Three Beacon Sensor Network localization through Self Propogation

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Master of Technology

by
Mohit Choudhary

to the

Department of Computer Science & Engineering
Indian Institute of Technology, Kanpur

May, 2005

Certificate

This is to certify that the work contained in the thesis entitled " *Three Beacon Sensor Network localization through Self Propogation* ", by *Mohit Choudhary*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

May, 2005

(Dr. Bhaskaran Raman)
Department of Computer Science & Engineering,
Indian Institute of Technology,
Kanpur.

Abstract

This thesis describes a distributed algorithm for estimating node positions in a sensor network. The approach makes use of three stationary beacons (with knowledge of their position apriori) to localize a few neighboring nodes in the network. These freshly localized nodes in turn propagate the location information to their neighbors and the process continues until the entire network is localized. The sensor nodes make use of the geometric constraints induced by their neighbors both in terms of radio connectivity and its negativity to decrease the uncertainty of their position and thus achieve localization. Fixed beacon placement approaches deploying a large number of beacons often result in being inadequate and non realizable practically considering the environments in which sensor networks may be expected to operate (with high terrain uncertainties).

In this thesis, we therefore motivate the need for self propagation of localization information in a sensor network without having to strategically deploy a large number of beacons. The proposed scheme provides an extremely low cost (in money, and power consumption) solution to the localization problem. We design, evaluate and analyze this novel self localization algorithm using simulations. Also, in simulation, we demonstrate that the algorithm scales to large networks and handles real-world deployment scenarios in presence of noise and gray areas. Further, in order to study the performance of the model in more realistic scenarios we have developed an in-house platform of sensor nodes using *Atmel AT89C52 microcontroller and radiometrix TX/RX*. We implemented the algorithm on a physical sensor network of such nodes and empirically assessed its accuracy and performance. Our on-ground implementation of the scheme and our results show the effectiveness of our localization mechanism.

Acknowledgements

A journey is easier when you travel together. Interdependence is certainly more valuable than independence. During the course of this thesis for over a year, I have been accompanied and supported by many people. It is indeed a pleasure that today I have the opportunity to express my gratitude for all of them.

I take this opportunity to express my sincere gratitude toward my supervisor Dr. Bhaskaran Raman for his invaluable guidance. His contagiously enthusiastic and integral view on research and his mission for providing "only high-quality work and not less", has made a deep impression on me. It would indeed have never been possible for me to take this work to completion without his relentless support and encouragement. I consider myself extremely fortunate to have had a chance to work under his supervision. Inspite of his hectic schedule he was always approachable and took his time off to attend to my problems and give the appropriate advice. Learning how to do research and how to write technical reports from him has being a life time experiences in itself. It has been a very enlightening and enjoyable experience to work under him.

My thanks are also due to Madhav Pandya , final year MTech student for his invaluable support in the experimental setup. My sincere thanks are due to Prashant Sharma and GopalRajpurohit, also final year MTech students, who have been of great help at various stages of this work. Inspite of their busy schedules all three of them took time off to help me.

I also wish to thank whole heartily all the faculty members of the Department of Computer Science and Engineering for offering us some excellent courses in the field of Computer Science and Engineering. I also extend my thanks to the technical staff of the department for maintaining an excellent working facility.

Lastly, I am grateful to my wife Frezy for the inspiration and moral support she provided throughout my stay at IIT. Without her loving support and understanding I would never have completed my present work.

Contents

List of Figures

Chapter 1

Introduction

A sensor network is a collection of communicating sensing devices, or nodes. All of the nodes are not necessarily communicating at any particular time, and nodes can only communicate with a few nearby nodes. The aim of the network is to get data back to the *base station* in an energy-efficient manner. Battery-powered and wireless sensor devices are small, expendable, and inexpensive. Practical applications for networks of these sensors have a varied range in the following areas and more: from environmental/ habitat monitoring [22] to bird and animal tracking [23] to structure monitoring [25] to surveillance to corrosion detection [24] etc. Future wireless sensor networks are envisioned to consist of hundreds to thousands of sensor nodes communicating over a wireless channel, performing distributed sensing and collaborative data processing tasks for a variety of vital applications. Such ubiquitous sensor networks will improve the safety of our buildings and highways, enhance the viability of wildlife habitats, dramatically shorten disaster response times, reduce commute times, and contribute in many other vital functions. We can imagine ad hoc sensor networks deployed for various kinds of applications, providing continuous and spatially dense observation of biological, environmental and artificial systems [1]. Such networks will be unattended, and must self configure to adapt to the particulars of their environmental setting and the availability of other nodes within the system. Localization in sensor networks has been the subject of a burgeoning amount of research in the past few years. Although a large number of techniques exist in sensor network localization, but much of the research thus far has focused on implementing a complex technology for localization, and beacon placement issues. Intuitively, approaches such as uniform and very dense beacon placement or ones utilizing mobile beacon should suffice to improve the quality of localization but the fundamental limitation of these approaches is that they do not take into account the terrain conditions. It is virtually impossible to predict apriori whether the terrain conditions would physically allow beacon mobility or accurate positioning of such large number of beacons.

Moreover, the sensor nodes as envisioned are supposed to be low cost void of any technology other than the bare minimum requirement of an onboard sensor and a radio. Clearly then, the ideal approach to localization should limit the beacon nodes to a very small number and further treat localization as a distributed problem void of any additional technical requirements that the sensor nodes inherently possess. We therefore aim to show that such a simple solution is indeed possible to solve the said problem.

1.1 Problem Statement

Spatial localization i.e., determining physical location of a sensor node is a critical service in these wireless sensor networks [2]. Both at the application layer as well as for the underlying routing infrastructure, it is often useful to know the locations of the constituent nodes with reasonable accuracy. An example of the inputs and outputs to the localization problem is shown in Figure 1.1. The work in this thesis focuses on 2D networks as shown, but the proposed scheme can be extended to 3D networks also.

Localization is an essential tool for the development of low-cost sensor networks for use in location-aware applications and ubiquitous networking [21]. For example, the Global Positioning System (GPS)[3] is a reliable technology for discovering the location of portable devices; however, it is expensive and does not work indoors because a view of the sky is obstructed. Localization of sensor networks provides similar

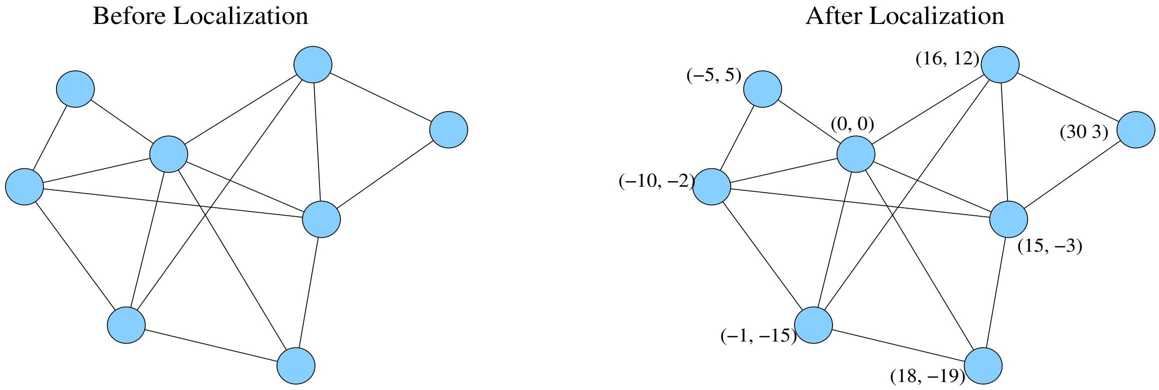

Figure 1.1: Localization

functionality to GPS, but it works indoors as well. A full-range of location-aware applications such as autonomous navigation, environmental monitoring, and "smart buildings" are realizable given a robust algorithm for sensor network localization.

This thesis describes a distributed algorithm for a sensor network localization in which nodes have the ability to assimilate and utilize the connectivity and negativity information provided by their nearby nodes. Distributed computation and robustness in the presence noise and gray areas are key ingredients for a practical localization algorithm to provide reliable results over a large scale network. We formulate the problem in terms of two-dimensional graph realization problem as follows: given a planar graph with edges defining connectivity, recover the Euclidean position of each vertex (within some error bound, defined in chapter 4) given the absolute position of only three reference beacon [1] nodes in the graph. This is a difficult problem for several reasons. First, there is often insufficient data to compute a unique position assignment for all nodes. Second, information itself may be erroneous, compounding the effects of insufficient data and creating additional uncertainty. Finally, it is difficult to devise algorithms that scale with the size of the network, considering the limited communications capacity of a wireless channel.

[1]We use the term *Beacon node* to refer to a node that has prior knowledge of its absolute position, either by manual initialization or an outside reference such as GPS.

Our approach therefore is a distributed localization technique requiring just three beacon nodes with known positions. In our scheme each node makes use of the negativity constraints in addition to the connectivity constraints that it receives from its one-hop neighbors, terms we explain formally in Chapter 2. The scheme in fact makes use of very minimal message exchange among one-hop neighbor nodes and requires simple and very little computation at each node. The scheme proposed here is considered more suitable for sensor networks where the gray area is small (such as in outdoor deployments [17]). We study the effect of gray areas on our algorithm in Chapter 5 and find that the mechanism is useful even when the gray area is as high as $1/3^{rd}$ of the transmission range.

1.2 State of the Art

Existing geolocation systems such as GPS do not always meet the operational (e.g. low power), environmental (e.g. indoors) or cost constraints of sensor network localization. Thus to get around the problem of providing localization in sensor networks the research community has concentrated on one of the follwing techniques:

- Use of multiple static reference beacons.
    1. Use of signal-strength estimation.
    2. Use of acoustic ranging.
    3. Use of connectivity information.
- Use of Mobile Beacon

Such techniques however also have various drawbacks. To start with, most of them are still very expensive. They require technology additional to the requirement of the application. They have a requirement of complex setup procedures. Further, the technique of using mobile beacon suffers an additional drawback of not taking into account the terrain uncertainties. It is virtually impossible to predict apriori whether the terrain conditions would physically allow such mobility of beacons.

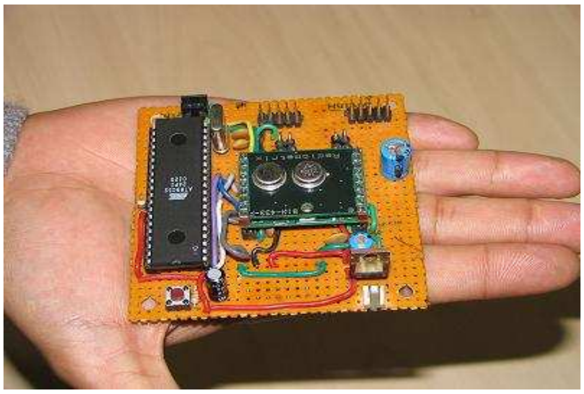

Figure 1.2: Sensor Node

1.3 Summary of Contributions

The thesis makes the following contributions:

- We develop a novel distributed algorithm in which nodes in the network apply the geometric constraints induced by information of their neighbours regarding radio connectivity and its negativity , to determine their position with a small error. The radio negativity is an important bit of information that no previous research work has explored the possibility of. We further make use only three beacon nodes that have the knowledge of their position apriori.

