure

6.2. Basic Schedule

6.3. Process Model

1. Introduction

1.1. Problem Definition and Purpose of the Project

The most important factors for the quality and productivity of plant growth are temperature, humidity, light and the level of the carbon dioxide. Continuous monitoring of these environmental variables gives information to the grower to better understand, how each factor affects growth and how to manage maximal crop productiveness. The optimal greenhouse climate adjustment can enable us to improve productivity and to achieve remarkable energy savings - especially during the winter.

In modern greenhouses, several measurement points are required to trace down the local climate parameters in different parts of the big greenhouse to make the automation system work properly. Conventional monitoring systems using cabling would make the measurement system expensive and vulnerable. Moreover, the cabled measurement points are difficult to relocate once they are installed. Thus, a Wireless Sensor Network (WSN) consisting of small-size wireless sensor nodes equipped with radio and one or several sensors is an attractive and cost efficient option to build the required greenhouse measurement system [3].

The purpose of the project is to construct a greenhouse monitoring system using wireless sensor networks. The project's ultimate aim is to increase the amount of the products produced in a greenhouse without observing it whole day. The system periodically measures the temperature, light and humidity levels inside the greenhouse. When a critical change in the temperature, humidity or light level is detected, the greenhouse manager will be notified via e-mail and cell phone text message. Thus, the greenhouse manager will be able to react to the critical change as fast possible and may be able to prevent possible effects of the critical changes.

1.2. Scope

In the first part of the project, which is protocol based on the star topology [4], the sensor nodes only need to communicate with the gateway, not each other. The gateway takes the data packets from every node, construct them and send data to the database.

In the second part, which uses the LEACH protocol [2], firstly the sensor nodes choose cluster heads. Each sensor nodes group has a CH and sends data to it, the CHs does not make measurements. After a while the CHs should change dynamically, the sensor nodes chooses some other nodes as CHs. Then, CHs take data from sensor nodes, construct them (CHs clarify or ignore mistaken or corrupted data) and send the packets to the gateway. The gateway also checks the packets, construct the data, then send them to the database.

The web application part is common in the two parts. User logs into the web page, can see all the greenhouses belongs to his/her, chooses one of them. Then, the new page shows the instant measurements, user can choose past data, compare data, some statistics. Furthermore, there is information about sensor nodes' health status such as battery levels, signal qualities, percentage of erroneous packets. About the measurements and ratio of the dead sensors, there are different thresholds for different greenhouses, if one of these thresholds is exceeded, the user keeps informed about them via e-mail and/or SMS.

1.3. Definitions and Abbreviations

WSN: Wireless Sensor Network

GWSN: Greenhouse Wireless Sensor Network

CH: Cluster Head

LEACH: Low-Energy Adaptive Clustering Hierarchy

IEEE: Institute of Electrical and Electronics Engineering

1.4. References

[1]	IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
[2]	Heinzelman, WRH, Chandrakasan, AC, & Balakrishnan, HB. (2000). Energy-efficient communication protocol for wireless microsensor networks
[3]	Ahonen, TA, Virrankoski, RV, & Elmusrati, ME. Greenhouse monitoring with wireless sensor network.
[4]	Greenhouse guide. (Referred 28 11 2010). [Online]. Available: http://www.littlegreenhouse.com/guide.shtml
[5]	M. Mancuso and F. Bustaffa, "A Wireless Sensors Network for Monitoring Environmental Variables in a Tomato Greenhouse," presented at 6th IEEE International Workshop on Factory Communication Systems in Torino, Italy, June 28-30, 2006
[6]	USB Gateway Interface Data Sheet(Referred 29.11.2010)[Online].Available: http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=144%3Amib520cb
[7]	H. Liu, Z. Meng and S. Cui, " A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses," presented at Wireless Communications, Networking and Mobile Computing 2007 (WiCom 2007), International Conference on 21-25 Sept. 2007 Page(s): 2344 – 2347.

2. Overall Descriptions

2.1. Product Perspective

In this project, there are three main tasks to be accomplished. Firstly, programming the sensor nodes, then programming the gateway, where all the information is gathered from the sensor nodes, and lastly informing the manager of the greenhouse with the information at the gateway, via both a user friendly web site and sending SMS.

Programming the Sensor Nodes:

It involves periodically measuring the temperature, light and humidity levels and sending these measurements to the gateway. However, this task should be completed in such a way that the sensor nodes use least possible energy. Thus, in this task, we will be implementing power conservative routing algorithms for sensor networks [7].

Programming the Gateway:

This task consists of three major operations. Firstly, if there is an error in the data coming from the sensor nodes, fix it. Then, consolidate the data. Lastly, save the data to an SQL database on the computer which the gateway is connected.

