An empirical investigation into a location sensing system based on RF signal strength of TinyOS motes

By Kevin Pereira
kevinp@cs.utexas.edu

Advisor: Aloysious Mok
<mok@cs.utexas.edu>

Computer Science Research and Writing
CS 379H
Fall 2002

Abstract:-

With the increasing usage of handheld, mobile wireless devices and the unprecedented popularity of celluar phones, the ability to know where a mobile device is physically located at any given time is fast becoming an intrinsic feature of any wireless application and service.

This paper will primarily deal with an investigation into the feasibility of using TinyOS motes to build a system in incremental stages that would be able to locate tagged objects within an acceptable boundary of error. The paper will be divided as follows. Section 1 will give a brief outline and history of the problem. This will be followed by Section 2 which will give an overview of the experiments performed and results obtained. Each proceeding experiment will attempt to build on the preceeding one. Limitations, advantages and disadvantages will be detailed carefully. Section 3 will finally dwell on a possible implementation of a location and tracking system.

Section 1: Introduction

a. Triangulation :

Before going on into the details of the systems, some definitions are necessary.
Positioning: positioning is the ability of the system to determine on a 2D Cartesian plane where the tagged object lies. There are many methods available to do this, the most popular being triangulation [1]. The triangulation location sensing technique uses the geometric properties of triangles to compute object location. Triangulation is divisible into subcategories of lateration which uses distance measurements, and angulation using primarily angle or bearing measurements.
Tracking: tracking is basically an extension of positioning in which case the tagged object is no longer stationery, but mobile and the system tracks the position of the object as it moves along a plane.
TinyOS: [2] is a micro threaded, event driven, network sensor which comes with a wide array of sensors. There are currently two flavors of these sensors Wec and Rene. In this research we use the Rene motes. The Rene motes have an expansion board and can sense and record the signal strength of received packets.
The following is a brief overview of the motes.
The motes are state based with levels of abstraction that separate the hardware from a clean user-interface. Commands and event handles transitions between each hierarchy thus making it easier to program and move from one state to another while maintaining a quick, low over-head.
The sensors communicate with a PC through a programming board. Their OS is installed and updated through the parallel port and data acquisition is forwarded to the serial port.

b. Related Work:-

There are a number of approaches that are in use today to determine location. For outdoor purposes, GPS is probably the best known technology. Using a satellite based system it is able to establish location on earth with an accuracy of about a 100m [4].
For indoor systems, finer grain technologies are needed and there are a wide variety available. One of the most successful location identification techniques is the Active Badges system. This system provides a means of locating individuals within a building by their Active Badges [5]. The Active Badge is a device that sends infra red signals every 10 seconds. Each office in the building is equipped with a network sensor that detected these signals and then sent that relevant information to a central network controller which polled the data and was able to discern where the personnel was located. Although successful on many levels there were drawbacks in this system. The system had no concept of distance. The system only resolved questions like “Is X in the kitchen ?”. Moreover due to the usage of infra - red signals the system was highly prone to interference and only supported a one-way communication.
BATS [3] is another location system developed at Cambridge that uses a combination of ultrasound and Infra-red that is based on the principle of trilateration.. In this system a short pulse of ultrasound is emitted from a transmitter (a Bat) and receivers in the ceiling measure the time of flight of the pulses. Given the speed of sound in the air and

information from at least three receiver it is possible to accurately determine a 3D position as well of the transmitter.
Although quite accurate a multitude of sensors are required to support the system creating a large overhead cost for this particular system.

There are other commercial applications that are available that use RF tagging technology for commercial purposes. Aetherwire [6] is a forerunner in this field using ultra-wide band solutions that are capable of centimeter accuracy over kilometer distances using tiny low powered devices to make up their particular system.

c. Outline and concepts:-

The location sensing system that I set out to build is conceptually simple even though the algorithm is non-trivial. The system that I envision will be able to establish position of a tagged object using the method of triangulation. Figure 1, illustrates the basics of this concept.

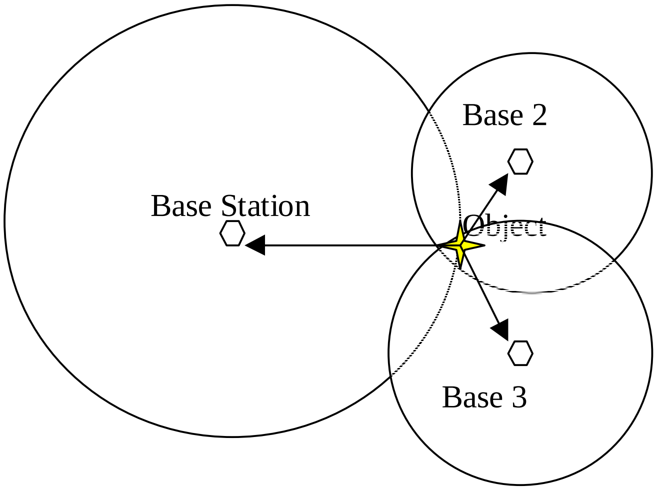

Figure 1: Location Sensing Concept of Triangulation

Basically triangulation works out as follows. An object A sends out a RF signal. Three base stations pick up the pick up the signal strengths and use calibrated data to infer approximate distance based on signal strength. A central polling server then aggregates the values to triangulate the precise position of the tagged object. Finally the computed object positions are published to client applications.