- We evaluate the scheme using simulations for a wide range of scenarios to establish a relation between the node density and transmission range per node essentially required to achieve effective localization with the proposed scheme. We also test the effectiveness of the scheme in presence of gray areas and also as the region of operation scales up.

- We undertake real life experiments with in-house developed sensor nodes (based on*Atmel AT89C52 microcontroller and radiometrix TX/RX*) as shown in 1.2 to establish the simulation results and the viability of the proposed scheme.

1.4 Organization of our work

In Chapter 2 we discuss the basic concepts used in the suggested localization approach. In Chapter 3 we mention related work. We give a brief description of an existing method, designed by Bulusu *et al.*[16]. They attempt to achieve sensor network localization using the connectivity constraint and a large number of beacons. We then discuss the drawback of their method. Chapter 4 describes the distributed localization algorithm proposed by us. Chapter 5 describes the simulations setup, the results obtained and the analysis. Chapter 6 give an overview of the real-life experiment setup and results obtained, and finally the conclusion.

Chapter 2

Related Work

In this chapter, we discuss the related work and describe Bulusu *et al*'s [16] work briefly. Localization in sensor networks has been the subject of a burgeoning amount of research in the past few years. Various localization approaches have been proposed and they fall into two broad categories: proximity based approaches [5][6] and multilateration based approaches [7][8].

Specifically, the recent efforts to develop localization procedures for wireless sensor networks such as [2] [16] make use of multiple static reference beacons, [12] uses a few mobile nodes equipped with GPS receiver, [9] uses signal-strength estimation, and [13] makes use of acoustic ranging. The common characteristics in these efforts are utilization of technology that is additional to the requirement of the application task. This results in a multifold increase in the cost of deployment of such sensor networks than the application itself requires. Also, such scenarios suffer the additional disadvantage of performing complicated setup procedures at the user level.

To summarize, much of the research thus far has focused on implementing a complex technology for localization, and beacon placement issues. Intuitively, approaches such as uniform and very dense beacon placement or ones utilizing mobile beacon should suffice to improve the quality of localization but the fundamental limitation of these approaches is that they do not take into account the terrain conditions. It is virtually impossible to predict apriori whether the terrain conditions would physically allow beacon mobility or accurate positioning of such large number of beacons. Clearly therefore, the ideal approach should limit the beacon nodes to a very small number and further treat localization as a distributed problem void of any additional technical requirements.

Doherty et al. [11] developed a centralized technique using convex optimization to estimate positions based only on connectivity constraints given some nodes with known positions. This approach requires central computation and would prove to be infeasible for ad-hoc applications because of the high communication costs, inherent delay and lack of central infrastructure. Other related work that is based on propagation of location information from known reference nodes is as follows: Bulusu et al. and Simic et al. propose distributed algorithms for localization of low power devices based on connectivity [2]. Savarese et al. use a two-phase approach using connectivity for initial position estimates and trilateration for position refinement [19]. Patwari et al. use one-hop multilateration from reference nodes in a physical experiment using both received signal strength (RSS) and time of arrival (ToA) [20].

In this thesis we make two major departures from all the previous research described above. First, we focus on a distributed localization technique requiring just three beacon nodes. Second, more importantly each node trying to localize itself makes use of the negativity constraints in addition to the connectivity constraints that it receives from its one-hop neighbors. This is an important bit of information that no previous algorithm has explored the possibility of. Now we go on to describe the Max approach proposed by Bulusu *et al*'s [16] that attempts at minimising the number of beacons to achieve localization making use of connectivity constraint information of the beacons.

The approach used by Bulusu *et al*'s [16] involves three main steps. In the first step, the region is devided in $step \times step$ squares. In the second step, they calculate localization error at all the points that correspond to the corners of each square. The localization error is calculated using the connectivity constraints available due to the beacons already in place. In the third and final step, they add new beacon at a point that has the highest localization error among all points. The algorithm

Step1 Divide the terrain into **step** $\times$ **step** squares.

Step2 Measure localization error at each point $(i \times step,\ j \times step)$ in the terrain that corresponds to a square corner. $(0 \leq i\ ,\ j \leq \frac{side}{step})$
Number of data points in terrain P_T - $(\ \frac{side}{step} + 1)^2$.

Step3 Add new beacon at the point (X_a, Y_a) that has the highest measured localization error among all points.

Figure 2.1: An overview of Bulusu *et al*'s Max approach [16]

is shown in Figure 2.1.

Even though this technique has been successful in minimising the number of beacons required for localization in a region, it is still far from being the optimal solution. To start with, the process requires on ground analysis of localization error at each iteration. This is a luxury that no sensor network deployment has the pleasure of. Secondly, the process is far too time consuming than desired.

In order to prove the effectiveness of their scheme they carried out various simulation. The simulation setup was designed for performing simulations in a square terrain of side 100m. Each node had a nominal radio range of 15m. To study the performance of their algorithms as a function of beacon density, they generated a variety of beacon fields of different densities. In each of their run, they varied the number of beacon nodes from 20 to 240 in increments of 10 beacon nodes. The corresponding beacon density varies from 0.002 beacons per square m to 0.024 beacons per square m.The results obtained in ideal conditions are shown in Figure 2.2. We make a comparison with these results in Chapter 5 to show how our proposed algorithm performs better under similar conditions.

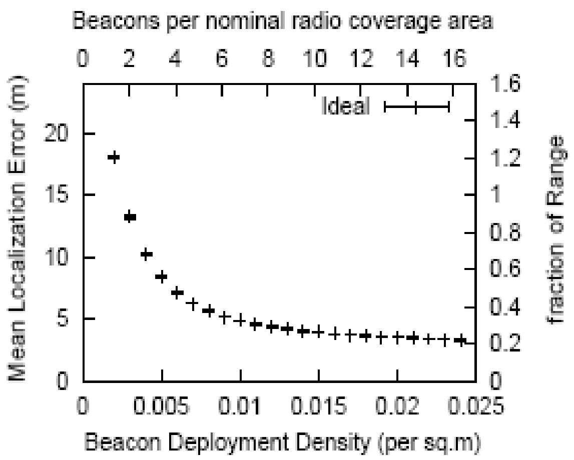

Figure 2.2: Mean localization error vs. beacon density (Ideal) [16]

Chapter 3

Preliminaries

This chapter explains the basic concepts used in our algorithm.To start with we define nodes to be *known* or *unknown* depending on whether or not they have localized themselves. *Unknown* nodes become *known* nodes by applying the connectivity and negativity constraints received from their neighbours. We now discuss these constraints in detail and how our proposed scheme implements them in the following sections.

3.1 Connectivity based RF Localization

If a node can communicate with another node, its position is restricted by the connectivity constraints to be in some region relative to the other nodes. That is if a node can communicate with another, a proximity constraint (geometric constraint due to radio connectivity) exists between them. As an example, if a particular RF system can transmit 10m and two nodes are in communication, their separation must be less than 10m. More than one such connectivity or proximity constraints define the set of feasible node positions in a network.

Every time a node senses another nodes transmission, it generates a new quadratic constraint that it uses to further reduce the uncertainty in its position. This is illustrated in Figure 3.1 below. After the first observation of node at (x_a, y_a) (Figure

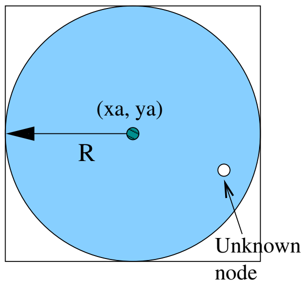

Figure 3.1: One connectiviy constraint

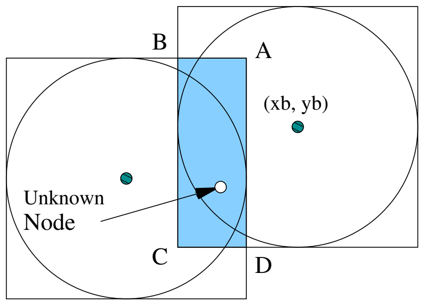

Figure 3.2: Two connectivity constraints

3.1), an unknown node denoted by a white dot limits its position to the circle of radius R around the Node at (x_a, y_a). The second observation by another node at (x_b, y_b)(Figure 3.2), further constrains possible node positions, reducing the uncertainty of its position (blue region). By approximating the sensing region by a rectangular bounding box, we can in effect replace the quadratic constraint by a weaker but simpler linear constraint, seen in (Figure 3.2). Although this approximation overestimates the uncertainty of the node's position, it guarantees that the actual position of the node is always within the bounding box and considerably simplifies the computation of the bounding box. More precisely, let Q be the bounding box for a node. Then, the unknown node will update its bounding box using the constraint imposed by the node at position (x_b, y_b) according to the following rule:

$$Q \rightarrow Q \bigcap [x_b - R, x_b + R] \times [y_b - R, y_b + R] \tag{1}$$

In fact, when the unknown node hears from nodes placed at (xa, ya) and (xb, yb) it computes the end points two square boxes centered at (xa, ya) and (xb, yb) respectively using the above rule. Further it performs intersection check on these points to generate and store the co-ordinates of points {[A, B, C, D]} (as in Figure 3.2) that define its bounding box.

3.2 Negativity based RF Localization

We can use radio unreachability information too for localization, as follows. In Figure 3.3 , node M is not in the range of another node N1 placed at (xc, yc) but is informed by its neighbor N that N can communicate with N1. The position of the node M is then restricted by the negativity constraints to be in some region relative to the other two nodes. The node M can now use this constraint information provided by its neighbor N to reduce its bounding box Q as shown by the gray region in Figure 3.3.

Let's assume that points A, B, C and D (in Figure 3.3) define the bounding box of node M as obtained after applying the connectivity constraint. Once the node M receives the negativity constraint in form of (xc, yc), it computes the co-ordinates of the points E, F, G and H using the same rule. Further, on performing intersection check and finding the co-ordinates of points J and K the node updates its bounding box as {[A,B,C,G](K,J)}. That is, of the four original end-points A, B, C, D, point D has been replaced with point G, and two additional points K and J are part of the boundary definition. If more negative constraints are received, one or more of these four end-points get replaced further. Thus, in practice, we always represent the bounding box as a set of four end-points that keep getting updated, if required, on reception of more constraints and a set of additional derived points.

In the proposed algorithm we assume that nodes in the network have uniform region of connectivity that spreads as a circular disc of fixed radius centered at the node. This assumption however does not always hold good in reality due to the presence of gray areas that introduce irregularity in the assumed radio model. Futher, there is also a possibility of errors cropping up due to false negative constraint being assumed by the node. The false negative constraints may arise due to non-receipt of a valid message (in turn due to packet errors). We do understand that such assumptions would lead to increased error of localization in the network and therefore we evaluate the effect of such errors due to gray areas and packet loss using simulations, in Chapter 5.