Develop a web application:

This application will provide a user friendly interface to the greenhouse manager, such that the manager will be able to see meaningful information about his/her greenhouse almost instantly. The manager will also be able to see the past information and the comparisons of them with the current information.

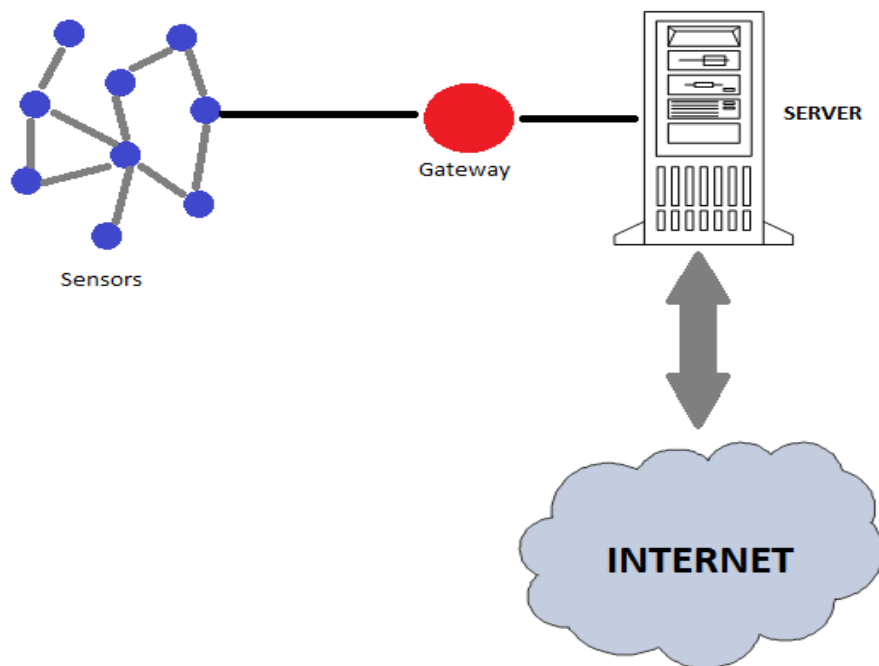


Figure 1 - Wireless Sensor Network

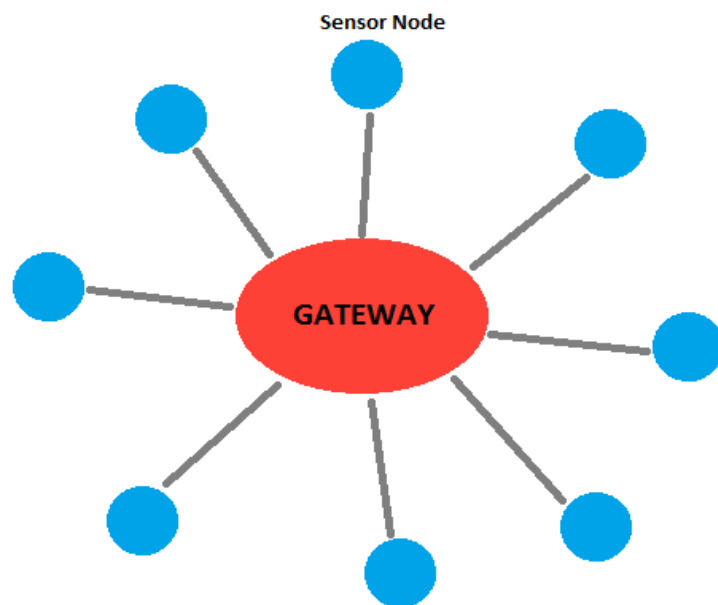


Figure 2 - Star Topology

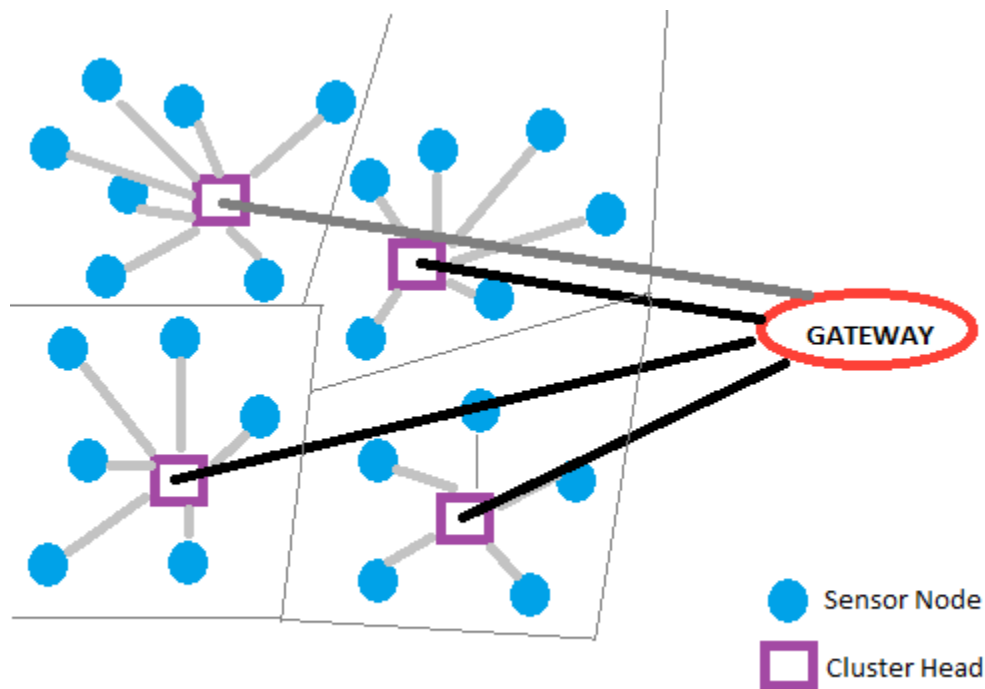


Figure 3 - LEACH Protocol

2.2. Product Functions

- Accurately measuring the environment's temperature, light and humidity.
- Transmitting the measured data to the gateway.
- Evaluating the data and warning the user in an unexpected situation.
- Storing the collected data in a database.
- Making it reachable through a user friendly website.

2.3. User Characteristics

The users are greenhouse managers who are supposed to have domain knowledge, such as knowing the optimal and critical temperature, light and humidity levels for specific crops. Other than that basic computer knowledge will be enough.

2.4. Constraints

The most important constraint is hardware related. Generally, a WSN consists of hundreds or maybe thousands of sensors. We will not be able to test our software on such a level since we have 30 Processor/radio boards and 20 Sensor boards for this project. However, these are absolutely enough for proof of concept implementation.

2.5. Assumptions and Dependencies

No assumptions are made.

2.6. Apportioning Requirements

Some enhancement to the LEACH protocol or an entire new protocol can be made after final product is finished. Except from defined website, some design and functionality for the website can be delayed.

3. System Requirements

3.1. Interface Requirements

The gateway and sensors will be programmed using MIB520CB IRIS / MICA USB PC Interface. This interface is specified in [6].

3.2. Functional Requirements

3.2.1. Programming the Sensor Nodes

3.2.1.1. Introduction/Purpose

Sensor nodes are programmed according to two different routing protocols. First one is the LEACH protocol and the second one is a protocol based on star topology. The algorithms identify sensor nodes' communications with gateway or/and each other.

