CMSC 684 WSN, 2018 FALL, Dr. Younis

Project Report

Dayuan Tan, Haoran Ren

Dec 14, 2018

RSSI based Localization Application for Wireless Sensor Networks

1. Abstract

Wireless Sensor Networks (WSN) has been widely applied in lots of fields, like agriculture, military, construction and so on. Localization is one of the most important issues, since it is essential and indispensable for the majority of practical applications. As one of the most popular and widely used localization algorithms, RSSI (Received Signal Strength Indicator) localization algorithm is easily to be implemented and has good localization precision. In order to have a better understanding of RSSI localization algorithm, as well as those problems that comes with it, we built a test-bed to implement this algorithm and analyzed its performance.

Key Words: *WSN, Localization, RSSI.*

2. Introduction

Wireless Sensor Networks (WSN) has been a hot topic for decades of years, and more and more popular with economic development as well as cost reduction. It consists of large amount of nodes, collecting information form the environment by sensors, communicating with a each other, and implementing their designed functions. Localization information is always important for always all WSNs applications, since only knowing the collected data without knowing where it was collected seems useless, especially when people want to execute some reactions after analyzing the collected data.

There are several methods to locate those nodes in WSNs. Global Positioning System (GPS) is a famous approach for solving this problem, but it’s expensive and impractical to equip every node in WSNs with it. That’s why we need special approached for WSNs. Usually existing approaches can be classified into Range-based approaches or Range-free approaches. Range-based approaches use distances or angles information between points and points obtained from special hardware to calculate the distances of different nodes, which will be used to calculate the coordinates of nodes later. Some famous Range-based approaches are received signal strength indicator (RSSI), time of arrival (TOA), time difference of arrival (TDOA), and angle of arrival (AOA). Range-free approaches do not need absolute range information. The location precision is not as good as range-based approaches but the hardware requirement is less. There are many range-free approaches such as centroid algorithm, DV-hop, amorphous, APIT and MDS-MAP.

RSSI is one of the most famous and widely used approach in real world WSNs applications. It is easily to be implemented and has good enough localization precision. It calculates the distance between two nodes after obtaining the RSSI value. A relationship between RSSI values and distance will be constructed, which will be elaborated in section 5.

The report is organized as follows. In section 2, we will introduce the theory of RSSI localization algorithm and what we actually did in our implementation. In Section 4 the environment and hardware we used for our implementation will be introduced. Section 5 elaborates the relationship between RSSI value and distance and how we calculate the distance after we have RSSI value. In section 6 we will discuss the performance of our implementation as well as how we meet the criteria.

3. Theory Background

RSSI value, which can be obtained by hardware, is widely used to represent distance since it has a relationship with distance as formula (1). After transforming form formula (1), we get formula (2) which can be used to calculate distance.

$$P_{recv} = c\,(P_{tx} / d^a) \quad (1)$$

$$d = \{ (c \cdot P_{tx}) / (P_{recv}) \}^{(1/a)} \quad (2)$$

Trilateration algorithm is a class algorithm to locate one node when we have the positions of more than three nodes. As shown in Figure 1, assume we have the distance r1, r2 and r3 between the Unknown node which is in the center, and three Beacon nodes which know their coordinates are (x1, y1), (x2, y2) and (x3, y3). Assume the coordinates of Unknown node is (x_u, y_u), it is obvious that we can have following equation system:

$$(x1-x_u)^2 + (y1-y_u)^2 = r1^2 \quad (3)$$

$$(x2-x_u)^2 + (y2-y_u)^2 = r2^2 \quad (4)$$

$$(x3-x_u)^2 + (y3-y_u)^2 = r3^2 \quad (5)$$

Subtracting equation (5) from (3) and (4), and then rearrange terms we can get linear equations in (x_u, y_u):

$$2(x3-x1)x_u + 2(y3-y1)y_u = (r1^2 - r3^2) - (x1^2-x3^2) - (y1^2-y3^2) \quad (6)$$

$$2(x3-x2)x_u + 2(y3-y2)y_u = (r2^2 - r3^2) - (x2^2-x3^2) - (y2^2-y3^2) \quad (7)$$

Then x_u and y_u can be easily solved.