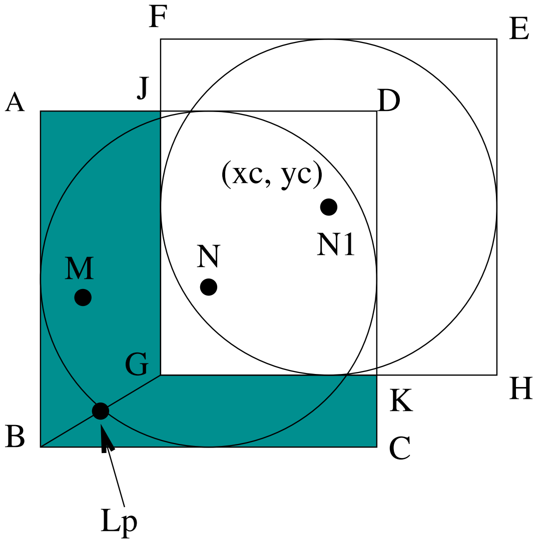

Figure 3.3: Effect of Negativity Constraint

3.3 Localization point and Localization Error

After the application of various connectvity and negativity constraints, a node ends up with a particular bounding box. We then compute a point within this bounding box as the localization point, as follows. Let {[B1, B2, B3, B4] (derived points)} define the finally calculated bounding box for a node in general. For example, in Figure 3.3 the point B1 is A, B2 is B, B3 is C and B4 is G. The node now calculates its Localization point (L_P) as follows:

1. *if (segments B1B3 and B2B4 intersect)* L_P = *point of intersection*

2. *else if (center points of both B1B3 and B2B4 lie inside the polygon)* L_P = *center point of B1B3 or B2B4 that lies closer to the centroid of polygon represented by {[B1, B2, B3, B4](derived points)}* .

3. *else* L_P = *center point of B1B3 or B2B4 that lies inside the polygon represented by {[B1, B2, B3, B4](derived points)}* .

The localization point L_P for Figure 3.3 is as shown. The procedure described above provides us with a point L_P that lies inside the region represented by {[B1, B2, B3, B4](derived points)}and is intuitively around the centroid of the polygon. We do not take the centroid itself to be the L_P considering the fact that for a concave polygon the centroid may not always lie inside it. The procedure defined here is by no means the optimal choice of a localization point but is in fact a fair approximation which is less intensive in terms of computation considering the computation capability of a sensor node. The localization error, ϵ_A can now be defined as the length of the line segment ML_P.

Chapter 4

Our Approach

In this chapter we describe an approach for localization of a sensor network in two-dimensional space. We make use of the connectivity and negativity constraints as described in previous Chapter to achieve the desired localization in a given network of sensor nodes. Consider a sensor network of N nodes, each equipped with an RF transceiver. We define a node's neighbors to be those nodes that have direct communication capability to it. We assume that the nodes are deployed with a randomly uniform density in a square area P=[0,L]x[0,L] called the region of operation. Each node has a unique Node-ID (NID). The basic algorithm assumes that the transmission range of each node is uniform and that the messages are communicated in the network reliably through retransmissions. We later quantify the retransmissions required and the effect of non-uniform transmission range through the simulations in Chapter 5.

Our algorithm starts with three beacon nodes B1, B2, and B3, which know their location in P. In practice, this can be achieved by hand-placing these three nodes at fixed positions. The algorithm proceeds in three main stages, as below:

- **Step-1**: First, the beacon nodes transmit their location. We assume that there is at least one other node which can hear all three beacons. This again can be guaranteed during the initial placement of the three beacons. A node which hears all three beacon nodes is termed as a pseudo-beacon node.

- **Step-2**: In the second stage, the pseudo-beacon localizes itself, and transmits its location to its neighbours. This transmission includes the negativity constraint of all the three beacon nodes. For this reason, this transmission is very expressive. Nodes which hear a transmission from a pseudo-beacon thus immediately localize themselves.

- **Step-3**: In the third stage, the algorithm proceeds in a distributed fashion. Nodes which have localized themselves transmit their location to their neighbours, along with any negativity constraints as appropriate. This in turn helps in the localization of further nodes.

This algorithm is run at the beginning, when the sensor network is deployed (or perhaps periodically if the nodes are expected to be mobile). Note that it is not necessary that the three beacon nodes transmit at or around the same time. They can start their transmission independent of one another. The algorithm will not proceed beyond step-1 until all three beacons have transmitted, and a node has determined itself to be a pseudo-beacon. Each node which has localized itself to a position (probably with some error) transmits a message announcing its location. At any stage, a message transmitted by a node is a 5-tuple (NID, a, b, MID, NCS). NID is the Node ID, and (a, b) are the x and y co-ordinates of its position. MID is the Message ID; MID=1 is used by beacon nodes, MID=2 is used by pseudo-beacon nodes, and MID=3 is used by other nodes. Finally, NCS is the negativity constraint set – a set of negativity constraints, each consisting of a 3-tuple (NID, a, b). Each 3-tuple describes a node which was heard by this particular node. For instance, the node N in Figure3.3, in sending a message to M, would include the 3-tuple (N1, xc, yc) in its NCS. The beacon nodes transmit with a null NCS. For reliability, the message from a node may be retransmitted a fixed number of times. After this, the node neither listens to nor transmits any messages (so far as the localization algorithm is concerned). The algorithm is given as pseudo-code in Figure4.1.

The crux of the algorithm is in the is_localized() procedure, where a node decides whether it has received enough information to localize itself. We now explain the algorithm with reference to the pseudo-code, as well an example, shown through Figures 4.2 to 4.4 . In the first stage, the beacons transmit their location, while the

```
while(! is_localized()) {
// listen for "appropriate" set of messages
}
transmit(NID, a, b, MID, NCS);
// possibly retransmit for more reliability
boolean is_localized() {
        if (type=beacon) { return true; }              // CASE C1
        if (heard 3 messages with MID=1){              // CASE C2
              type = pseudo-beacon;
              // localize with information received
                    return true;
        }

        if (heard 1 messages with MID=2){              // CASE C3
              // possibly more messages with MID=1 were also recd.
              // localize with information received
                    return true;
        }

        if (two valid sets of messages recd){          // CASE C4
              // localize with information received
                    return true;
        }

} // End is_localized()
```

Figure 4.1: Pseudo Code for the Algorithm

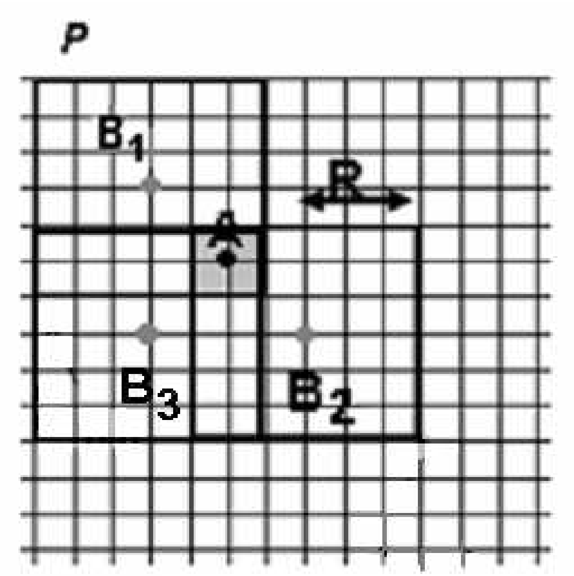

Figure 4.2: Step One- Example run

other nodes are all waiting to hear from other localized nodes. In Figure4.2, node A hears B1, B2, and B3. Node A now falls under case-C2 of is_localized(), and determines its position. It marks itself to be a pseudo-beacon.

In the next step, the pseudo-beacon transmits its message, and two kinds of nodes localize themselves. First are nodes such as Y in Figure4.3, which hear only one message - from the pseudo-beacon, with MID=2. The message transmitted by a pseudo beacon is very powerful - it contains one connectivity constraint and the three negativity constraints (corresponding to B1, B2, and B3). This message is therefore sufficient to localize nodes such as Y. This falls under case-C3 in is_localized().

It is relevant to mention here one more condition which must be met in the original placement of the beacons. Suppose we had used only two beacon nodes B1 and B2 in our approach, node Y would have been able to localize itself, albeit with the well-known flip-ambiguity (in Figure4.3, Y and Y1 would have been indistinguishable). However, this is avoided by the use of the third beacon B3, and its appropriate placement with respect to B1 and B2. B3 can be placed such that it is heard by all

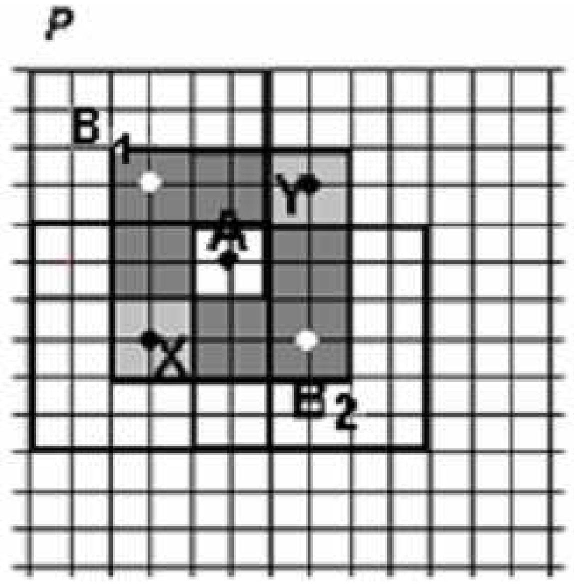

Figure 4.3: Step Two- Example run

nodes in the region X, as suggested in the figure. In fact, this is the only role of the third beacon B3 in our algorithm. Another class of nodes which localize themselves under case-C3 of is_localized() are the nodes in region X and those in region E such as Xi (i=1..4), Y2. These are shown respectively in Figure4.3 and Figure4.4. These nodes hear from at least one of the beacons, and the pseudo-beacon as well. Note that the nodes Xi (i=1...4) all localize to the same point (x2, y2) (each with a small localization error).

All the nodes that have been localized in case-C3 further propagate their own connectivity and negativity constraint. The region of reachability of these messages is shown by the different light grey squares in Figure4.4 (only shown for Xi (i=1...4)). These nodes transmit packets containing the 5-Tuple with MID=3. Now, for the further steps in the algorithm, we define of a valid set of messages. A set of threshold number of messages of MID=3, with the same x and y values, but different NID's constitutes a valid set. These different messages would be sent by different non-beacon nodes which localize themselves to the same point. Also, a single message

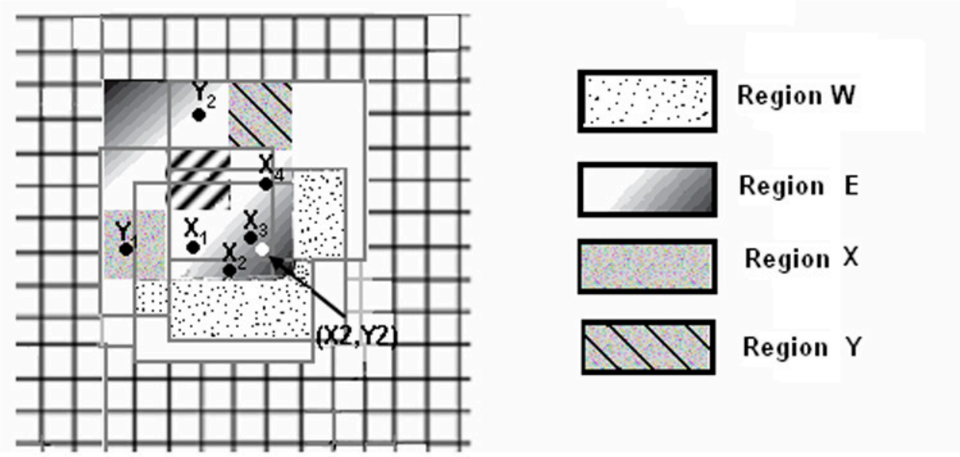

Figure 4.4: Step Three- Example run

with MID=1 constitutes a valid set.