Since the implementation of the second protocol is less complicated compared to the implementation of LEACH protocol, firstly, we will be implementing it in order to gain experience with sensor nodes. Then, we will implement the LEACH protocol. Communication between the sensor nodes and the base station is expensive, and there are no "high-energy" nodes through which communication can proceed. Therefore, automated methods of combining or aggregating the data into small set of meaningful information, combining several unreliable data measurements to produce a more accurate signal by enhancing the common signal, and reducing the uncorrelated noise is required. In addition to these, LEACH is able to distribute energy dissipation evenly throughout the sensors, doubling the useful system lifetime for the network [5].

In LEACH implementation, some of the sensor nodes will be CHs, which will have direct communication with the gateway, and the other sensors will communicate with the gateway through these CHs. The clusters and the CHs will be decided dynamically in all given time intervals. Each node makes its decision about whether to

be a cluster-head independently of the other nodes in the network and thus no extra negotiation is required to determine the CHs.

3.2.1.2. Stimulus/Response sequence

Since the implementation of the sensor nodes in star topology based protocol and LEACH protocol are fairly different from each other, the stimulus/response sequences of them will be examined separately.

3.2.1.2.1. Stimulus/Response Sequence for Sensor Nodes in Star Topology Based Protocol

- The sensors measure at regular intervals and send the data to the gateway.
- Then the sensor sleeps until the next time it sends data to gateway.
- Loops forever.

3.2.1.2.2. Stimulus/Response sequence for Sensor Nodes in LEACH Protocol

- Some of the sensor nodes elect themselves as CH for this round (There are some constraints and randomization algorithm about who can elect itself as a CH)
- Each node that elects itself a cluster-head for the current round broadcasts an advertisement message to the rest of the nodes
- The non-cluster-head nodes must keep their receivers on during this phase of set-up to hear the advertisements of all the cluster-head nodes
- Each non-CH decides the cluster to which it will belong for this round and informs the CHs that it is a member of that cluster (This decision is based on the received signal strength of the advertisement).
- The CHs receive all the messages for sensor nodes that would like to be included in the cluster. Based on the number of nodes in the cluster, the CHs create a schedule telling each sensor node when it can transmit.
- The CHs must keep its receiver on to receive all the data from the nodes in the cluster , however , the non-CH nodes can sleep for the next time it sends data to

CH. Then, CHs send compressed and consolidated data, which is free of error, to gateway. The data also includes the total number of alive sensor nodes associated with this CH.

