Module TII:

Timing and Implementation Issues in Wireless Sensor Networks

DRAFT

F. Khan and W. Li (TSU)

Module Contents

Part 1. WSN Time Synchronization
Part 2. WSN Data Fusion and Aggregation
Part 3. WSN Implementation

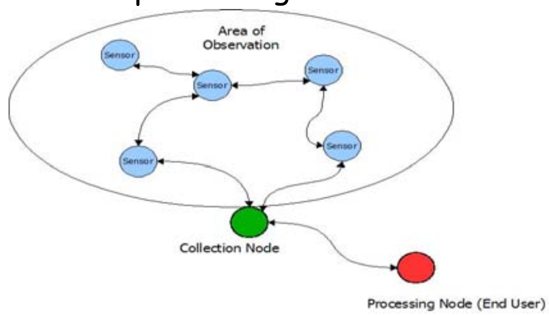
Part 1: WSN Time Synchronization

1.1. Overview

- Scope
  - Time Synchronization in wireless sensor networks and security measures
- Intent
  - Survey and evaluation of current time synchronization protocols and the effectiveness of their security measures

1.1. Overview

- Wireless Sensor Networks
  - Network of multiple resource-constrained sensor nodes that monitor and forward information
  - Typically has a centralized node that collects the data and another node that will be used for processing the data.



1.1. Overview

- Why is Time Synchronization Important
  - Location and proximity of siblings
  - Intranetwork coordination
  - Proper functioning of security measures
  - Maintain ordering of messages
  - Use of Time Division Multiplexing (TDMA)
  - Energy efficiency

1.1. Overview

- Why not NTP or GPS?
  - The majority of sensor nodes are designed with minimal resources available and consumed
    - ✓ NTP requires highly accurate clocks
    - ✓ GPS requires an additional receiver for the GPS transactions
      - Some networks with less restrictions may be able to afford a GPS sensor infrastructure.
  - Additional details are available in chapter 1 of the Handbook of Sensor Networks: Algorithms and Architectures: *Time Synchronization and Calibration in Wireless Sensor Networks* by Kay Römer, Philipp Blum, Lennart Meier available from http://www.vs.inf.ethz.ch/publ/papers/wsn-time-book.pdf

1.1. Overview

- Clock Synchronization Basics
  - Clock Offset
    - Difference between clocks at 2 nodes $\delta_{AB} = C_A(t) - C_B(t)$
  - Clock Skew
    - Change in offset over time
      - One clock running faster than another $\eta_{AB} = \frac{\partial C_A(t)}{\partial t} - \frac{\partial C_B(t)}{\partial t}$
  - Drift Error
    - Random changes in clock frequency
      - Temperature differences
      - Aging of the hardware

$$\lambda_{AB} = \frac{\partial^2 C_A(t)}{\partial t^2} - \frac{\partial^2 C_B(t)}{\partial t^2}$$

1.1. Overview

- Effects of Unsynchronized clocks
  - Invalid observations
  - Inefficient/ineffective coverage
    - ✓Areas left uncovered during certain timeslots
  - Disabled communications architecture (worst case

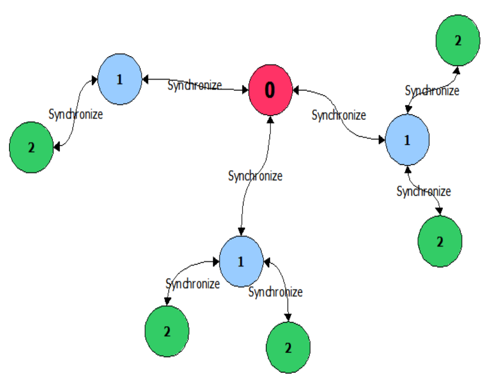
1.2 WSN Time Synchronization Protocol

- Create a spanning tree with a designated "root" node
  - Each pair of nodes can be considered a root-child, where the child becomes the root for the next node
  - Child requests synchronization from the root, which responds with an acknowledgement message
    - ✓ The message data is the departure and arrival time of each, so the child, upon receipt of the last message can calculate its clock offset relative to the parent
- Attacks
  - Compromised node can cause it's child node to calculate an incorrect offset, and this will trickle down the tree

4/15/2015 Module TII in WSN 9

1.2 WSN Time Synchronization Protocol

- Typical sender-receiver synchronization
- Hierarchical
  - Root nodes at each level synchronize with child nodes