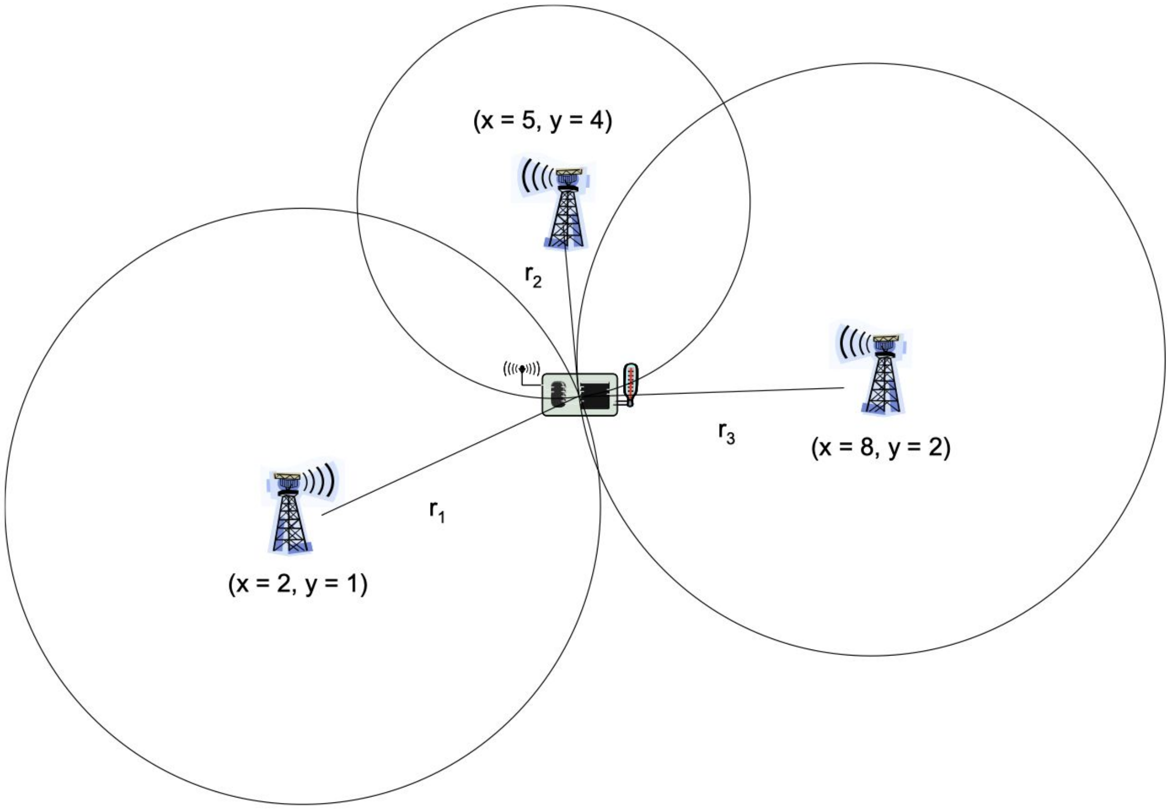

Figure 1: Trilateration algorithm.

4. Implementation

We developed our software on Tiny-OS 2.1.2. The mote we use is iris mote XM2110CB. The programming board is MIB520.

In our test-bed, four motes will be used as Beacon nodes, which know their coordinates, while one more mote will be used as Unknown node, which needs to use RSSI algorithm to calculate its estimated coordinates, as shown in Figure 2.