Note that with a value of threshold=1, the case-C4 reduces to hearing any two messages with different (x,y) coordinates. This is equivalent to not introducing a "threshold" parameter. The motivation for introducing the "threshold" parameter is as follows. We have observed in our simulations 5.1 that with threshold=1, the localization progresses "too rapidly" and the localization error is higher overall. For instance, in Figure 4.4, all of the nodes Xi (i=1..4) that fall in region E localize to the same position (x2,y2). With threshold=1, any node hearing one message from any Xi, and say a message with MID=1, would localize itself immediately. The use of a higher threshold restricts the set of nodes which localize themselves in the next step. Of course, a very high value of threshold implies that the progress of the algorithm may be affected drastically. In our simulations, we have found that the use of threshold=2 usually works well (region W in Figure4.4).

Also, another minor optimization which has been found to be useful (but not mentioned in the algorithm above) is the following. A node which does not receive two valid sets of messages, but only one valid set, temporarily localizes itself with just this information and sends a message. On receiving the second set, it further refines its position, and sends a second message announcing this information. The effect of this optimization is that nodes which do not get a second valid set of messages, end up localizing themselves anyway, and help in the propagation of the algorithm. Without this feature, we have found in our simulations that the algorithm often proceeds only to a smaller fraction of the nodes in the region. The above algorithm proceeds, with more nodes localizing themselves at each stage. There are of course, several sources of error possible: (a) there is error inherent in the connectivity-based and negativity-based constraints, (b) a negative constraint may be false, due to non-receipt of a valid message (in turn due to random errors), (c) the presence of gray areas and non-uniform radio connectivity can cause further errors in calculations, and (d) finally, some nodes which are "far away" from other nodes may not receive enough information to localize themselves. We evaluate all of these extensively using simulations, and subsequently also use a real-life implementation.

Chapter 5

Simulation Setup and Results

We have performed detailed simulations to evaluate our algorithm above. In this Chapter, we present the evaluation and the detailed results in the following sections. Section 5.1 provides a brief overview of the simulation setup used and the parameters defined therein. Section 5.2 outlines the results obtained for the various simulations carried out. Section 5.3 provides the analysis of the results obtained and addresses the issues such as presence of gray areas and effect of unreliability in message transmission. The section also analyzes how the proposed scheme scales with increase in area of operation. Finally, we conclude the chapter by comparing the proposed scheme to [16].

5.1 Simulation Setup

We have carried out simulations in Visual Sense platform, which is a simulation framework for wireless and sensor networks that builds on and leverages Ptolemy II [14]. The simulations set up consisted of a set of fifty non beacon nodes and three beacon nodes. Two sets of simulations were carried out, one with TX radius of 30mts/ node and the other with 50mts. In both the sets the dimension of the region of operation were varied to analyze its effect on the proposed scheme. We also studied the effect of varying the value of threshold in each region of operation. We carried out simulations for a threshold value of 1,2 or 3 for each region of operation. The

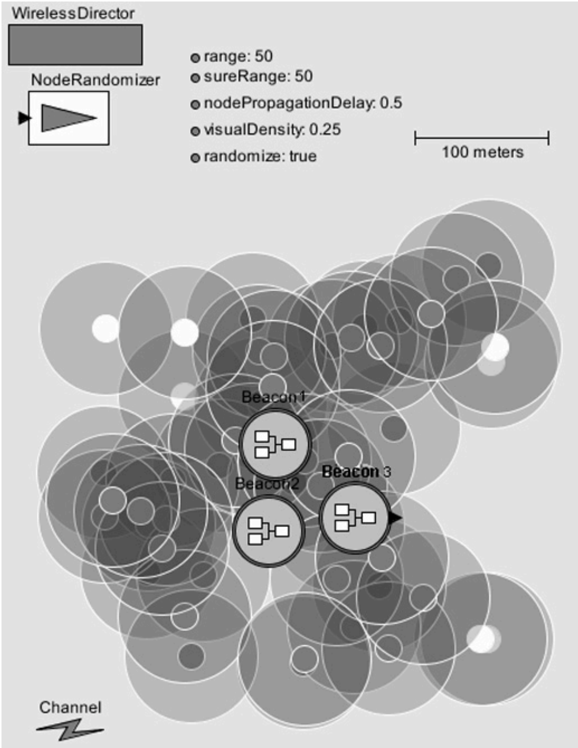

Figure 5.1: Simulation Setup

nodes within a localization error of 20% of their transmission range are considered to be localized as such an upper bound on error is considered to be acceptable to majority of sensor network applications [18]. Figure 5.1 is a depiction of one such simulation run with a node transmission radius of 50 mts (as depicted by the sure range parameter). In case the value of range parameter and sure range parameter are different then the node transmission range varies randomly between the two. The node propogation delay parameter allows a delay of 0.5 seconds to a node before transmitting the required information. The randiomize: true parameter enables random distribution of sensor nodes in the region of operation specified by the node randomizer. A visual density of 0.25 has been used to maintain the clarity of image during the simulation run. Also, in Figure 5.1 nodes within 10% of localization error during the simulation are shown as dark grey and those within 20% of error are shown grey. The nodes that do not get localized are shown white.

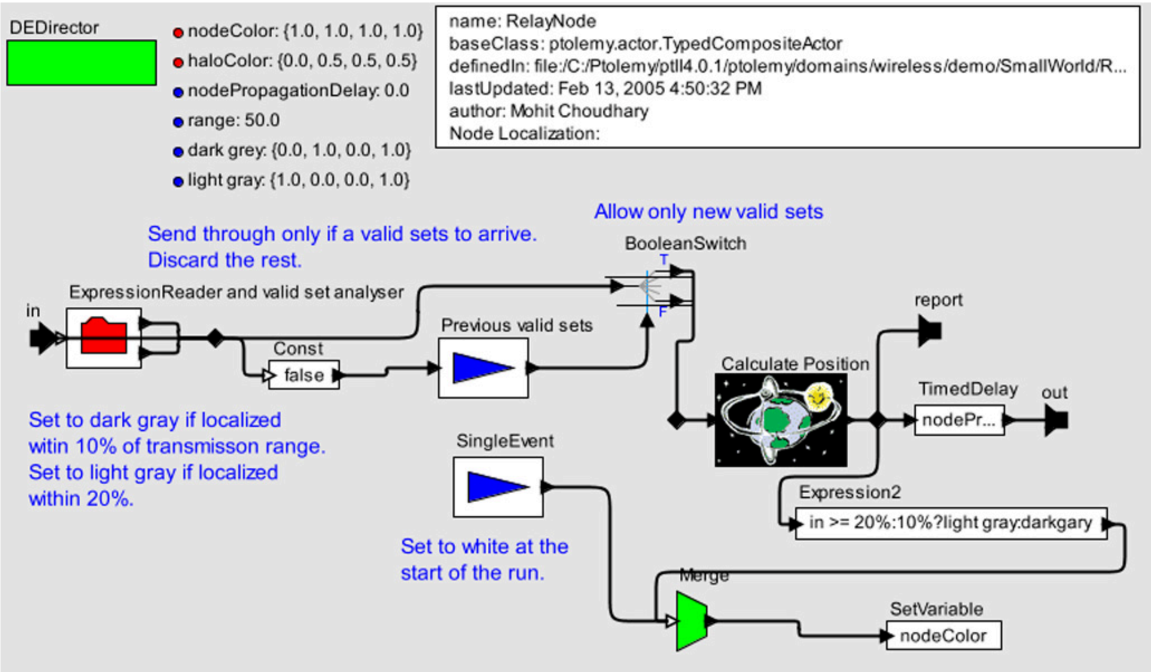

Figure 5.2: Look Inside a Sensor Node

Finally, if we were to take a look inside any of the sensor nodes in the simulation setup. We see the picture as shown in Figure 5.2. the node uses an expression read to analyze the input signal and form a valid set. The *boolean switch* allows only new a valid set to pass through. The *calculate position* module finally calculates the position of the node based on the valid sets obtained. The output of this module is used for the following :

1. To turn the node dark gray or light gray depending on its localization error.

2. To report the error to the report channel

3. To transmit the information further after a set delay.

5.2 Simulation Results

We carried out three simulation runs for each region of oparation and threshold value pair. Out of these three simulation runs for each pair the results displayed here are the ones obtained with highest average error in each case. The results have been displayed in Figure 5.3 through Figure 5.8 as cumulative frequency graphs. Figures 5.3 to 5.5 are the results of setup with node transmission radius of 30 mts. Figures 5.6 to 5.8 are the results with node transmission radius of 50 mts.

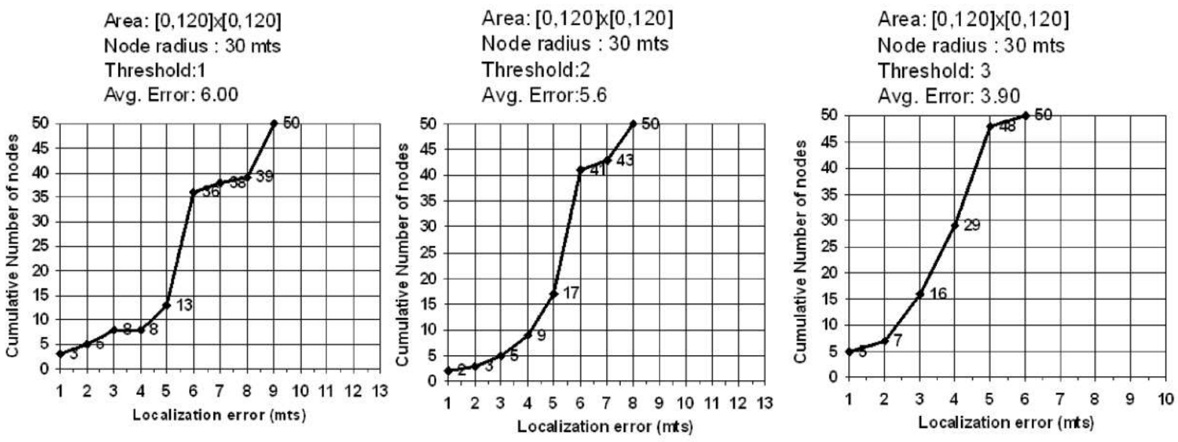

Figure 5.3: Simulation Results: Area[0,120]x[0,120], Node radius 30 mts

Figure 5.3 to Figure 5.8 provide a great deal of insighht into the factors affecting the algoritm. To begin with they clearly depict that the average localization error and the number of nodes that localize depends upon the following parameters:

- Node Density (Number of Nodes deployed / Region of Operation)
- Node TX Range
- Value of Threshold

To understand the effects more clearly we combine the first two parameters to form a relation called coverage that depicts the node density in the transmission area of