4/15/2015 Module TII in WSN 10

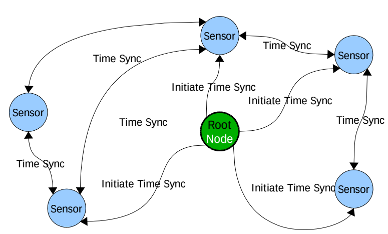
1.2 WSN Time Synchronization Protocol

- Flooding Time Synchronization Protocol
  - Root node is elected (lowest ID wins) and is the node that sends out the synchronization message
  - If no synchronization message is received after a preset time period, a node can elect itself root and initiate the synchronization
    - Self healing tree structure in case of node failures
  - Attacks
    - Any compromised node could declare itself root since the lowest ID always wins
      - Node then sends incorrect time synchronization data that corrupts the entire tree

1.2 WSN Time Synchronization Protocol

- Flooding Time Synchronization Protocol
  - Root node broadcasts time synchronization initiation message
  - Non-root sensors coordinate root message amongst neighbors
    - Non-root sensors use multiple neighbors timestamps to determine their own time.

Area of Observation



Part 2: WSN Data Fusion and Aggregation

2.1. WSN Data Fusion

- A data fusion node collects the results from multiple nodes.
- It fuses the results with its own based on a decision criterion.
- Sends the fused data to another node/base station.
- Advantages:
  - Reduces the traffic load.
  - Conserves energy of the sensors.

Key Concepts in Data Fusion

- Three questions needs to be addressed:
  - ✓ First, at what instance does a node report a sensed event?
  - ✓ Second, how does a node fuse multiple reports into a single one?
  - ✓ Third, what data fusion architecture to use?

❑ Reporting

- Periodical reporting: Sensor nodes periodically send reports to the base station.
- Base station inquiry response reports: the BS queries sensors in specific regions for current sensed information.
- Event triggered reports: The occurrence of a certain event can trigger reports from sensors in that particular region.

❑ Fusion Decision

- Voting: the oldest and most widely used fusion decision method.
- Fusion node arrives at a consensus by a voting scheme like:
  - Majority voting
  - Complete Agreement
  - Weighted voting
- The popularity of voting arises from its simplicity and accuracy.
- Other fusion decision algorithms include probability-based Bayesian Model and stack generalization.

Fusion Architecture

Centralized:

- Simplest
- A central processor fuses the reports collected by all other sensing nodes.
- Advantage: Erroneous report(s) can be easily detected.
- Disadvantage: inflexible to sensor changes and the workload is concentrated at a single point.

Fusion Architecture

Decentralized :

- ✓ Data fusion occurs locally at each node on the basis of local observations and the information obtained from neighboring nodes.
- ✓ No central processor node.
- ✓ Advantages:
  - scalable and tolerant to the addition or loss of sensing nodes or dynamic changes in the network.

❑ Fusion Architecture

➢ Hierarchical:

✓Nodes are partitioned into hierarchical levels.

✓The sensing nodes are at level 0 and the BS at the highest level.

✓Reports move from the lower levels to higher ones.

✓Advantage:

- Workload is balanced among nodes

❑ Fusion level and energy consuming

➢ The key idea is to combine data from different sensors to eliminate redundant transmissions, and provide rich, useful information of the environment being monitored, as shown in the following fig.

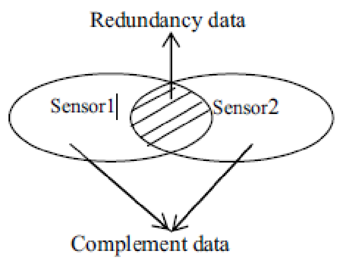


Figure 6. Construction of data fusion

❑ Data Fusion in wireless sensor network

- The simplest data aggregation function is duplicate suppression if two sources both send the same data, fusion node will send only one of these forward.
- Other aggregation functions could be max, min, or any other function with multiple inputs.