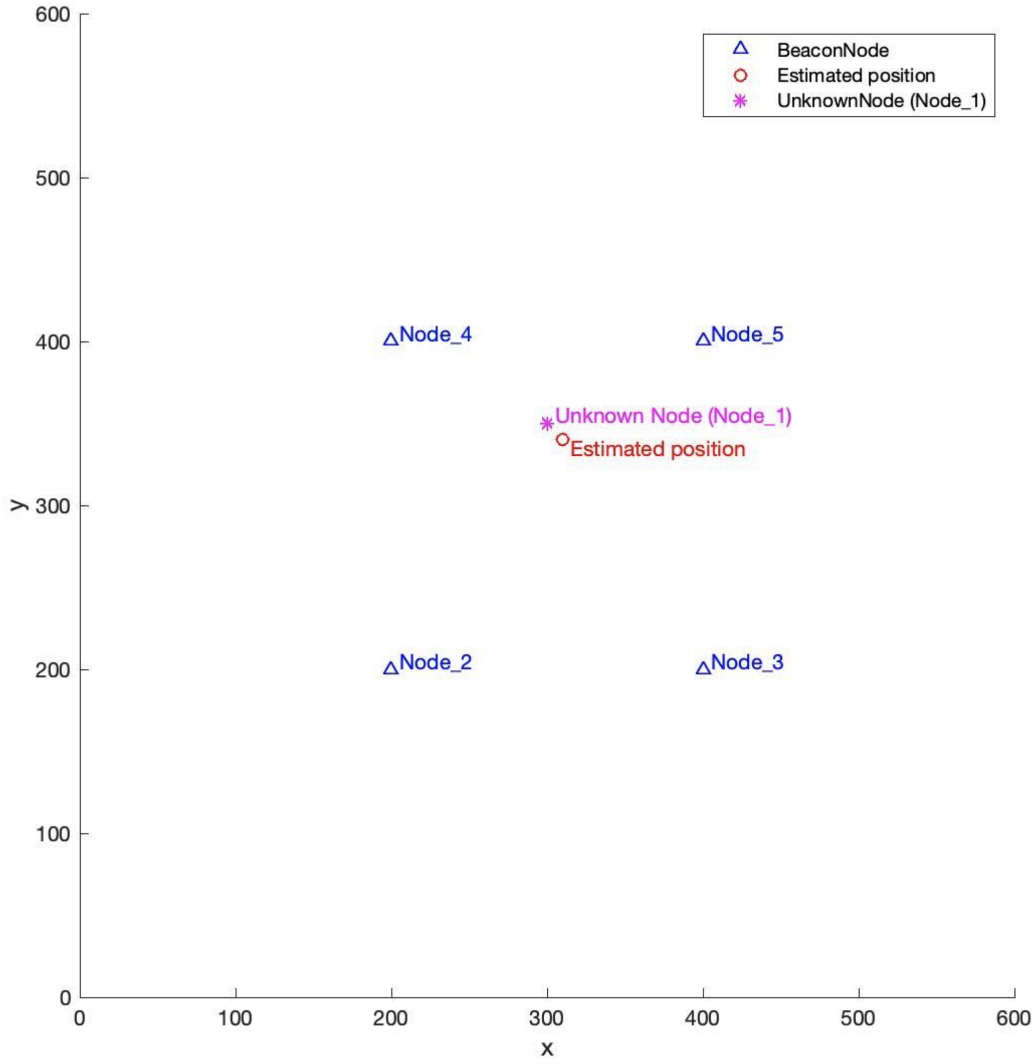

Figure 2: Test-bed diagram.

After Unknown node reads a RSSI value of the packet transmitted from a Beacon node, a fitting function representing the RSSI value and distance correspondence relationship specially for that transmitting mote will be used to calculate the estimated distance between Unknown node and that transmitting node, which will be used in trilateration algorithm later. This fitting functions for each Beacon nodes will be elaborated in Section 5.

When Unknown node has three distances to Beacon nodes, trilateration algorithm will be used to calculate the estimated coordinates of the Unknown node, using formula (6) and (7). Since in our test-bed the Unknown node can actually collect four RSSI values form four Beacon nodes, thus four distances to them can be obtained. Hence four triangles exists in this system, which means trilateration algorithm will be used four times (one time for each triangle) to provide Unknown node with four estimated coordinates, ad shown in Figure 3. The average value of those four estimated coordinates will be calculated as the final estimated coordinates.