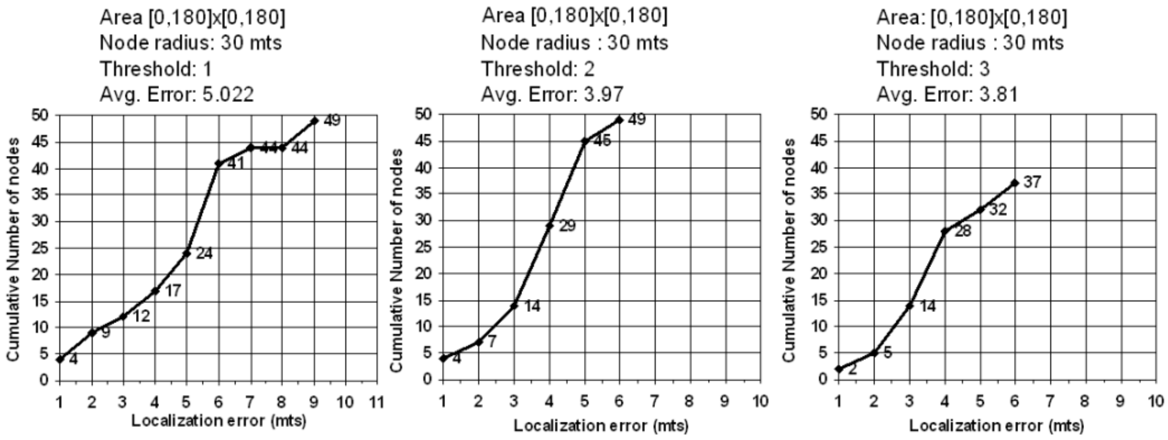

Figure 5.4: Simulation Results: Area[0,180]x[0,180], Node radius 30 mts

one node. The relation coverage is defined as follows:

$$\textbf{Coverage} = \textbf{(Node Density)} \times \textbf{(Node TX Range)} \tag{1}$$

Table 5.1 is a compilation of results shown through the graphs. In order to calculate the percentage of nodes localized in each case we select only those nodes that have achieved localization with an error less than 20% of the node's Tx range.

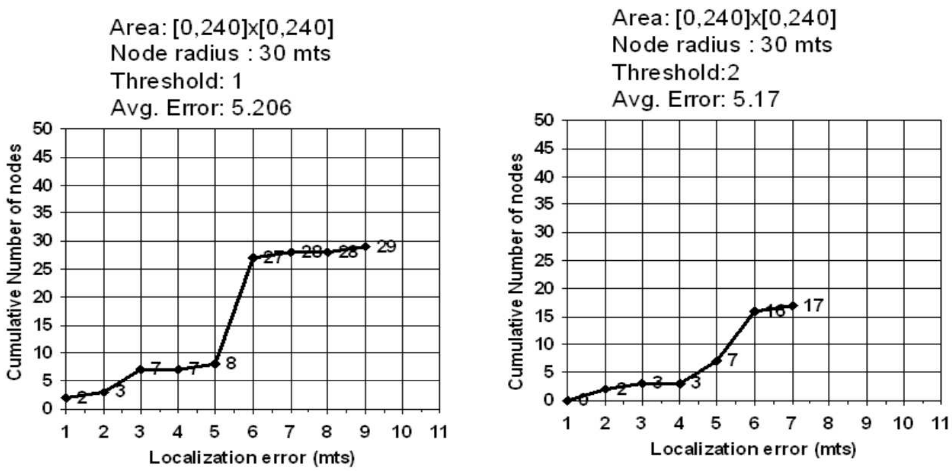

Figure 5.5: Simulation Results: Area[0,240]x[0,240], Node radius 30 mts

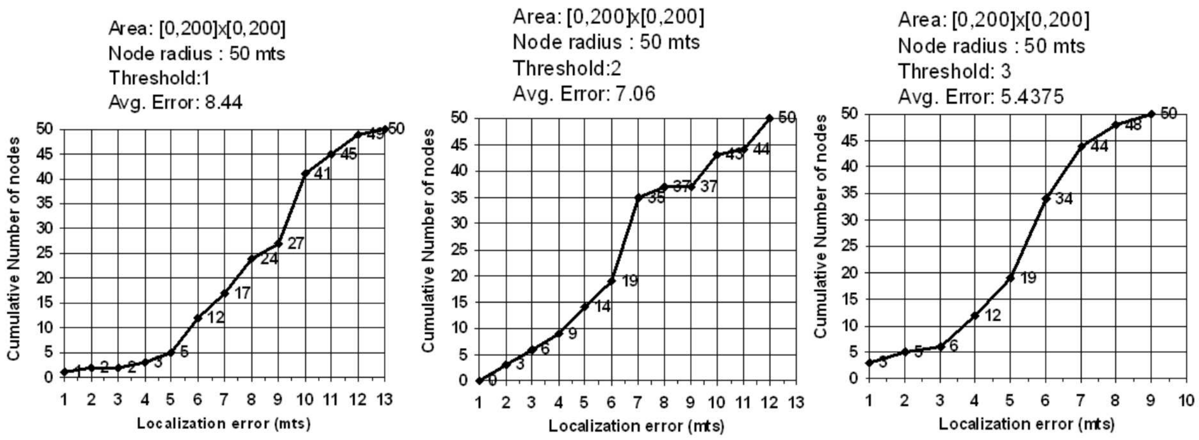

Figure 5.6: Simulation Results: Area[0,200]x[0,200], Node radius 50 mts

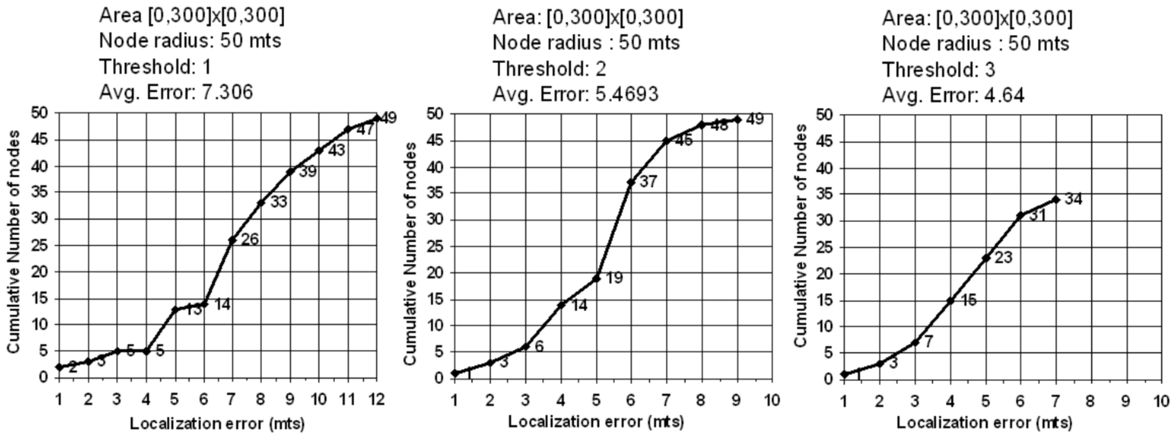

Figure 5.7: Simulation Results: Area[0,300]x[0,300], Node radius 50 mts

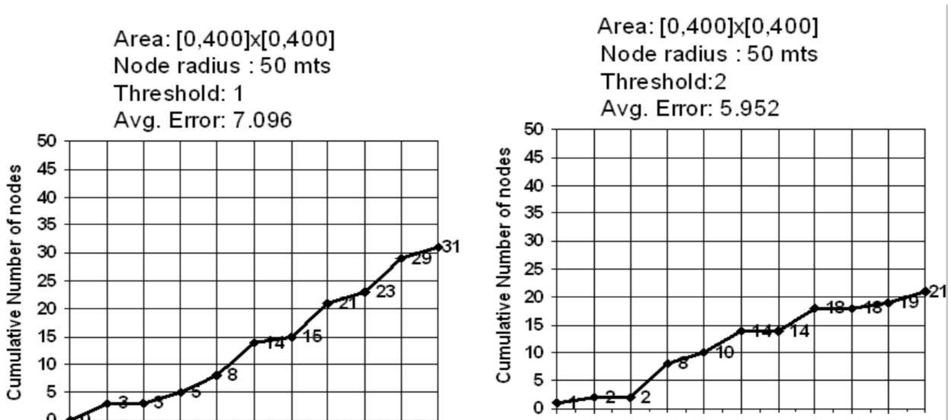

Figure 5.8: Simulation Results: Area[0,400]x[0,400], Node radius 50 mts

5.3 Analysis

The compilation in Table 5.1 shows a few interesting characteristics. Firstly, it is evident that below a certain value of coverage (≈ 4.36) the percentage of nodes that localize sharply drops. Thus, it may be added that this is the minimum value of coverage that has to be maintained for the algorithm to work as desired. Secondly, there is trade off as we increase the value of threshold from 1 to 3. As the value of threshold is increased the average error comes down but so does the percentage of nodes localized.

More precisely, it may be inferred that a value of threshold=2 is good both in terms of percentage of nodes localized and the avg. localization error. Another interesting fact that comes across is that the average error, irrespective of the threshold or coverage, increases as the transmission radius increases. This implies that the proposed algorithm may not be really useful for nodes with high transmission radius.

Having established the optimal selection of coverage and threshold we now address the other important issues that relate to practical implementation of the proposed algorithm. We put forward the following questions that we aim to answer:

- *How well does the algorithm scale with increased area of operation?*
- *Can the algorithm be used in harsh environments such as indoors considering the multipath effect and gray areas produced in RF communication?*
- *What is the effect of unreliable communication on the proposed scheme and what are the measures needed to ensure reliability in communication?*

To address the first issue we carried out simulations for large areas of operation. We however maintained the optimal values of coverage and threshold. The results of simulations have been brought out in Figure 5.9. The results in Figure 5.9 show that the scheme scales well with increase in area. The slight increase in the average error seen in the results can be attributed to propagation of error along with localization information as the network size increases. However, the error propagation effect is

TxRadius	Threshold	AreaofOps	Coverage	%of Nodes localized	Avg.Loc.Error
30	1	[0,120]x[0,120]	9.8125	72	6.0
30	2	[0,120]x[0,120]	9.8125	82	5.6
30	3	[0,120]x[0,120]	9.8125	100	3.9
30	1	[0,180]x[0,180]	4.36	82	5.022
30	2	[0,180]x[0,180]	4.36	98	3.97
30	3	[0,180]x[0,180]	4.36	74	3.81
30	1	[0,240]x[0,240]	2.452	54	5.20
30	2	[0,240]x[0,240]	2.452	32	5.17
50	1	[0,200]x[0,200]	9.8125	82	8.44
50	2	[0,200]x[0,200]	9.8125	86	7.06
50	3	[0,200]x[0,200]	9.8125	100	5.437
50	1	[0,300]x[0,300]	4.358	86	7.306
50	2	[0,300]x[0,300]	4.358	98	5.469
50	3	[0,300]x[0,300]	4.358	68	4.69
50	1	[0,400]x[0,400]	2.452	58	7.096
50	2	[0,400]x[0,400]	2.452	38	5.952

Table 5.1: Compiled Simulation Results

limited to a certain extent by the use of threshold value >1. Also, it may be noted that the percentage of localized nodes reduce as the area of operation increases. Only, 94% nodes localize as the ratio of area of operation becomes [0,1000]x[0,1000].

