

# Project Report : Object tracking in Wireless Sensor Networks

Pulkit Gambhir , Mohit Rajani

July 25, 2005

## **Abstract**

Wireless Sensor Networks are ad-hoc networks that consist of nodes with limited computational power and limited power resources that are fitted with a radio transceiver, and may be deployed easily in a hostile environment due to their nearly zero configurability. These sensor networks are generally used to do environment monitoring; wherein there is a need to design energy and resource efficient algorithms for data acquisition and to come up with schemes to route data efficiently through the network.

The aim of this project was to set up an ad hoc wireless sensor network and use it to track moving objects that are *bugged* with a radio transmitter. This project addressed issues such as multi-hop message routing and (interrupt based) scheduling , data collection & collation and time synchronisation in wireless sensor networks as well as trying to achieve a mapping from signal strength parameters (*RSSI*) to distance.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Hardware &amp; Software Used</b>	<b>4</b>
<b>3</b>	<b>Design Strategy</b>	<b>5</b>
3.1	The routing phase . . . . .	5
3.1.1	Avoiding collisions . . . . .	5
3.1.2	Updating <i>Parent</i> and <i>Hop Distance</i> . . . . .	6
3.1.3	Time Synchronisation . . . . .	6
3.1.4	Adding and deleting nodes . . . . .	7
3.2	The listening phase . . . . .	7
3.2.1	Avoiding collisions . . . . .	7
<b>4</b>	<b>Experimentation</b>	<b>8</b>
4.1	Signal strength and Distances . . . . .	8
4.1.1	Obtaining empirical results . . . . .	10
4.2	Collision rates . . . . .	10
<b>5</b>	<b>User End Processing</b>	<b>11</b>
<b>6</b>	<b>Extending Further</b>	<b>12</b>

# 1 Introduction

Wireless Sensor Networks today find their main usage in environmental monitoring problems. The aim of this project was to put these networks to use in a slightly different context, that of tracking objects. This *sensing* of moving objects was done in this project by measuring radio signal strengths rather than using sensors such as magnetometers, accelerometers etc. In order to achieve this, we had to find out an empirical relation via experimental techniques between a received signal strength metric (*RSSI*) and distances.

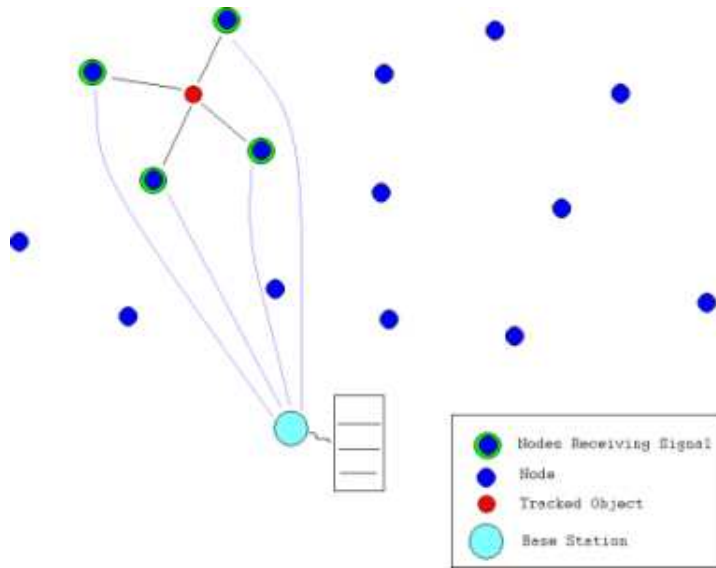


Figure 1: Locating objects in a Wireless Sensor Network

The motivation for doing this project was to allow to set up a nearly zero configurable network easily in a limited geographical environment where only restricted access is to be allowed to vehicles, people etc. A typical example would be tracking moving vehicles inside a wildlife sanctuary, where it is pertinent to ensure that these vehicles only stay restricted to certain paths.

## *Why Wireless Sensor Networks ?*

Wireless Sensor Networks offer distinct advantages over other alternative approaches to solve such an object tracking problem. Some of these are mentioned below.