- After each election, CHs put battery level information of each sensor to the packets.
- After a certain amount of time, which is determined a priori, the next round begins with each node determining if it should be a CH for this round and advertising this information.
- Loops forever.

3.2.1.3. Associated functional requirements

3.2.1.3.1. Functional requirement 1

Accurately measuring the environment's temperature, light and humidity.

3.2.1.3.2. Functional requirement 2

Sending the data to the gateway.

3.2.1.3.3. Functional requirement 3

Sending the battery level information.

3.2.2. Programming the Gateway

3.2.2.1. Introduction/Purpose of feature

Gateway gets data coming from any sensor nodes (CHs for LEACH), and checks the data. If there is not any problem about the measurements, and packets, data is saved to an SQL database. Otherwise, if the data is unrealistic, packet is corrupted or a sensor does not send any packets, gateway handles the problems in different ways.

3.2.2.2. Stimulus/Response sequence

3.2.2.2.1. Stimulus/Response sequence for Star Topology Based Protocol of the Gateway

- Gateway gets the data at any time.
- Then, gateway checks the data, if there is not any problematic data, saves it to an SQL database.
- If there is some problematic data, gateway handles it. There could be 3 possible problematic data types. Firstly, the data could be unrealistic, such that the temperature is 200 C degrees. Secondly, the data that should have come may not have been received. In both cases, greenhouse manager is informed, for the repeated occurrences. (i.e. if the total number of working sensor nodes goes below a threshold). Thirdly, the data packet that has been received may be corrupted (i.e. missing bytes in the packet). In this case, the gateway discards that packet.
- Loops forever.

3.2.2.2.2. Stimulus/Response sequence for LEACH Protocol of the Gateway

- Gateway gets the data at any time.
- Then, the gateway further checks the data, which has already been checked at the CHs. If there is not any problematic data, saves the data to an SQL database.
- If there is a problematic data, the gateway handles it. There could be 1 possible problematic data type. The data packet, that has been received, may be corrupted (i.e. missing bytes in the packet). The gateway discards that packet.
- Loops forever.

3.2.2.3. Associated functional requirements

3.2.2.3.1. Functional requirement of Programming the Gateway -1

Getting the data coming from the CHs/sensor nodes.

3.2.2.3.2. Functional requirement of Programming the Gateway -2

Consolidating the data and saving them to an SQL database.

3.2.3. Developing Web Application

3.2.3.1. Introduction/Purpose of feature

Greenhouse managers can choose one of their greenhouses, and see the instant measurements. Furthermore, the managers can see the past measurements, averages, and comparisons. If there is an abrupt change in the measurements (temperature, humidity, light), manager would be informed via e-mail and/or SMS. User can view the battery levels, signal qualities, percentage of erroneous packets of the sensor nodes. Moreover, when the dead sensor nodes ratio exceeds a threshold, the manager is informed via an e-mail and/or an SMS. The aim is to provide real-time greenhouse environment observation to the managers.

3.2.3.2. Stimulus/Response sequence

- User shall log in to the main page with User ID and Password, to see his/her greenhouse list.
- User shall click one of the greenhouses, then the new page loads, on which, instant temperature, humidity and lightening measurements are shown.
- User shall enter a date, and see the average measurements for the date, can make some comparisons.
- User shall check the health statuses of the sensor nodes, such as their battery levels, signal qualities.
- User is informed via an e-mail and/or an SMS when the ratio of dead sensor nodes exceeds a threshold.

3.3.2.3. Associated functional requirements

3.3.2.3.1. Functional requirement of Web Application - 1

Showing the greenhouses list for the user.

3.3.2.3.2. Functional requirement of Web Application - 2

Showing the instant and past measurements, their comparisons, and health statuses of sensor nodes, of the chosen greenhouse.

3.3.2.3.3. Functional requirement of Web Application - 3

Sending an alert message to the user via e-mail or SMS, whenever the instant measurements or the ratio of dead sensor nodes exceeds a threshold.

3.5. Software System Attributes / Non-Functional Requirements

3.5.1. Reliability

A reliable network protocol is one that provides reliability properties with respect to the delivery of data to the intended recipient. The protocol we implement will be reliable in terms of data accuracy and time accuracy. Data accuracy is checked to make sure sensors do not send wrong data. Time accuracy is achieved by programming the hardware, sensor nodes.

3.5.2. Availability

The web interface and SMS request interface will always be available. For network to be available, batteries of sensors should be replenished when 50% of the network is unavailable.