Next we look at the issue of gray areas and multipath effect in RF communication. A detailed study on the characteristics of RF communication has been carried out in [17]. The study brings out a few important observations. Firstly, the extent of gray areas is less for lower transmission ranges such as those simulated or experimented with in our work. Secondly, the extent of gray areas varies greatly with the environment (indoor, out door, habitat) even for the same transmission range. Thirdly, though it may not be possible to predict the extent of gray areas exactly for different environments but it can be stated with good degree of certainty that for a given transmission range the extent of gray area is the maximum for indoor environment. Moreover, more importantly as shown by the results in [17] the extent of gray areas in indoor environment measures up to $1/3^{rd}$ of the transmission range.

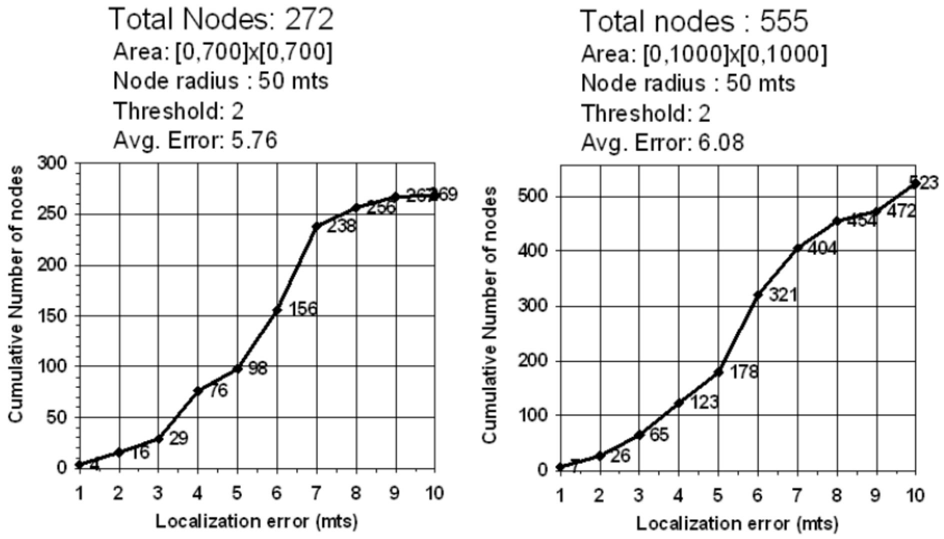

Figure 5.9: Simulation Result: Effect of incresing the area of operation

This implies that for an indoor environment which is the harshest the reliable transmission range of a node effectively reduces to $1/3^{rd}$ of its actual range.We therefore carried out two sets of simulations to study the effect of variable transmission range (varied by $1/3^{rd}$ of actual). In the first set we allowed the nodes in the network to transmit at a random range between 50-65 mts and in the second set from 35-50 mts. The receiver node however in both the cases assumes the range to be 50 mts for its calculations of constraints. The results obtained are shown in Figure 5.10. The results in Figure 5.10 bring an important observation to light. It is interesting to note that by varying the range lower than that assumed by the receiver nodes, not only does the average error increase but the percentage of localized nodes also comes down. In the other case however the effect of varying transmission range is seen only in terms of increased average error which is higher than expected but acceptable. This implies that if we are able to establish the minimum transmission range of a node in presence of gray areas and set the coverage in accordance with the minimum range established, we will still be able to get effective results for the

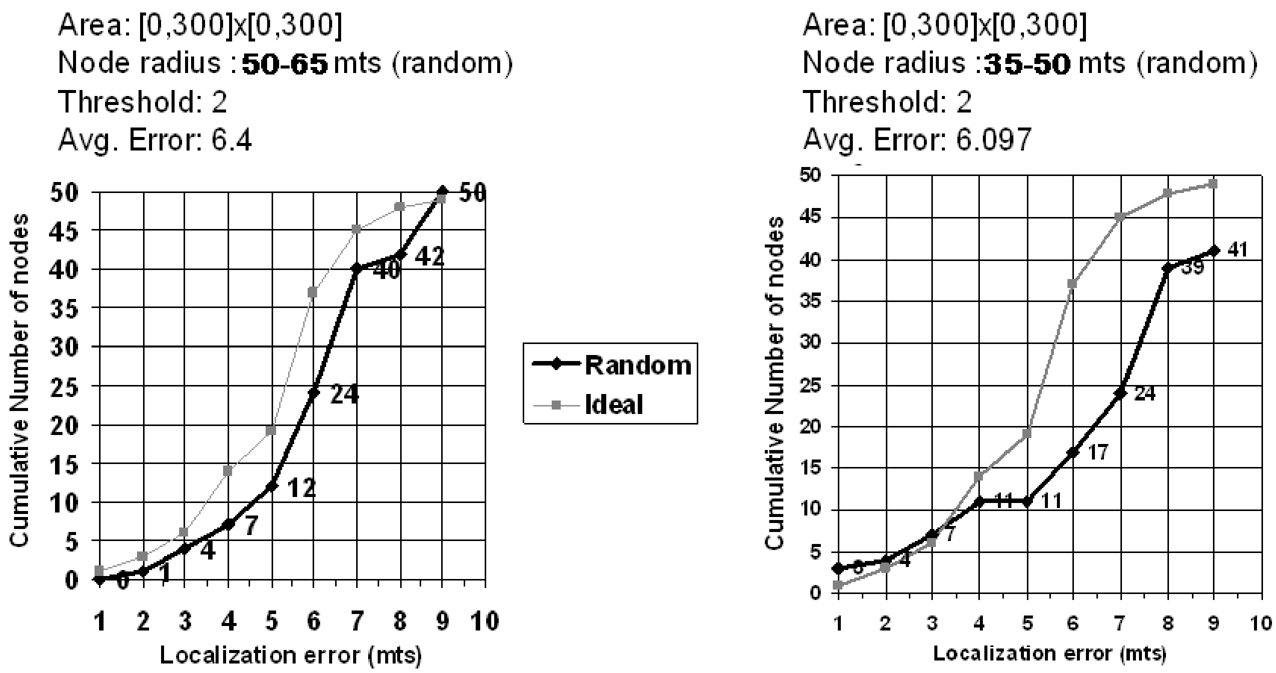

Figure 5.10: Simulation Result: Effect of Random Tx range

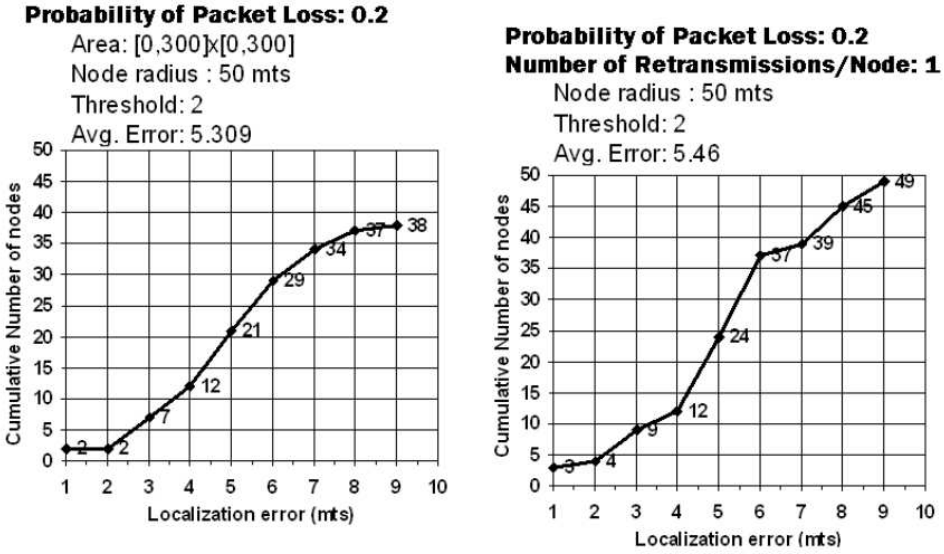

Figure 5.11: Simulation Result: Effect of unreliability in message transmission

proposed localization scheme.

Thirdly, we attempt to answer the question that addresses the importance of reliable communication in the network. To study this effect we carried out two sets of simulations. The first set included simulations with built in probability of transmission error of 0.2 (packet loss/ damage due to collision). The second set included simulations with same error probability but with each node retransmitting its message once. The results obtained are presented in Figure 5.11. The results show that the proposed scheme is reasonably sensitive to loss of packets. Thus an implementation of the proposed scheme would make it imperative to use a minimal collision avoidance scheme and reasonably effective encoding mechanism to start with. The effectiveness of the proposed algorithm can however be improved by making the nodes to retransmit the messages as shown in Figure 5.11.

Finally, in order to analyze the effectiveness of our scheme we compare our work with the adaptive beacon placement algorithm proposed by Bulusu *et al*'s [16]. We first establish the number of nodes required in a given area of [0,100] x [0,100] to

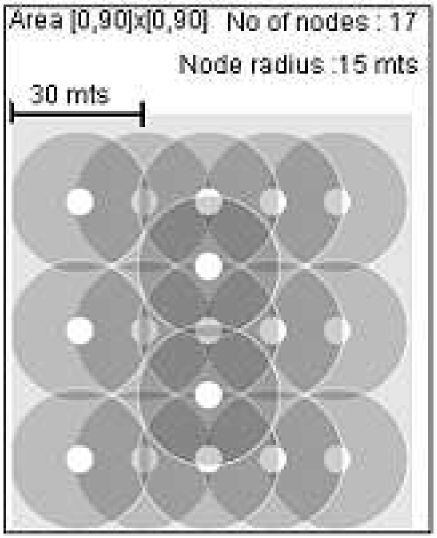

Figure 5.12: Complete Coverage

provide complete coverage. The area of [0,100] x [0,100] and node transmission radius of 15 meters has been chosen keeping in mind the work in [16]. As can be seen in Figure 5.12 at least 17 nodes with transmission radius of 15 meters are required to achieve complete coverage in an area of [0,100] x [0,100]. That is to say that this is the minimum number of nodes needed to be deployed in the given area of operation to provide complete coverage. However, in order to propagate localization to the entire region of operation of [0,100] x [0,100] through our proposed algorithm, keeping the value of coverage = 4.36, we require a total of 61 nodes. This shows that our proposed algorithm inflicts an overhead of 44 nodes in this special case. Further, on carrying out the simulation with 61 nodes spread randomly uniformly in a region of operation of [0,100] x [0,100], a threshold value of 2, and with a node transmission radius of 15m we obtained the results as shown in Figure5.13. The results of simulation show that we obtain an average localization error of 3.56. It is evident by the simulation results in [16](as seen in Figure2.2) that the beacon deployment density (per sq.m) required to achieve an average localization error of 3.56 under the same coverage and region criteria is 0.009. This density in the above area of operation of [0,100] x [0,100] evaluates to deployment of 90 beacons. This

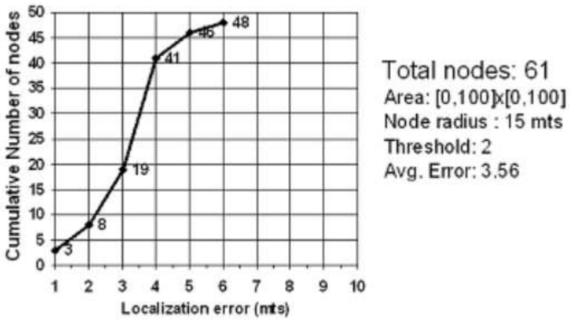

Figure 5.13: Result of algorithm run under same conditions as [16]

comparison proves that the 44 additional nodes required by our proposed algorithm over the ones that may have been required just to provide complete coverage in a given area is reasonably lesser than the additional number of beacons required in [16] to provide the same localization.