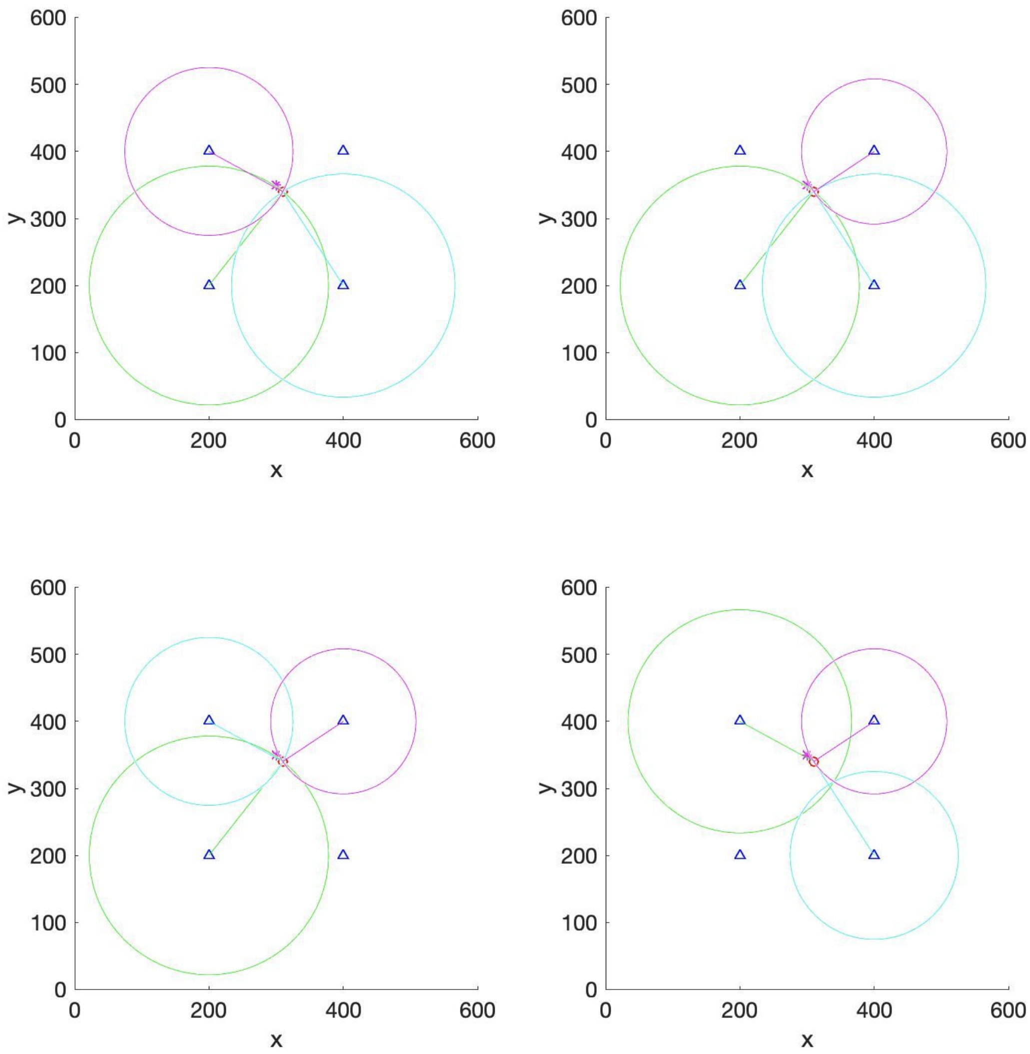

Figure 3: Calculate the estimated coordinates with trilateration.

The program on motes are wrote in NesC on TinyOS 2.1.2. The Unknown node will also transmit its received RSSI value to a serial port of our laptop. The program receiving message from serial port, get corresponding distance through fitting functions, and calculating the estimated coordinates using trilateration algorithm, as well as displaying the results are wrote in Java on Ubuntu 12.04.

5. RSSI-Distance Relationship

Instead of use ideal formula, like formula (2), to calculate the distance by using RSSI value, we chose to measure the practical relationship between RSSI value and distances. Firstly, we placed our mote at the position which has distance *d* to a transmitting mote, then read the RSSI value and measure this distance, as shown in Figure 4. So we have a RSSI value and distance correspondence table for the transmitting mote, as shown in Table 1. Repeat this step we can get the RSSI value and distance correspondence table for each transmitting mote. Secondly, we drew the picture of those table and fitted them into cubic polynomial function, which are shown in Figure 5. So in our test-bed, after we have RSSI value, we use these functions to calculate the distance.

Figure 4: Collect RSSI value and Distance correspondence tables for each node.