- Nearly zero configurability
- Low cost of sensor nodes compared to GPRS modems

- Minimal energy consumption
- Easy expansion & reduction of coverage area

## 2 Hardware & Software Used

The network was setup using Berkeley motes from Crossbow. The programming was done on MPR-410 Mica2 motes. These motes have an Atmega128L micro-controller, a CC1000 radio transceiver, and can be integrated with a variety of sensors using a 51-pin connector to an external sensor board. The motes are powered by two AA batteries.

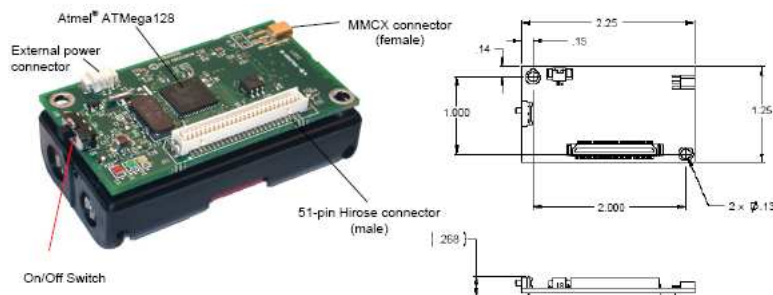


Figure 2: The MPR-410 network mote

These motes can be interfaced with a computer using a programming board that connects via a serial port. This interfacing is necessary for programming as well as for transferring data over UART.

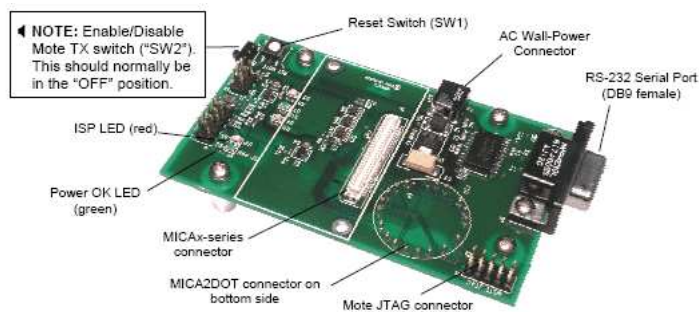


Figure 3: The MIB-510 programming board

## 3 Design Strategy

The sensor network setup consists of a number of nodes that have all been programmed with the same code. These nodes are capable of initially establishing routing in the network and once this is done, they go into a listening mode during which they relay all packets from the transmitter being tracked to the base station. The transmitter is assumed to be hostile and does not transmit any information in the packet, only its own ID. These raw packets are taken and a time stamp and a received signal strength parameter value is augmented to their data pay load by all listening nodes. The packets are then routed through the network. It is also ensured that during the routing period, time synchronisation is achieved in the network. All the data is collated at a base station which then passes all the data onto a processing server over UART.

### 3.1 The routing phase

To begin with, all the sensor nodes are booted up. The last node to be booted is the base station. The base station on booting broadcasts a route finding message onto the network. This message carries the node ID of the base along with a hop count field as well as the base station's time stamp. The hop count is initially set by the base station to 1. All nodes within a single hop distance of the base receive this message and set their *Parent* to be the base station and their *Hop Distance* to be 1. These nodes also reset their clocks to synchronise with the base station. All these nodes then re-broadcast the routing message writing inside their own node ID instead of the base's, incrementing the hop count by 1 and also update the time stamp of the message.

In this manner the route finding messages are flooded through the network and all nodes soon know who their parent node is and also at what hop distance they are residing from the base. All message transmission in the listening period of the node is directed towards the parent node.

#### 3.1.1 Avoiding collisions

When a packet flooding strategy like the one described above is used to determine routing, it is very likely that due to the large number of broadcast packets in circulation, some nodes may not receive the packet in one go due to interference. Hence it becomes essential to re-broadcast packets for a certain period of time to ensure with a probability that every node has received a routing packet atleast once. To ensure this the base station keeps sending out these broadcast messages for a fixed period of time (around 10 seconds) with a random back-off (of the order of 1 second).

Every node will thus receive much more than just a single routing packet and it must make a decision about which of these packets it will further transmit as otherwise there will be excessive packet flooding in the network and functionality may get severely affected. To counter this, a simple heuristic was adopted. A node re-broadcasts only those packets which lead to an updation of its *Parent* and/or *Hop Distance*.