❑ Data Fusion in wireless sensor network

- A three-level hierarchy is widely accepted by data fusion practitioners.
  - ✓ data-level fusion,
  - ✓ feature-level fusion
  - ✓ and decision-level fusion

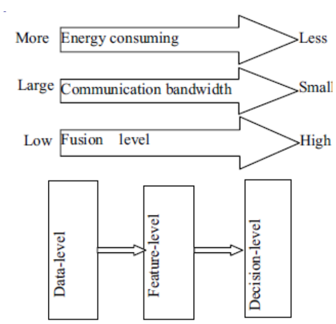


Figure 7. Relationship between fusion level and energy consuming

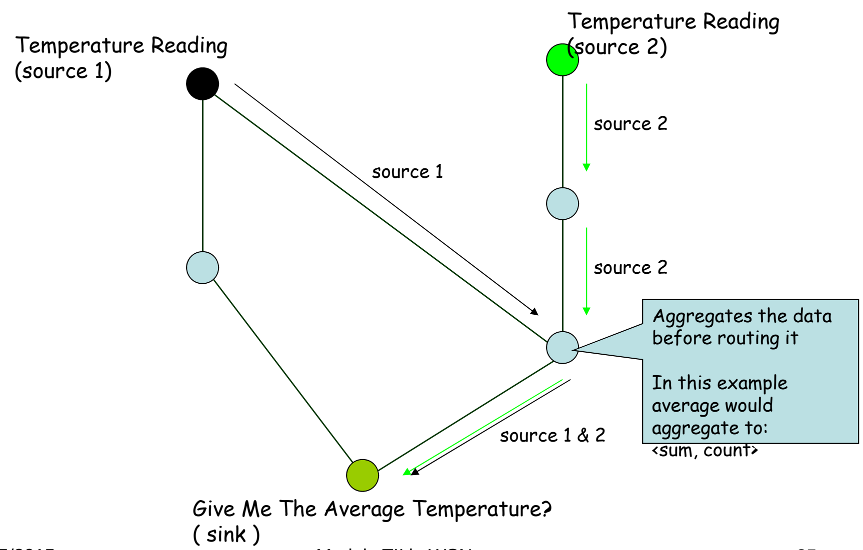
❑Fusion technique

- ➢ sensor fusion is concerned with distributed detection, target identification, decision fusion. Various techniques such as least square method, Bayesian method, Dempster-Shafer, Fuzzy logic has been proposed.
- ➢ All of these techniques used for fusion are not as simple as max, min, however, sensor node are so limited in computational capacities that some technique may be difficulty for sensor nodes implementing

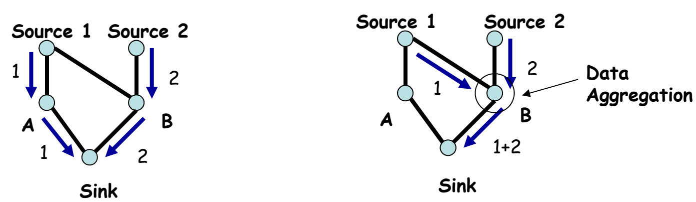
Part 2: WSN Data Fusion and Aggregation

2.2. WSN Data Aggegation

- ❑ Redundant Data/events
- ❑ Some services are amenable for in-network computations.
  - "The network is the sensor"
- ❑ Communication can be more expensive than computation.
- ❑ By performing "computation" on data en route to the sink, we can reduce the amount of data traffic in the network.
- ❑ Increases energy efficiency as well as scalability
  - The bigger the network, the more computational resources.



Transmission modes AC vs DC



a) Address-Centric (AC) Routing (no aggregation)

b) Data-Centric (DC) Routing (in-network aggregation)

❑ Theoretical Results on Aggregation

- Let there be k sources located within a diameter X, each a distance d_i from the sink. Let N_A, N_D be the number of transmissions required with AC and optimal DC protocols respectively.
  - ✓ The following are bounds on N_D:

    $$N_D \leq (k-1)X + min(d_i)$$
    $$N_D \geq min(d_i) + (k-1)$$

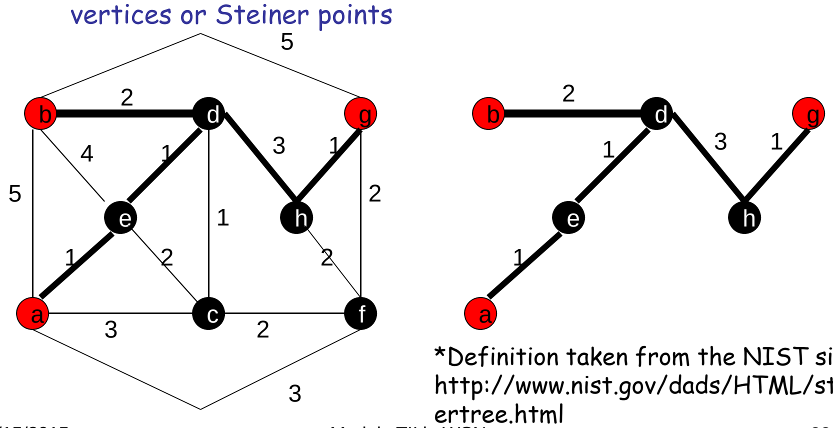
  - ✓ Asymptotically, for fixed k, X, as d = min(d_i) is increased,

    $$lim_{d\to\infty} \frac{N_D}{N_A} = \frac{1}{k}$$

4/15/2015 Module TII in WSN 27

❑ Optimal Aggregation Tree
Steiner Trees

- *A minimum-weight tree connecting a designated set of vertices, called terminals, in a weighted graph or points in a space. The tree may include non- terminals, which are called Steiner vertices or Steiner points*