Table 1: RSSI value and distance correspondence table for each mote.

Nodeid	2	3	4	5
Distance (CM)	RSSI value	RSSI value	RSSI value	RSSI value
0	69	72	71	68
5	71	72	67	69
10	66	66	57	63
15	60	63	56	57
20	57	62	54	57
25	54	60	54	55

30	54	60	52	53
40	49	54	48	51
50	48	51	46	47
60	45	48	45	44
70	45	48	42	42
80	42	43	41	40
90	42	41	39	39
100	39	39	36	38
110	36	38	35	37
120	39	39	33	36
130	35	39	33	34
140	32	34	30	33
150	30	31	30	32
160	32	30	30	30
170	30	30	30	30
180	30	30	27	28
190	29	27	26	25
200	28	24	28	24
210	27	24	28	24
220	26	24	27	24
230	25	27	27	23
240	24	24	22	24
250	21	24	23	24
260	18	26	24	21
270	18	24	23	21
280	18	24	23	21

290	23	24	21	19
300	21	24	21	18
320	21	24	20	18
340	18	24	19	17
360	17	21	18	18
380	15	18	18	15
400	14	21	18	13
420	15	18	17	18
440	15	17	15	16
460	14	15	12	18
480	14	15	15	15
500	14	10	12	12

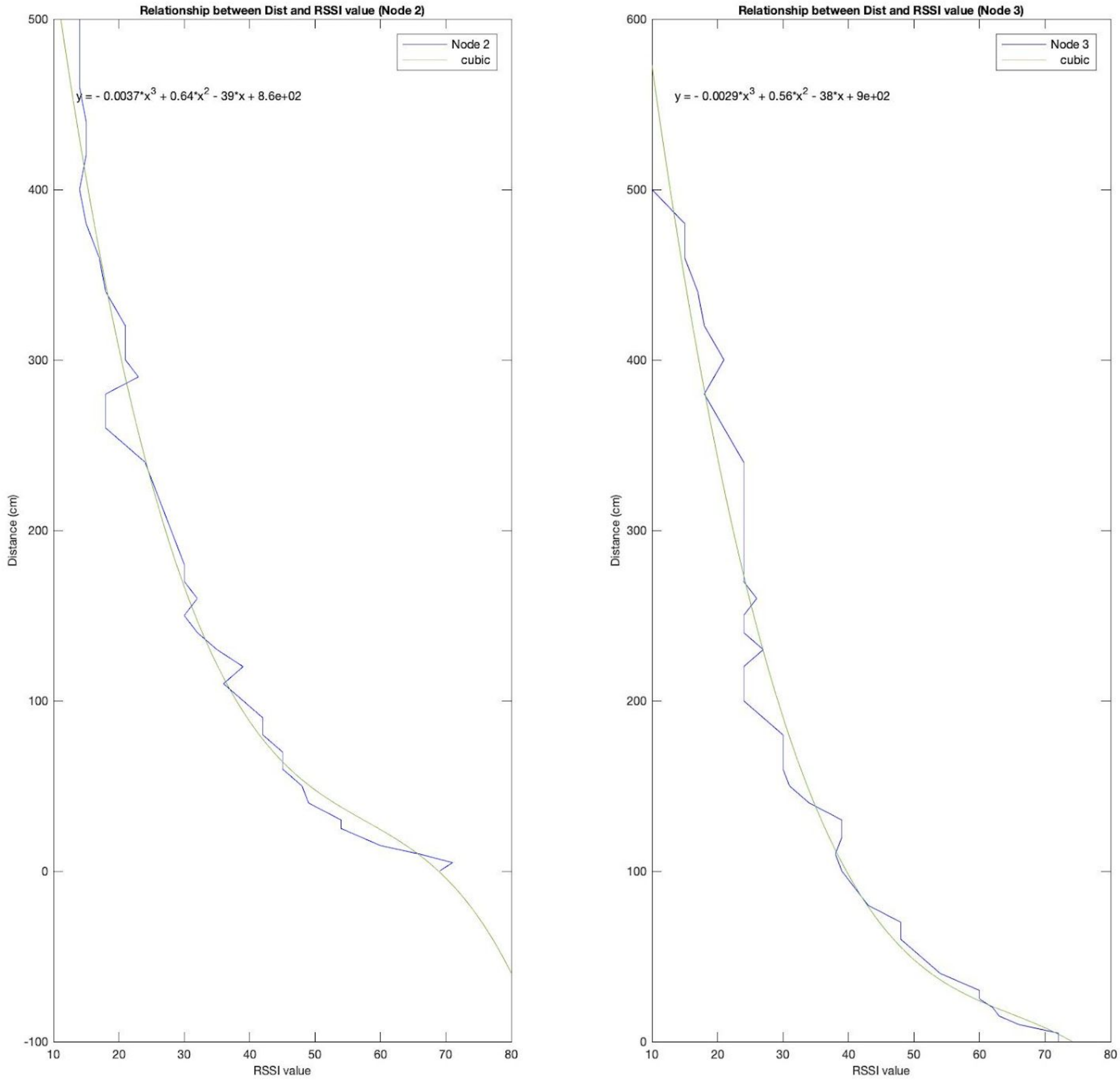

Figure 5 (a): Cubic fitting polynomial function for node 2; (b): Cubic fitting polynomial function for node 3.