3.5.3. Security

The only security constraint will be user information security. Greenhouse managers will be able to choose sharing or not sharing their data. Also, when the user does not remember his log in password, a request for a new password can be made. Since an attack to the greenhouse network is not expected we do not need to implement security constraints on the network.

3.5.4. Maintainability

After some time, the batteries of sensors will run out. The power a sensor consumes is inversely proportional to the square of distance to the gateway. In star

topology, this will cause farther sensors to go down sooner than the closer sensors. In LEACH topology, batteries will almost finish at the same time. In our case, batteries should be replaced yearly.

3.5.5. Portability

Though there are portability issues with embedded programming, the code written for sensors will be mostly portable, being written in nesC. There are no issues with the other parts of the project.

3.5.6. Performance

The project mostly focuses on energy consumption and availability of the network as a whole. Therefore, the performance is not the most important constraint. Performance is mostly not considered for the sake of power saving.

4. Object Data Model

4.1. Data Description

The data collected from sensors are stored in an SQL database. These data includes temperature, humidity, light measurements, battery condition and erroneous data transmission rates of each sensor node, and broken or dead sensor information. In order to implement a robust system an object data model system for the database is needed.

4.1.1. Data Objects

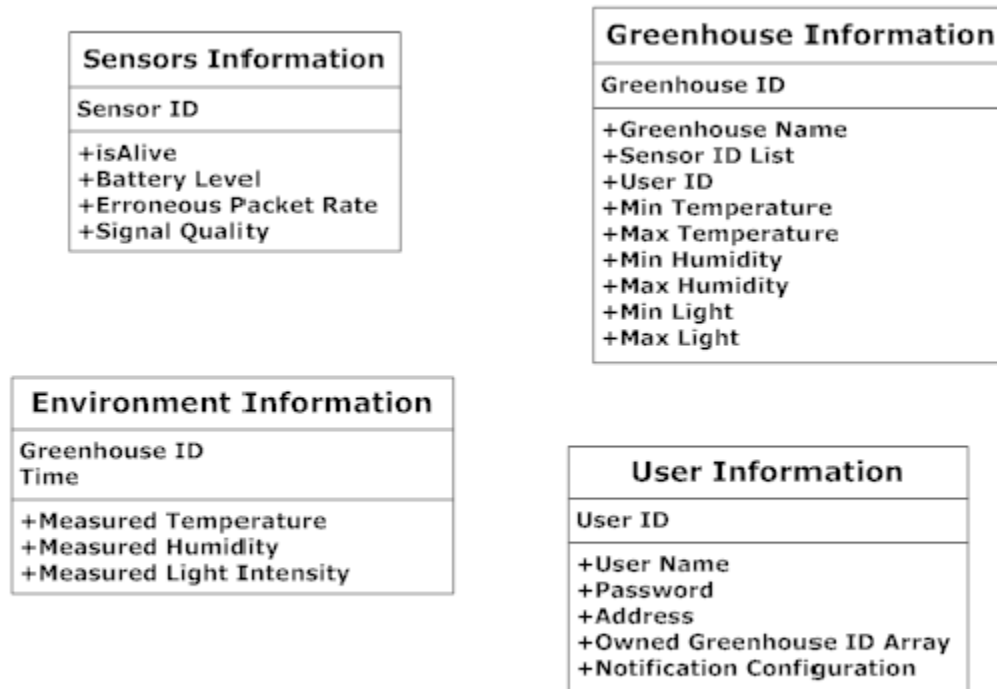


Figure 4 - Data Objects

4.1.2. Complete Data Model and Relationships

Since there are a few data objects in this project, the complete data model is equivalent to this single level relation among the objects.