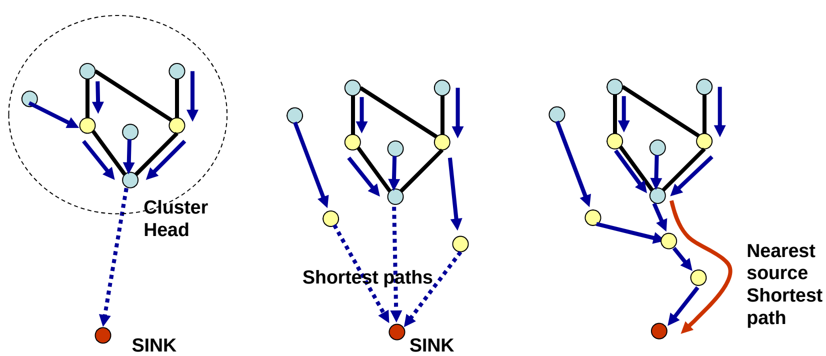
*Definition taken from the NIST site. http://www.nist.gov/dads/HTML/steinertree.html

4/15/2015 Module TII in WSN 28

❑ Aggregation Techniques

- Center at Nearest Source (CNSDC): All sources send the information first to the source nearest to the sink, which acts as the aggregator.
- Shortest Path Tree (SPTDC): Opportunistically merge the shortest paths from each source wherever they overlap.
- Greedy Incremental Tree (GITDC): Start with path from sink to nearest source. Successively add next nearest source to the existing tree.

Aggregation Techniques



a) Clustering based CNS b) Shortest Path Tree c) Greedy Incremental

❑ Data Aggregation

- Data aggregation can result in significant energy savings for a wide range of operational scenarios
- Although NP-hard in general, polynomial heuristics such as the opportunistic SPTDC and greedy GITDC are near-optimal in general and can provide optimal solutions in useful special cases.
- The gains from aggregation are paid for with potentially higher delay.

Part 3: WSN Implementation

2.1. WSN Data Fusion

- ❑ A data fusion node collects the results from multiple nodes.
- ❑ It fuses the results with its own based on a decision criterion.
- ❑ Sends the fused data to another node/base station.
- ❑ Advantages:
  - Reduces the traffic load.
  - Conserves energy of the sensors.

Part 3: WSN Implementation

3.1. Introduction

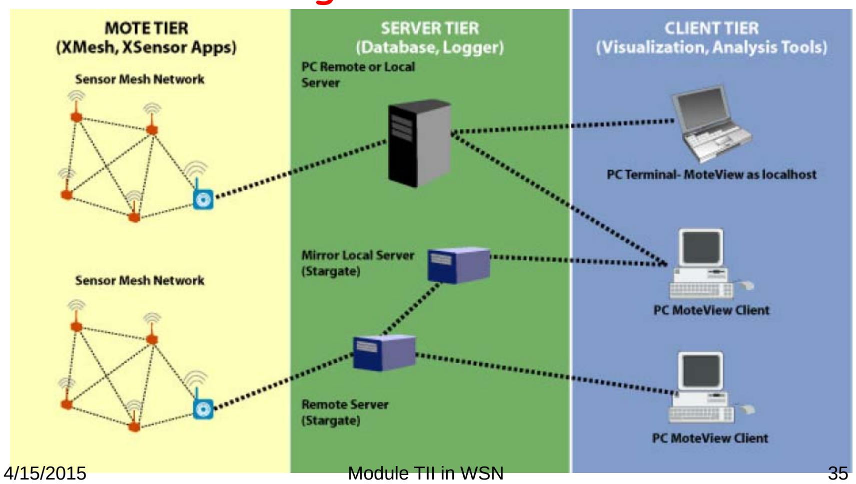
- Understand and Configure an environment to facilitate the Mote and Client tier of the MoteWorks framework.
- Deploy a network of motes to collect data about a selected environment.
- Modify the mote tier to collect, record and tabulate only light and temperature data in an environment.
- Implement an event-driven approach to the sensors behavior.