The following is an outline in which I will incrementally build the framework around my location positioning system.

1. To calibrate the Motes by varying the distance and taking in the respective signal strength.

2. Using two motes, try and make a contour map of signal strengths to try and establish the shape of the map
3. Using four motes with the receiving antennas focused in North, South, West and East direction, try and discern in which of the four quadrants the transmitting object is located.
4. Using three base stations and a tagged object obtain signal strength data from the object perform aggregation and use the corresponding values to build a possible set of distance locations. Finally from the resulting set deduce a location that would be a good approximation of the objects actual distance.

Section 2: Experiments

Experiment 1 :- Calibration of motes

a. *To calibrate the Motes by varying the distance and taking in the respective signal strength.*

Setting up the environment:
The test environment was a small room indoors on a flat table. Two motes were used , one kept as the receiver and the other as a transmitter. The motes were placed in a manner such that the antennas have a clear, horizontal line of sight across the plane. Obstructions and noise interference were minimized .
The signal strength was set to 10 in Main.c (located in the nest/tools/platform/mica folder)
The range can be set between 0 – 100 with 0 being the highest transmission power. The following is part of the Main.c code which can be altered to change transmission range

```
int main() {
   /* reset the ports, and set the directions */
   SET_PIN_DIRECTIONS();
   TOS_CALL_COMMAND(MAIN_SUB_POT_INIT)(10); //range is set here. It is currently kept at 10
   TOS_sched_init();

   TOS_CALL_COMMAND(MAIN_SUB_INIT)();
   TOS_CALL_COMMAND(MAIN_SUB_START)();
   dbg(DBG_BOOT,("mote initialized.\n"));

   while(1){
         while(!TOS_schedule_task()) { };
         sbi(MCUCR, SE);
         asm volatile ("sleep" ::);
      asm volatile ("nop" ::);
      asm volatile ("nop" ::);
   }
}
```

Transmission:

I used the following set of tools and programs to help me acquire the following sets of data

Transmission: Oscilloscope_RF

1. the oscilloscope_RF applications is located in the ~/nest/apps/oscilloscope_RF folder. A small modification is required in this part of the code. Because the initial clock setting of (128,3) is a little to fast to get the required data, I instead set it to 50 so that it would be easier to determine the incoming data packets on the receiving end of the setup.

Explanation: The default oscilloscope_RF program is used in conjunction with the light sensor. It basically is used to send light intensity data over the radio signal to a receiving mote. In this experiment I primarily use it to send arbitrary data over regular clock intervals so that location can be discerned.

Receiving:

Receiving:- generic_base_high_speed

Data that is sent from the oscilloscope node will be picked up by the mote programmed with a modified generic_base_high_speed program. The piece of the program to be modified is in the following piece of code.
Essentially the data packet coming in is a struct of the form MSG_VAL described below

```
#define TOS_MsgPtr struct MSG_VALS*

struct MSG_VALS{
short addr;
char type;
unsigned char group;
char data[DATA_LENGTH];
short crc;
short strength;
};
```

For this experiment we are interested in the strength field of the data packet which is highlighted above for easy reference.

```
TOS_MsgPtr TOS_EVENT(GENERIC_BASE_RX_PACKET)(TOS_MsgPtr data){
	TOS_MsgPtr tmp = data;

	dbg(DBG_USR1, ("GENERIC_BASE received packet\n"));
```

```
//if(VAR(send_pending) == 0 && (data->group == (LOCAL_GROUP & 0xff))){
if(1 == 1){
        tmp = VAR(msg);
        VAR(msg) = data;
        data->addr = TOS_UART_ADDR;
        data->data[3] = data->strength & 0xff; //low bits here
        data->data[2] = (data->strength >> 8) & 0xff; //top high bits here
        dbg(DBG_USR1, ("GENERIC_BASE forwarding packet\n"));
        TOS_COMMAND(GENERIC_BASE_FLASH_RX)();

if(TOS_COMMAND(GENERIC_BASE_SUB_UART_TX_PACKET)(data)){
                dbg(DBG_USR1, ("GENERIC_BASE send pending\n"));
```

The data packet coming in thus has the strength of the incoming radio signal encapsulated in it. This information is forwarded onto the by placing it in the second and third index of the data packet array before forwarding it to the UART. The UART is in turn connected to the serial port of the computer which is sent to client applications to inspect the incoming data.