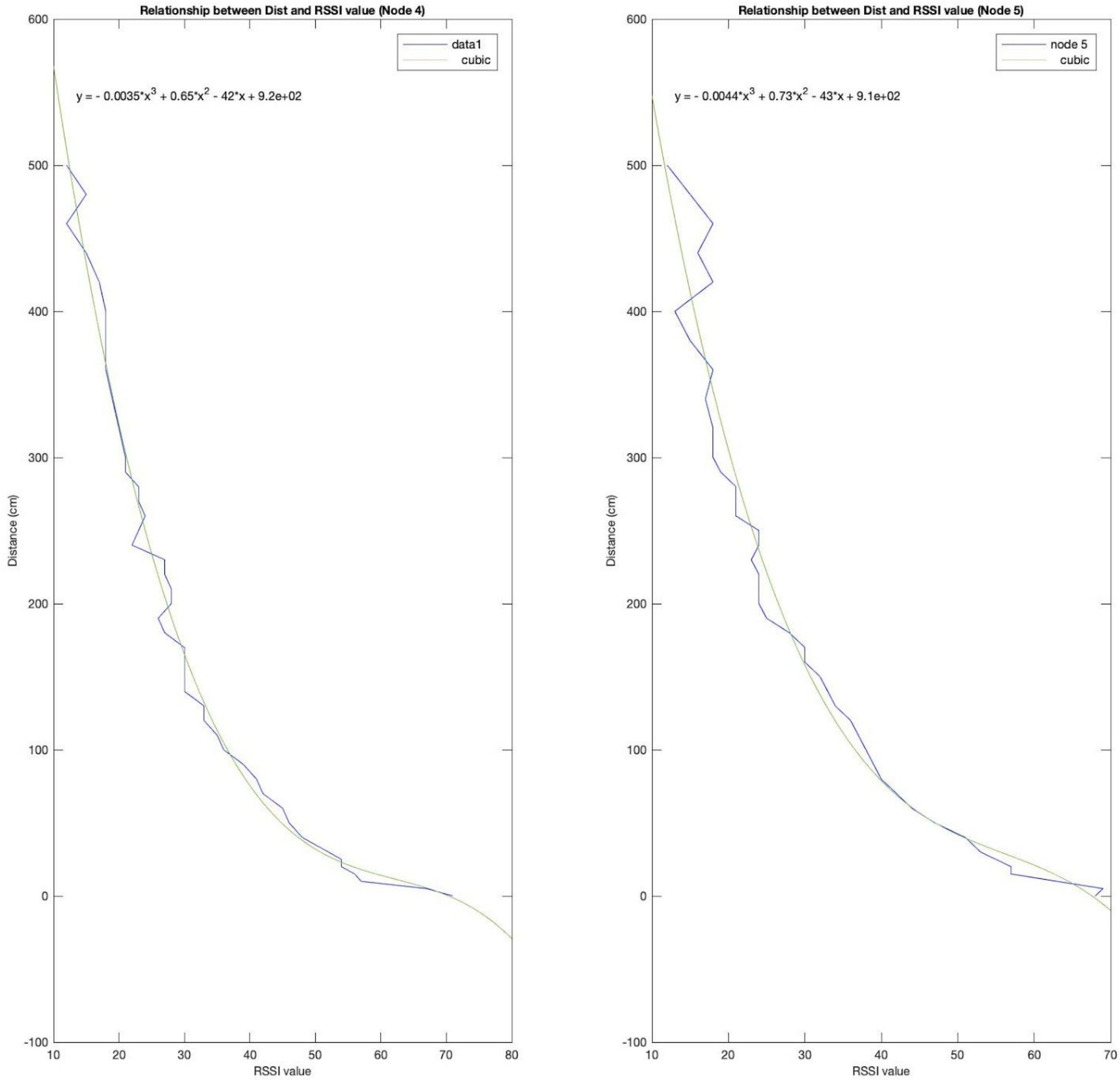

Figure 5 (c): Cubic fitting polynomial function for node 4; (d): Cubic fitting polynomial function for node 5.

6. Performance

When we did our experiments using our test-bed, we placed our Beacon nodes in an area of 200*200 cm^2. The distance error between real Unknown node position and estimated position can be around 80 cm.

Reasons to distance error:

After analyzing the environment and our experiment data, we think the influencing factor to the performance may be following points:

a. WiFi signals around. Since the building we did our experiment is full filled with strong WiFi signals, and motes can also read the RSSI value of WiFI signals, it may affect our received RSSI value used for computing.
b. Interference: Other singals like WiFI and RF broadcast signals may interfere those signals transmiited by Beacon nodes.
c. Reflection: Signals received by Unknown node may are those signals which were reflected by walls, people or other obstacles, instead of directed signals from Beacon nodes. Hence the RSSI values Unknown node obtained can not represent the distance between it and Beacon node, but are still used for computing, since we can not differ them.
d. Battery used for motes: New and old batteries are mix used during our experiment and we can not differ them. So batteries' performance in experiment is not exactly as same as it was when we did the measurement for RSSI value and distance correspondence table. Hence the power of antennae are not same.

7. Code

All code will be zip in a separate file which includes a README to introduce how to run it. Our code includes three part. First part should be burn into Beacon nodes, which transmits packet broadcast. Second part should be burn into Unknown node, which receives the packets and read RSSI and then write them into serial port to PC. Third part should run on PC, which read data from serial port and find correspondent distance and then calculate estimated coordinates. .

8. Performance

a. Understanding the problem and its complexity.