3.1. Introduction

- The main characteristics of a WSN include:
  - Ability to cope with node failures
  - Mobility of nodes
  - Dynamic network topology
  - Heterogeneity of nodes
  - Scalability to large scale of deployment
  - Ability to withstand harsh environmental conditions
  - Ease of use
  - Unattended operation

3.1. Introduction

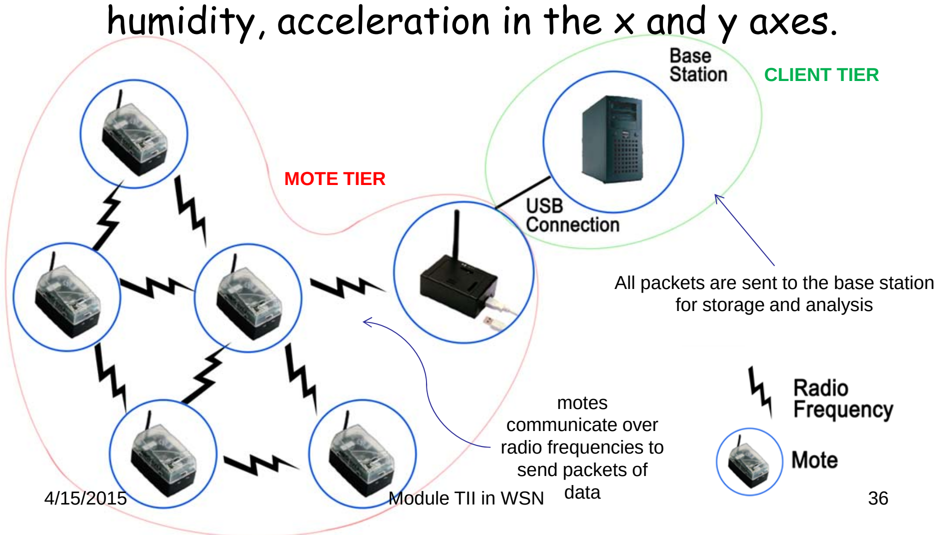
- Understanding WSN



4/15/2015 Module TII in WSN 35

3.1. Introduction

- Motes pick up environmental data about light, atmospheric pressure, temperature, humidity, acceleration in the x and y axes.



4/15/2015 Module TII in WSN 36

3.2. WSN Implementation

- The implementation of the sensor network required specific hardware supplied by MEMSIC. The MEMSIC's Professional Kit consists of the following hardware:
  - 6 Sensor Nodes
  - 1 Base Station
  - 1 Processor/Radio Module
  - 1 Data Acquisition Board*
  - 1 USB Programming Board

3.2. WSN Implementation

- Additional hardware that is required for the set up of the network includes:
  - Batteries to power the sensor nodes.
  - A laptop computer to connect the base station to. This computer will be used for data collection and analysis. It will also be used to write the programs for the notes as well as upload these programs to the nodes. The laptop should have the following minimum requirements:
  - 1 GB of free space in the destination drive.
  - 550 MB of free space in the C drive.
  - USB extension cables to connect the devices to the laptop.

3.2. WSN Implementation

- The implementation of the sensor network requires also specific software supplied partly by MEMSIC. The sensor networking kit includes several software packages that are needed to connect the motes, collect and analyze data, write and compile programs for the nodes, and upload these programs to the nodes.
- The software (with the exception of XMesh) works successfully only on a machine with Windows XP Professional Service Pack 2 (XP-Pro SP2).

3.2. WSN Implementation

Software	Function
XMesh	Multi hop networking protocol installed on each node.
TinyOS	An event-driven OS for wireless sensor networks. It also provides tools for debugging.
NesC compiler	An extension of the C-language designed for TinyOS.
Cygwin	A Linux-like environment for Windows.
XSniffer	Network Monitoring Tool for the RF environment
MoteConfig	GUI environment for Mote Programming and Over the Air Programming (OTAP).
MoteView	An interface between a user and a deployed network of wireless sensors. Provides the tools to simplify deployment and monitoring.
Programmer's Notepad 2	A simple IDE for nesC code.

3.2. WSN Implementation

- In Experiment #1, #2 & #3, the results were controlled by:

- using air-conditioned rooms in our test building to control the temperature and humidity.

-the lights were turned off/on in specific rooms.