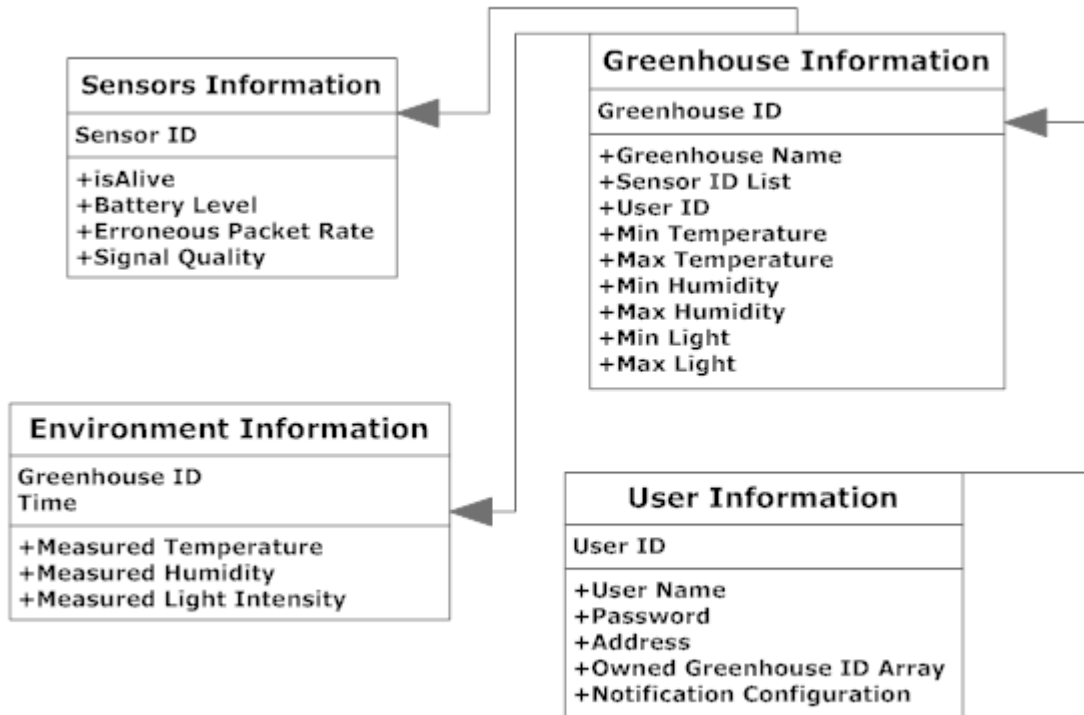


Figure 5 - Complete data model and relationships

4.1.3. Data Dictionary

isAlive: Boolean value which keeps the information whether the sensor is alive.

UserId: A unique id for each greenhouse manager.

GreenHouseId: A unique id for each greenhouse.

SensorId: A unique id for each sensor node.

5. Behavioral Models and Description

5.1. Description for software behavior

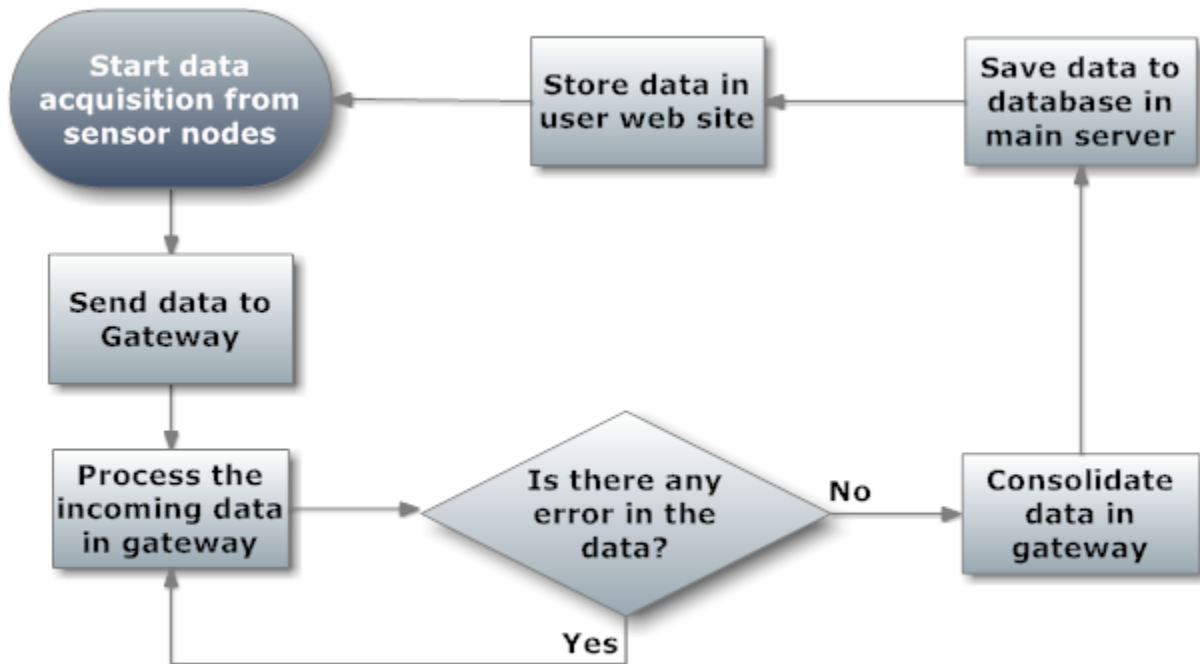


Figure 6 – Data flow diagram of greenhouse monitoring with WSN

A prototype greenhouse monitoring system will be implemented using wireless sensor networks. Pre-programmed sensor nodes will be placed in several points of a greenhouse. These sensors will periodically measure the temperature, light and humidity levels inside the greenhouse and send this information to the main server in a remote control center. When a critical change in the temperature, humidity or light level is detected, the greenhouse manager will be notified via e-mail or cell phone text message.