Because the data array is of type char(8 bits in size) and the type of the strength field is short (16 bits in size), it is necessary to make use of 2 index char slots.

Basically the low bytes of the data is place in index 3 and the high bytes are placed in index by usage of the shift operator.

<u>Analysing the data:-</u>

Distance(cm)	Average Strength	Standard Deviation
0	656.5	1.649915823
5	630.3	4.164666186
10	622.2	1.619327707
15	624.5	3.308238874
20	612.3	1.82878223
25	601.2	2.043961296
30	600.4	2.319003617
40	568.1	3.21421553
50	557.3	4.46582183
65	533.6	2.983286778

Figure 2: Empirical results

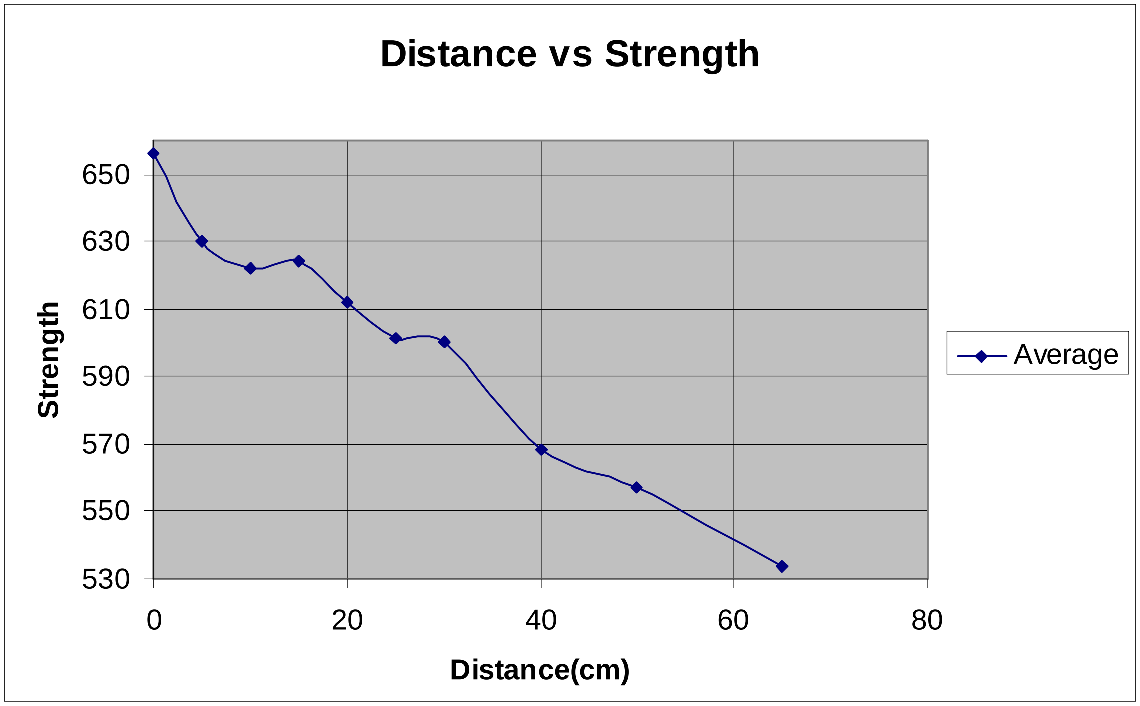

Figure 3: Distance vs Strength curve

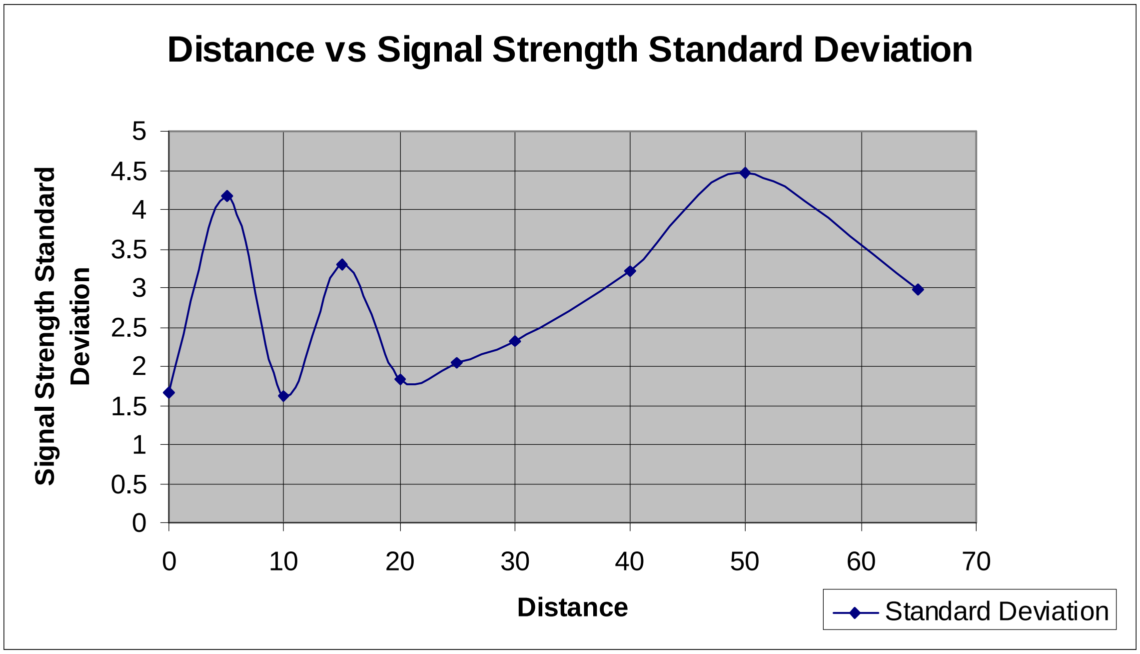

Figure 4: Distance vs Standard Deviation curve

Figure 2 shows a calibration of distance vs signal strength average. The results show a general deprecation in radio signal strength with increasing distance. Figure 3 shows that there seems to be a roughly linear regression manner to the graph. Although not noted in the graph it was observed that the signal strength kept fluctuating dramatically as seen in the 10-15 cm range where there signal strength increases minutely from 622 -624 and in the 25-30 cm range where the signal strength almost remains constant from 600-601. It would seem that even though great care was taken to minimize the amount of noise in the environment, the graphs indicate that there was indeed a lot of noise and byte corruption.

Perhaps a good indicator of the level the amount of noise played into the equation, I should have checked the connectivity of the motes, that is, checked of the number of packets being sent, how many were actually received by the transmitter. A distance vs counted packet loss would have given me a better understanding of how much connectivity played into the equation. Figure 4 shows the standard deviation of the readings against the distance. In all 20 readings were taken to get an average. The standard deviation shows a great fluctuation within the set of readings taken. It is interesting to note that at smaller distances the standard deviation is quite high while it increases almost steadily for larger distances before falling again. Where standard deviation is high indicates where there was considerable noise in the readings. This is especially true at smaller distances, which is surprising as one would expect the standard deviation to be consistent at these points. Standard deviation increases as we increase distance probably because connectivity drops between the motes so that the receiving motes get “bursts” of packets in short intervals of time thus causing irregularities in the reading.