### 3.1.2 Updating *Parent* and *Hop Distance*

Each node receives multiple routing packets during the initial working phase of the network. All of these cannot be allowed to change its *Parent* and/or *Hop Distance*. The simplest strategy to follow is that both the values are changed only if the new *Hop Distance* is less than the old one. This simple check also ensures that atleast one path to the base is available to every node in the network.

### 3.1.3 Time Synchronisation

The routing messages that flood the network also carry time stamps. Whenever a node receives a message and decides to change its *Parent* and *Hop Distance*, it also updates its clock. In such a manner time synchronisation may be achieved in the network. There are however a couple of issues with this strategy.

The first problem is that of the clocks of adjacent nodes in the network being slightly out of phase (by an amount  $\mu$ ). This will happen due to the delay introduced due to message radio transmission, reception and processing. This delay although not the same in every case will roughly be of the same order of magnitude. Therefore in an  $n$ -hop network, the worst case delay would be  $\mu \times n$ . This delay due to phase shifting is almost unavoidable, unless some correction factors are introduced in the processing of received data. The strategy we use to counter this phase difference is that although we increment the node clocks every  $\frac{128}{1024}$  milliseconds; the basic unit of time is 1 second. Thus, some of this phase difference is just ironed over. However, if  $n$  were to become fairly large, this might pose a problem.

The second problem is that of clock skew. This is basically the difference in clock times of two nodes that might occur even though they may have started at the same time value. In case of the hardware used clock skew was of the order of 1 millisecond in every 50,000 milliseconds. Clock skew can be completely taken care of by periodically putting the system into the routing phase; thus re-synchronising clocks.

### 3.1.4 Adding and deleting nodes

The addition and deletion of nodes to the network can be done as long as re-routing is done. Hence to allow for easy addition and deletion of nodes the routing stage needs to be initiated repeatedly.

## 3.2 The listening phase

After the reception of a routing packet at a node, a short timer is started that signals the end of the routing period and the beginning of the listening period. In this period, the aim of each node is to listen for messages sent out by the transmitter being tracked and then try to route these packets all the way to the base station. Before routing, the nodes augment three pieces of information to the received message :-

- Node ID
- Time Stamp
- Received Signal Strength (*RSSI* value)

All the augmented data is essential for the front end processing in order to determine the location of the transmitter. A node ID to coordinate mapping is maintained at the processing server. Hence when the messages reach the server, it knows from where what signal strength is being received. This is the data needed for location estimation. The time stamp is necessary to ensure that when the location estimation is being done, signal strengths measured at the same time time points are being used as otherwise irrelevant results may be obtained.

### 3.2.1 Avoiding collisions

All nodes getting the transmitter's signal will try to forward the message to their parent. To ensure no collisions are occurring firstly random back-offs are used. Secondly, all message forwards await an acknowledgement from the parent, this will ensure that messages reach the parent if not in the first go then in subsequent tries.

## 4 Experimentation

Experiments were carried out with the intention of either arriving at certain empirical results, or at optimal values of certain parameters to enhance network functionality. In this section, we describe some of the important experiments that were carried out.

### 4.1 Signal strength and Distances

It is essential to have knowledge of a relation between received signal strength and distance of the emitting node, in order to make estimates of location. Hence, experiments were carried out in different field locations to establish the nature of this relation. A signal strength measure is provided by the sensor nodes in the form an RSSI (*Reduced Signal Strength Indication*) value. This value has a known linear relationship with signal strength. Our aim is to correlate this signal strength indicator with distance. In order to do so, we conducted experiments in different environmental settings to establish such a relation. The