Chapter 6

Experiment Setup and Results

In order to verify the results of our simulations we carried out real life experimentation of the proposed algorithm with nodes as shown in Figure 1.2. The sensor nodes were spread randomly over an area of operation of [0,100] X [0,100] meters of a football ground. The sensor nodes are made up of an onboard *Atmel AT89C52* microcontroller, a radiometrix *BiM-433-F* radio TX/RX operating at 433 MHz frequency, a voltage stabilization circuit and a 9V battery. The block diagram of the sensor node is as shown in Figure 6.1.

6.1 Hardware Details

The *AT89C52* is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The device is compatible with the industry-standard 80C51 and 80C52 instruction set and pin out. The on-chip flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. The radiometrix module integrates a low power UHF transmitter and matching superheat receiver together with the data recovery and TX/RX change over circuits to provide a low cost solution to implementing a bidirectional short range radio data link. The data rates (up to 40kbit/s) and fast TX/RX changeover ($<$1ms) make the BiM transceiver ideal for high integrity one to one links / multi-node packet switch networks. Rapid

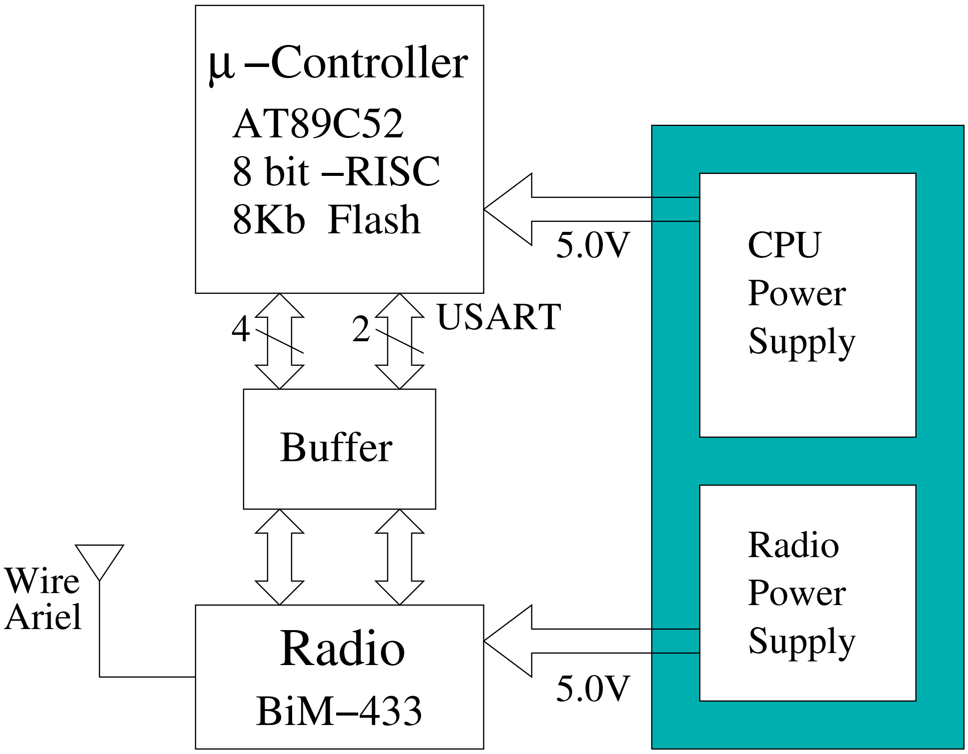

Figure 6.1: Block diagram of Sensor Node developed

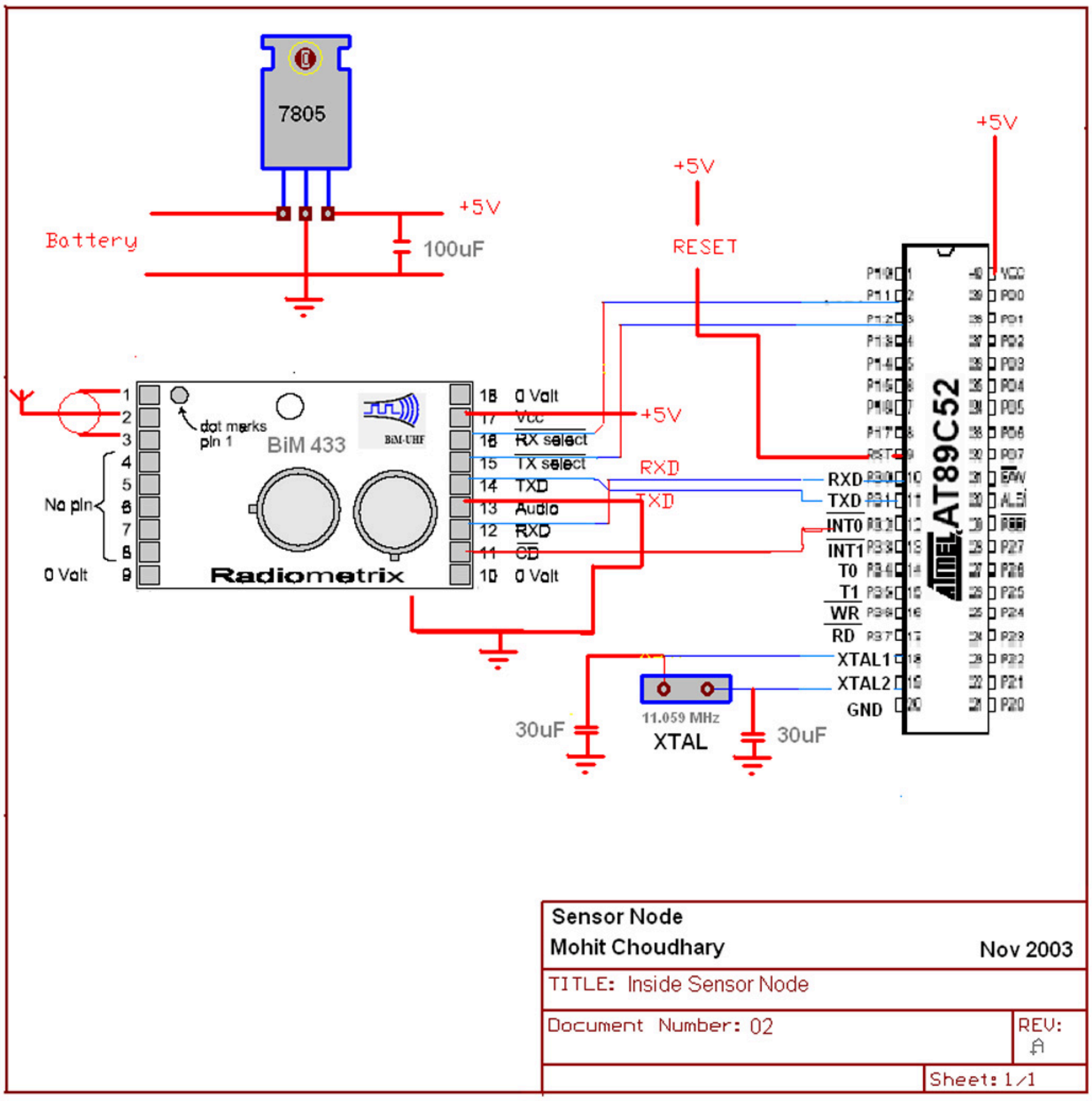

Figure 6.2: Circuit diagram of Sensor Node developed

RX power up (<1ms) allows effective duty cycle power saving of the receiver for sensor network applications. The circuit details of the developed sensor node are given in Figure 6.2.

6.2 Communicaton Protocol Details

We built the communication stack from scratch. The microcontrollers on all the sensor nodes were configured to communicate with the radio in UART mode 1 [15] at 9600 Bauds. As can be seen in Figure 6.2, the RXD and TXD (UART pins) of the microcontroller are connected to the same pins of the radio. Further, we make use of pins P1_1 and P1_2 of the microcontroller to select the radio to transmit or receive. Lastly, the pin CD(carrier detect) of the radio is connected to INT0(the highest priority interrupt of the microcontroller) to assist this changeover from transmit mode to receive mode. Keeping in mind the requirements of the radiometrix chip, the packet structure for data transfer between nodes was as follows(shown in Figure 6.3):

PREAMBLE	DATA	CONTROL	CRC

Figure 6.3: Packet Structure

Preamble: This is mandatory for the receiver in the radiometrix chip to stabilize. The chip requires 3ms to stabilize. Thus at 9600 Bauds we used four bytes of Preamble.

Control: We used a unique Byte pattern (0xAA) to indicate the start of message (frame sync.).

Data: The data as used in the actual implementation is as shown in Figure6.4.

CRC: 16 Bit Checksum of control - data fields has been used by the decoder to verify the integrity of the packet.

1	2	3	4	5
NID	a	b	MID	NCS

Figure 6.4: Data Field

To conduct the experiment effectively we had to first establish the accurate minimum transmission range per node and the gray areas . For this purpose we carried out the range measurements. We recorded the packets received at three different points on the circumference of an imaginary circle. The circle originated at the transmitting node and was incrementally varied starting from 15 mts in steps of one meter radius. The results of the spatial profile of the node has been shown in Figure 6.5. The Figure 6.5 shows that the packet reception drops after approx. 30 meter radius and thus for the purpose of our experimentation we assume the minimum range of the node as 30 meters. Once the minimum range was established we calculated the node density required for the optimal value of coverage = 4.36.
Node Density = 4.36/ (3.14x30x30)
Node Density = 0.00154
Thus, for area of 10,000 sq meters (100x100), the number of nodes required would be
Number of nodes required = $(Nodedensity) \times (Area)$
Number of nodes = 0.00154×10000
Number of nodes = 15.4

6.3 Experiment Details

To conduct the experiment we had four sensor nodes available with us. In order to meet the requirement of 15 nodes and three beacons with the four nodes physically available with us we used the technique of rotating the nodes that had once achieved localization. The nodes were programmed to output there x and y co-ordinates on

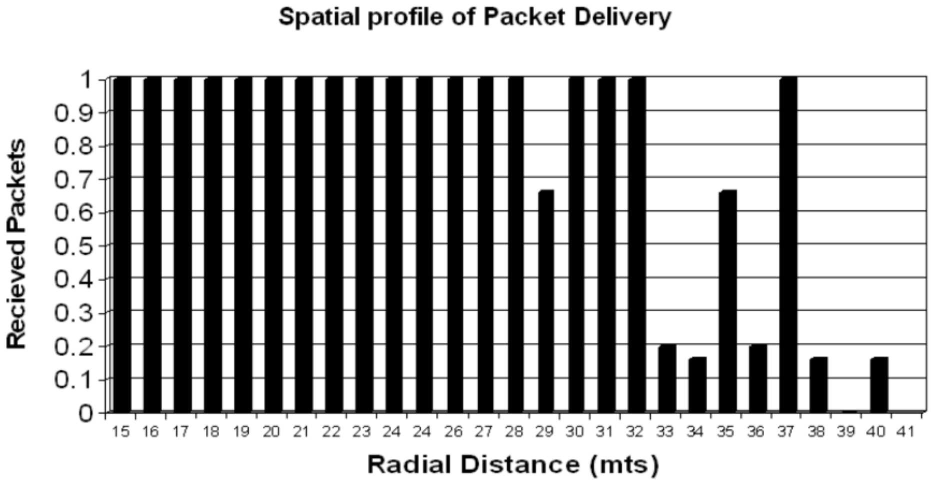

Figure 6.5: Spatial Range Profile

the I/O ports 0 and 2 respectively once localized. We polled the nodes periodically to check for the required output. Once the localization results were available on a node's output port we reset the node and placed it further down in the region of operation.