The difficulties in the project can be classified into four points: Communication between motes, communication between a mote and PC, the correspondence relationship between RSSI value and distance and calculation of coordinates using trilateration.

Our test-bed is just a simple implementation of RSSI localization approach. It can be much more complex if we want to use it in practical WSN applications. Those reasons to distance error explained above should also be taken in account and solve.

b. Application of the course material.

The idea of localization approach comes form Dr. Younis's slides, as well as the RSSI formula and trilateration formula. Homework 3 is used for us as the beginning course to NesC programming. All hardware comes from professor. So we think we apply the course material very well.

c. Identifying the design alternatives.

d. Feasibility (practicality) of the approach.

The RSSI approach for localization problem is totally practical. Actually it has been used for lots of applications, especially indoor localization. Our test-bed also works well and actually it can work well for some practical applications like locating cars in parking lots or locating all wheelchairs in hospital or airport.

e. Demonstration of data backed decisions.

Instead of using ideal formula for RSSI and distance, we measured and build a special table to represent the correspondent relationship between them. This helps us to have a good location precision.

f. Ability to defend the data.

We think our data are good. For the RSSI and distance correspondent table we measured the RSSI every 5 cm or 10cm for each mote, which gives us practical and reliable data.

g. Compelling nature of the argument.

As this report shows, form introduction to theory background and formula demonstration, then implementation detail, lastly experiment performance, all are compelling.

h. Organization / Completeness of the presentation.

The organization is good and It's complete.

i. Creativity, development of a unique approach.

Even though the RSSI localization approach is not creative, the correspondent table part is creative. Using four trilateration instead of only one is also a development for original trilateration algorithm.