5.2. State Transition Diagrams

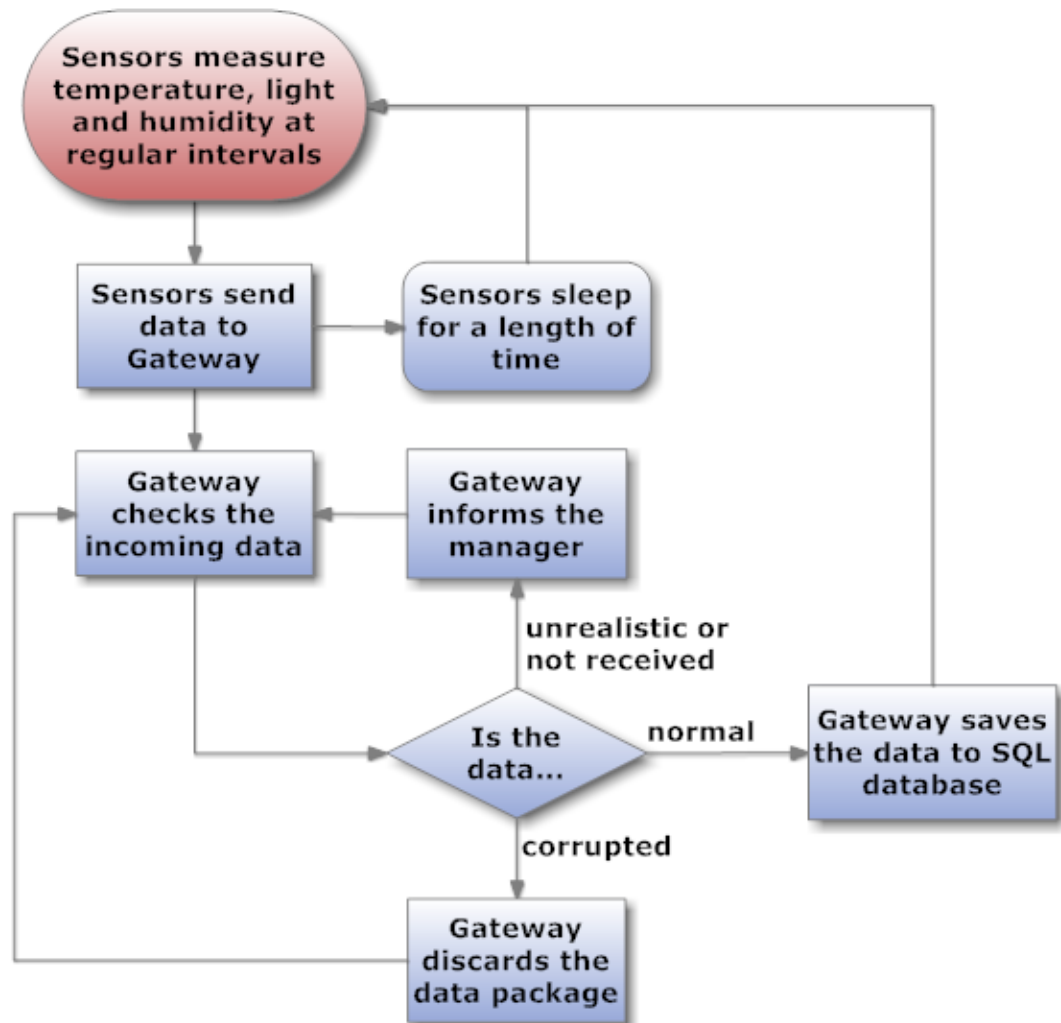


Figure 7 – Data flow diagram for Star Topology Protocol

Star topology includes a single hop to the gateway. Gateway serves to communicate between nodes. Nodes cannot send data to each other directly. Its advantages are the lowest power consumption and easily scalable. On the other hand, there are some disadvantages that cannot be ignored. It is not very reliable as one point of failure and in addition to this, there is no alternate communication paths.