We do not claim that our experiments or our environment or the particular conditions under which we conducted our experiments cover all the possible deployment scenarios for sensor networks. But we are fairly confident that our experimental conditions were not pathological either and consider them sufficient to provide an evidence of the fact that the algorithm is a simple distributed mechanism that may be implemented in real-life. The experiments were conducted on different days. In conducting our experiments, we tried to keep the environment's gross characteristics as consistent as possible (in addition to making sure we were able to replicate placement exactly, using markers).

We had four nodes available with us (Node 1, 2, 3 and 4). To start with we used node 1 as a beacon transmitting its packets from three different initial beacon positions. We placed node 2 in the region of overlap of the three beacons. Node 3 was initially placed at a point situated at a random distance between 0 - 25 meters (25m is used to maintain a randomly uniform node density in the entire region of operation) from node 2 and also at a random angle from it. Node 4 was then placed at a point using the same criteria as above with respect to node 3. On localization of either one or both of nodes 3 or 4, post transmission of node 2, both node 2 and the newly localized node are moved further down in the region of operation. If in case two or more nodes get localized to the same point, only one node out of such a group is used to propagate its position further with different node ID's and the rest of the nodes of the group are positioned further. Thus the method employed to conduct the experiment made use of the four physical nodes available with us to produce the effect of 15 nodes required.

6.4 Experiment Results

The results obtained for two different runs both for the [0,100] x [0,100] area of operation are enumerated in Figure 6.6. In the first run all the 15 nodes localized themselves correctly but in the second run one node failed to achieve localization. The results obtained represented in Figure 6.6 confirm that the proposed scheme can provide localization within reasonable error accuracy and for a large percentage of nodes in the network even in real life scenario provided the value of threshold and a coverage of **4.358** is maintained for a given combination of node set and area.

6.5 Conclusions and Future Work

In this thesis, we described the design, implementation, simulation, analysis, and experimental evaluation of a novel method for localization through self-propagation in distributed wireless sensor networks. There are many lessons we can draw from our preliminary experimentation. The above method of localization is a complete and reliable process in itself. Secondly, applications based on sensor networks requiring reasonable localization error accuracy (true for most of the sensor net applications today) can effectively utilize the proposed method to cut down on additional cost of

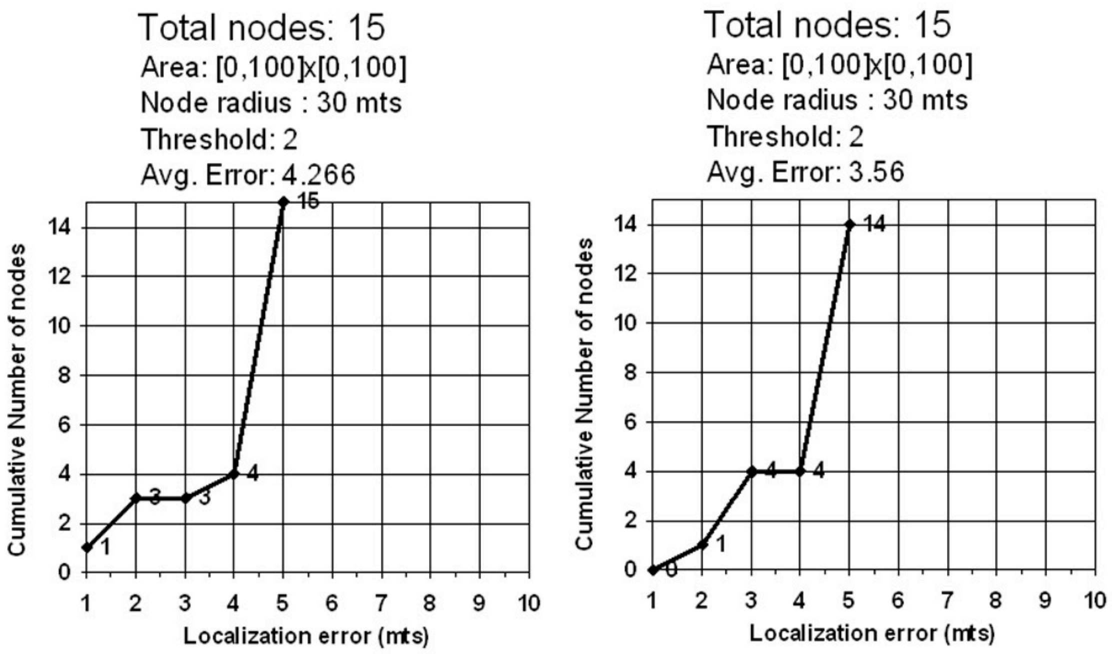

Figure 6.6: Experiment Results

localization otherwise incurred. Thirdly, as a few numbers of nodes in each region localize to a common point by the proposed method, this information can be further utilized by the applications to either perform power management by utilizing a very small number of such nodes at one time or perform easy clustering of all such nodes. The future work includes analysing the effect of using other radio models such as a hexagon instead of a square on the proposed scheme. We would also attempt to extend the methodology to 3D networks. On the hardware level as a first step we would attempt to integrate the developed node with temperature sensors in form of DS 1631 (Dellas semiconductor digital thermometer chip). As a second step we aim to integrate self developed RF-id reader to the node as onboard sensor.

Bibliography

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. *Scalable coordination in sensor networks.* In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99), pages 263-270, Seattle, Washington, USA, August 15-20 1999.*

[2] N. Bulusu, J. Heidemann, and D. Estrin. *GPS-less low cost outdoor localization for very small devices.* In *IEEE Personal Communications Magazine, 7(5):28-34, October 2000. Magazine, 4(5):42-47, October 1997.*

[3] B. Hofmann-Wellenhoff, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice.* In *Fourth Edition. Springer Verlag, 1997.*

[4] H. Koshima and J. Hoshen. *Personal locator services emerge.* In *IEEE Spectrum, 37(2):41-48, February 2000.*

[5] N. Priyantha, A. Chakraborty, and H. Balakrishnan. *The cricket location support system.* In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), pages 32-43, Boston, Massachusetts, USA, August 6-11 2000.*

[6] R. Want, A. Hopper, V. Falcao, and J. Gibbons. *The active badge location system.* In *ACM Transactions on Information Systems, 10(1):91-102, January 1992.*

[7] L. Girod. *Development and characterization of an acoustic rangefinder* In *Technical Report USC-CS-TR-00-728, University of Southern California, April 2000.*

[8] A. Ward, A. Jones, and A. Hopper. *A new location technique for the active office.* In *IEEE Personal Communications Magazine, 4(5):42-47, October 1997.*

[9] F. Koushanfar, A. Savvides, G. Veltri, M. Potkonjak, and M. B. Srivastava. *Iterative location discovery in ad hoc sensor networks.* In *Slides from UCLA CS review, April 2000. http://www.cs.ucla.edu/farinaz/location2 1.ppt*

[10] P. Bahl and V. N. Padmanabhan. *Radar: An in-building user location and tracking system.* In *In Proceedings of the IEEE Infocom 2000, volume 2, pages 775-84, Tel-Aviv, Israel, March 2000.*

[11] L. Doherty, K. S. J. Pister, and L. El Ghaoui. *Convex position estimation in wireless sensor networks.* In *Infocom 2001, Anchorage, AK, 2001.*

[12] Aram Galstyan, Bhaskar Krishnamachari, Kristina Lerman, and Sundeep Pattem. *Distributed Online localization in Sensor Networks Using a Moving Target* In *Proceedings of the third international symposium on Information processing in sensor networks, pages: 61 - 70 2004.*

[13] Nirupama Bulusu, Deborah Estrin, Lewis Girod John Heidemann. *Scalable Coordination for Wireless Sensor Networks: self-Configuring Localization Systems.* In *Proc. of the 6th International Symposium on Communication Theory and Applications (ISCTA'01), Ambleside, UK, July 2001.*

[14] Edward A. Lee. *Overview of the Ptolemy Project.* In *Technical Memorandum No. UCB/ERL M03/25, University of California, Berkeley, CA, 94720, USA, July 2, 2003.*

[15] *Atmel AT89C52 Datasheet.* In *www.atmel.com/atmel/acrobat/doc0313.pdf.*

[16] Nirupama Bulusu, John Heidemann, Deborah Estrin. *Adaptive Beacon Placement.* In *Proc. of the 21st International Conference on Distributed Computing Systems page:489, Phoenix, Arizona, April 2001.*

[17] Zhao J. and Govindan R. *Understanding Packet Delivery Performance in Dense Wireless Sensor Networks.* In *Proceedings of the First International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, 2003.*

[18] Karim Seada and Ahmed Helmy and Ramesh Govindan. *On the effect of localization errors on geographic face routing in sensor networks.* In *Proceedings of the third international symposium on Information processing in sensor networks,71–80, Berkeley, California, USA, 2004.*

[19] Chris Savarese, Jan Rabaey, and Koen ngendoen. *Robust positioning algorithms for distributed ad-hoc wireless sensor networks.* In *In USENIX Annual Tech. Conf., pages 317-327, Monterey, CA, June 2002.*

[20] Neal Patwari, Alfred O. Hero III, Matt Perkins, Neiyer S. Correal, and Robert J. O'Dea. *Relative location estimation in wireless sensor networks.* In *IEEE Trans. Signal Process., 51(8):2137-2148, August 2003.*

[21] Deborah Estrin, Ramesh Govindan, and John Heidemann. *Embedding the internet:introduction.* In *Commun. ACM, 43(5):38{41, 2000.}*

[22] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. *Habitat Monitoring: Application Driver for Wireless Communications Technology.* In *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, April 2001*

[23] Christopher Sadler, Pei Zhang, Margaret Martonosi, Stephen Lyon. *Hardware Design Experiences in ZebraNet.* In *ACM Sensys, November 2004*

[24] Narasi Sridhar. *Corrosion Sensors and Integrity Monitoring.* In *http://www.swri.edu/4org/d18/mateng/corrosn/sensors.htm.*

[25] Ning Xu, Sumit Rangwala, Krishna Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, Deborah. *A Wireless Sensor Network for Structural Monitoring.* In *ACM Sensys, November 2004*