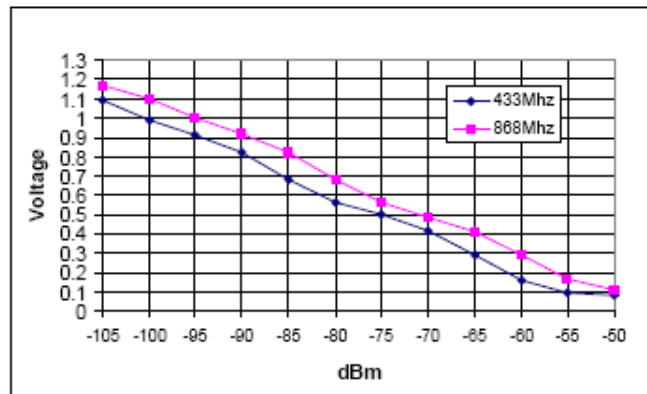


Figure 4: RSSI to Signal Strength relationship as specified in the CC1000 manual

$$RSSI(dBm) = -51.3 \times V_{RSSI} - 49.2$$

experiments were conducted by placing a node programmed to emit a fixed number of messages, at certain measured distances. These messages were then counted at the base station and their RSSI values on reception noted. These data readings were then used to do a quadratic least square curve fitting to determine an empirical relation between distance and RSSI values. The experiment was performed in three environments :-



1. Building Corridors
2. Forest Type Environment
3. Open Ground

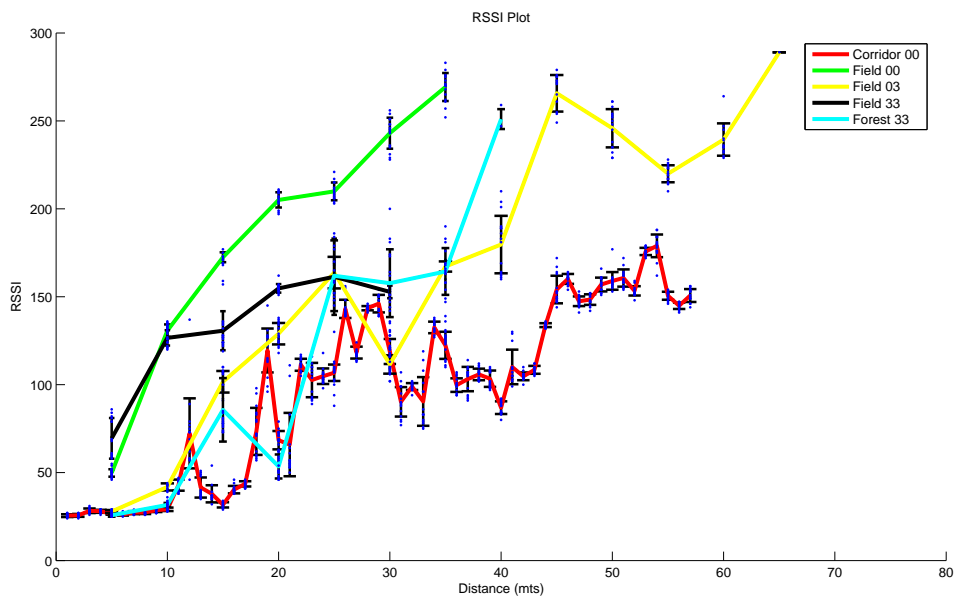


Figure 5: Results of field experiments.  $XXXX_{ij}$  represents experiment carried out in  $XXXX$  with the base station at height  $i$  mts from the ground and the transmitter at height  $j$  mts from the ground.

The major conclusions to be drawn from the experiments were :-

- Each environment offered slightly different relations for the RSSI to distance conversion, but the data obtained showed a quadratic relation by in large.
- Often small perturbations in the environment such as people walking through could render the readings irrelevant.
- In indoor conditions, there was no smooth variation of RSSI with distance.
- Other parameters such as bit error rate could be used more reliably than RSSI.

### 4.1.1 Obtaining empirical results

The experimental data is obtained as a set of RSSI values for each distance measurement. This data showed a roughly quadratic variation. To get hold of an empirical conversion formula; a least square quadratic curve fitting was done using MATLAB.