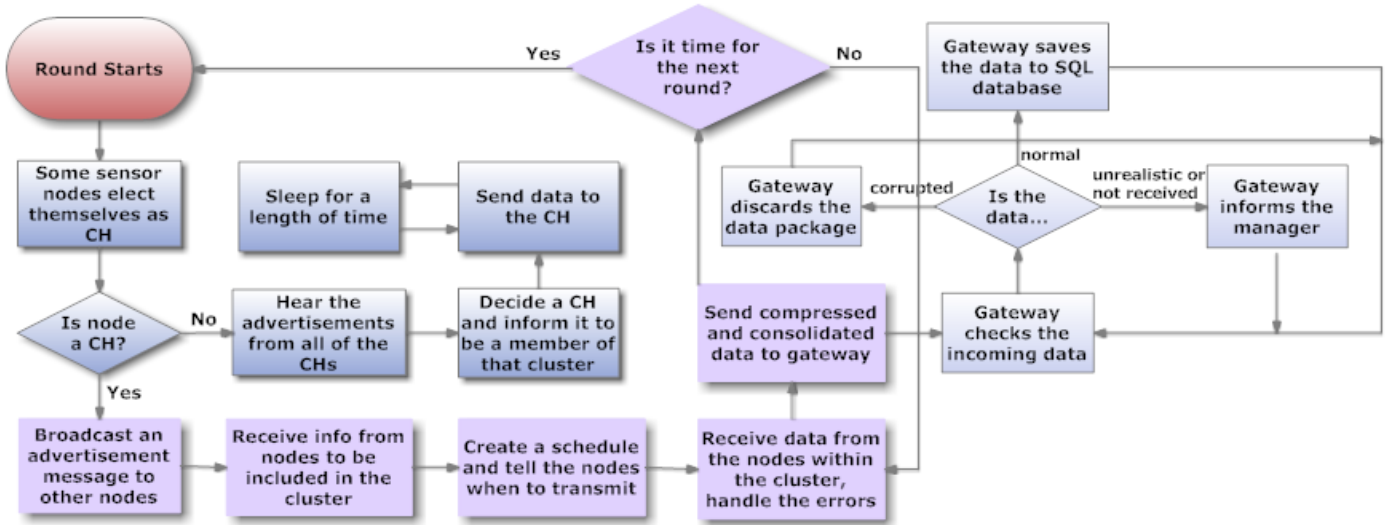


Figure 8 - Data flow diagram of LEACH protocol

Sensors elect themselves to be local cluster-heads at any given time with a certain probability. These cluster-head nodes broadcast their status to the other sensors in the network. Each sensor node determines to which cluster it wants to belong by choosing the cluster-head that requires the minimum communication energy. Once all the nodes are organized into clusters, each cluster-head creates a schedule for the nodes in its cluster. This allows the radio components of each non-cluster-head node to be turned off at all times except during its transmit time, thus minimizing the energy dissipated in the individual sensors. Once the CH has all the data from the nodes in its cluster, the cluster-head node aggregates the data and then transmits the compressed data to the gateway.

6. Planning

6.1. Team Structure

We are 4 people working on this project: F.Aybike Avşaroğlu, Osman Kaya, Esra Gariboğlu and Önder Kalacı. We are all senior students at Middle East Technical University, Computer Engineering department.

6.2. Basic Schedule

December 5th	SRS completion
January	SDD completion and learning the tools
February	Start of star topology based protocol implementation
March	Finishing and testing the star topology based protocol implementation
March	Start of LEACH protocol.
April	Finishing the phase and testing.
April	Implementing the website and other user interface tools.
May	Final testing and finishing the project.

6.3. Process Model

Our process model is the “Waterfall Model”, since we distinguished all the development phases such as requirements analysis and definition, system and software design, implementation and testing. The documentations will be produced at each phase. This document is the first one, for the requirements analysis.

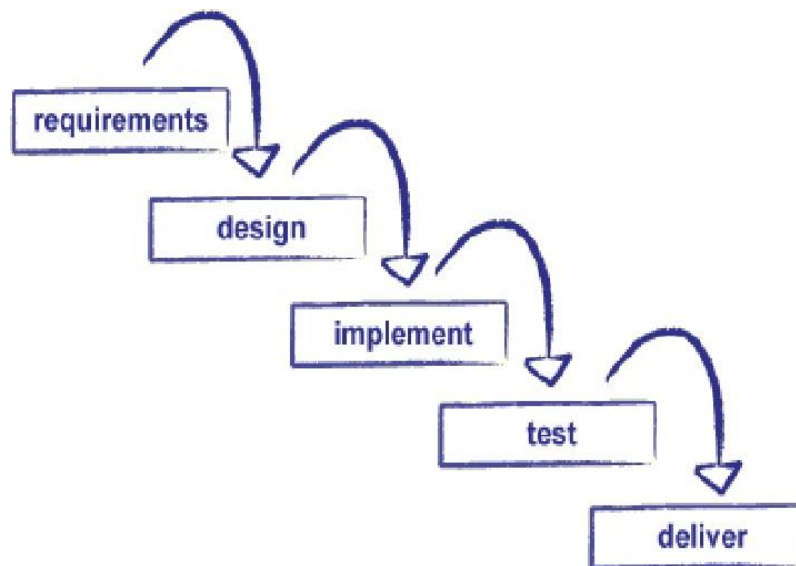


Figure 9 - Waterfall Process Model