These controls were used to predict and validate our results.

3.2. WSN Implementation

- **Experiment #1:**
  - Only four motes were used for the initial setup. The motes were powered up and placed at different locations. The location of the motes and the approximate distance between each and the base station is given in the table below.

		Distance from (in meters)				
Node ID	Location (Room #)	Base Station	7251	7252	7253	7254
Base Station	131		5	15	35	15
7251	131	5		10	15	20
7253	132	15	10		10	15
7254	Lobby	30	25	15		15
7256	Copy Room	15	10	10	15	

3.2. WSN Implementation

➢ **Results from Experiment#1**

✓ COLLECTING LIVE DATA: A sample set of results is shown in table. Given the environment, the data collected over the time period was very consistent. The data collected included humidity, humidity-temperature, present-temperature, pressure, light, horizontal acceleration, and vertical acceleration. All data were displayed in standard engineering units.

ID	Humidity (%)	Humidity-temperature (C)	Pressure-temperature (C)	Pressure (mba)	Light (Lux)	Horizontal acceleration (m/s^2)	Vertical acceleration (m/s^2)	Time 02/22/2012 PM
7651	45.65	23.71	23.87	1006.3	308.89	3.332	-30.772	5:05:06
7653	48.79	24.89	24.91	1005.89	294.17	20.384	-24.5	5:05:06
7654	48.53	23.07	23.25	1005.69	514.3	0.196	-0.196	5:05:06
7656	43.97	24.32	24.78	1005.59	285	0	0	5:05:06

3.2. WSN Implementation

❑ **Experiment #2:**

➢ The purpose of experiment #2 was to successfully develop a light and temperature monitoring system using MEMSIC Professional Kit for WSN. This involved the reprogramming of the motes to get them to collect only light or temperature data (humidity temperature) and transfer this data to the base station. Both methods to reprogram motes is through MoteConfig.

Method #1:
The radio module is connected to the programming board then connected via USB to the computer.

Method #2:
Radio frequencies via the Base Station are used to reprogram, or even send basic commands to the motes.

3.2. WSN Implementation

➢ Results from Experiment #2: Light Sensing

✓ COLLECTING LIVE DATA: A sample set of results from our Light sensing application is shown in table. Given the environment, the data collected over the time period was very consistent. The data collected included light. All data were displayed in standard engineering units.

Id	Time	parent	voltage [V]	lightc [lux]
7653	2/29/2012 15:25	0	2.5506	8.51
7652	2/29/2012 15:25	0	2.5249	10.35
7654	2/29/2012 15:25	0	2.5454	24.61
7651	2/29/2012 15:25	0	2.5351	11.27

3.2. WSN Implementation

❑ **Experiment #3:**

➢ The application from experiment #2 was developed further to implement an event-driven system. Using the raw units for the environmental data collected, a threshold criteria was used to force the motes to determine whether or not it should sleep or send the data to the base station.

➢ Sensors were placed in a box to control the amount of light exposure. The behavior of the sensors was observed depending on whether or not the sensors slept or sent the data and the amount of light it received.

3.2. WSN Implementation

➢ Results from Experiment #3: Event-Driven System

✓ COLLECTING LIVE DATA: A sample set of results from our event-driven system application is shown in table. Given the environment the data was collected or observed that the sensor slept since no packets were sent. The data collected included light. All data were displayed in standard engineering units.

Id	Time	parent	voltage [V]	lightc [lux]
8000	4/10/2012 15:25	0	2.5506	743.68
8000	2/29/2012 15:26	0	-	-
8000	2/29/2012 15:27	0	-	-
8000	2/29/2012 15:28	0	2.5505	750.53

Sensors were placed outside of box

3.2. WSN Implementation

❑ **Completed Objectives includes:**

- ➢ Implementing a live wireless sensor network and the specified requirements for it to function successfully.
- ➢ Developed a custom program application to target specific environmental elements specifically as light and temperature.
- ➢ Reconfigured database properties and Base station activity to interact with the MoteView monitoring software and store results of incoming data packets.
- ➢ Implementing the event-based protocol to force motes to send data packets in the event of specific environmental changes.

❑ **Research objectives for the future:**

- ➢ Implementation of a Clustering Algorithm and allow fusing of data from a cluster-head consequently cutting down network traffic.
- ➢ Implementation of Security Measures to identify Malicious Nodes and avoid them in the WSN.
- ➢ Identifying algorithmic patterns to allow motes to automatically adapt to environmental changes and determine drastic events to report back to the base station.