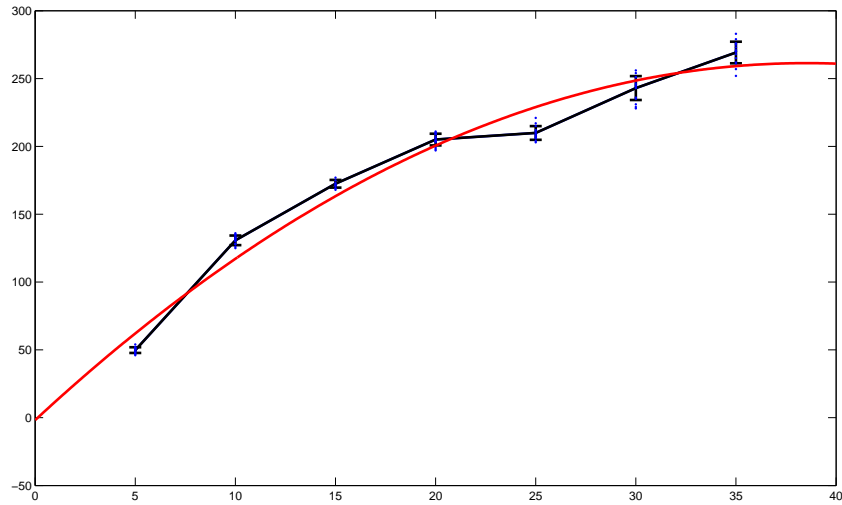


Figure 6: Quadratic curve fitting for reading of field00.  
Equation obtained  $-0.1772x^2 + 13.6629x - 1.7547$

## 4.2 Collision rates

The entire design strategy as far as route determination as well as package routing is concerned uses a lot of *non-intelligent* random back-offs to prevent collisions of data packets. It was therefore necessary to establish the order of magnitude of these random back-offs in order to ensure that firstly time spent in communication is minimised and secondly that packet loss rate or collision rate is minimised.

In order to determine these values, a small single hop network was setup using one base station, three listening nodes and a single transmitter. The nodes simply relayed all messages to the base exactly once without waiting for acknowledgements. The main conclusions drawn were :-

- Simultaneous transmission from three nodes lead to almost 100% packet loss
- With differences of the order of 200+ ms there was nearly 0% packet loss
- With decrease in time interval between transmissions, there was increase in packet loss, but this increase was not very dramatic.
- Even with small intervals of about 20ms, packet loss was below 10%

## 5 User End Processing

At the user end, the server keeps receiving the signal strength values from base station. These readings are recovered from the UART packet and correlated to get distance estimates. These are then used along with a database that lists the locations of all the sensor nodes to get an estimate of the location of the object being tracked.

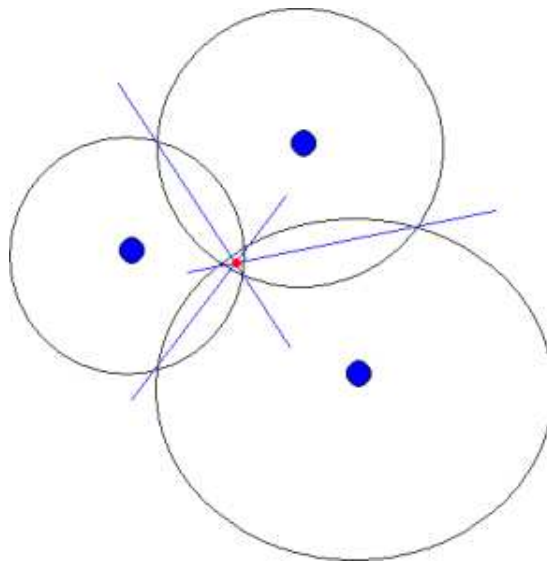


Figure 7: Location estimation technique

Presently we use a simplistic scheme to determine the location of the object. This is well described by the figure above. We only consider the top three strongest signal values and use them to estimate the location. This sorting of readings and figuring out of top three strongest signal measurements is done at the base station itself and is transparent to the server. This simplistic approach is however not a good idea since it gives no result in a lot of cases due to the nature of solution that is being sought and the unreliable relation between RSSI and distance.

The idea is to have a Wireless Sensor Network up and running, this network being in constant communication with a computer on the network. The User End processing is then fed to a Web Based Interface from where the position of the transmitter can be tracked on screen.

## 6 Extending Further

This project has a lot of scope for further improvement. A major amount of time was spent in familiarisation with the software tools and programming languages used, and hence complicated protocols for communication could not be implemented. Some further areas of improvement could be :-

- Implementing a better multi-hop communication scheme that minimises collision
- A store and forward approach to collate data
- Decision making about when and which nodes to put to sleep
- Better algorithms for location estimation
- Use of more reliable parameters for distance measurement than *RSSI*
- Incorporating GPS to determine position of nodes, instead of manually feeding coordinates