Even with the almost linear regression it would still be hard to create a system just by matching signal strengths with the appropriate distance because of the erratic nature of the RF signal. In the paper “An empirical analysis of TinyOS networking” [7] graduate students performing similar experiments used the following system to create a a good working model. Essentially they designed an algorithm that used Kalman [8] filtering on signal strength data and aggregated the distance measurements using a mass-spring model. A Kalman filter [7] (which is not unlike a Naïve Bayesian Model) stores a belief about a position in a theoretical model. With each time step it updates its estimate of the incoming data. If the incoming data fits in well with the predicted model they are weighed highly. Poorly predicted points are weighed minimally and are deemed inaccurate. These techniques are used in current VR trackers, Information Retrieval Search Engines and Automatic Categorization of Text.

Experiment 2 :- Making a contour map

b. *Using two motes, try and make a contour map of signal strengths to try and establish the shape of the map*

<u>Outline:</u>

Experiment 2 is an extension of Experiment 1. In this experiment we place the motes at different locations on the plane instead of just within a fixed line and try to draw a line connecting all equivalent signal strengths.

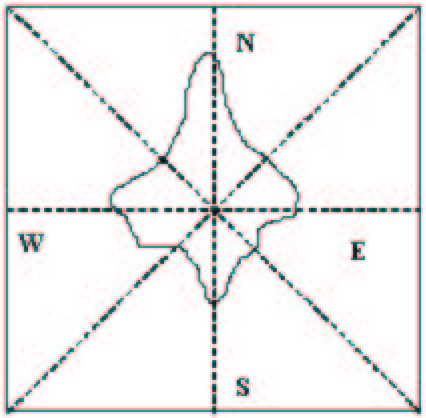

Figure 5: Rough contour map for signal strength 575

Figure 5 shows my results for the contour map. For triangulation to work (Figure 1) we assumed that the contours for the radio signal strength would be roughly circular in nature. However the experiment shows that the contours have a roughly “amoebaic” like or are irregular in shape. This is probably due to the fact that the radio signals do no and are not received from a point source but seem to be directed. This is illustrated by the fact that when the transmitting mote was place roughly north of the receiving mote, it received the strongest signal strength. When the same mote was placed roughly north west of the receiving mote (such that it was placed at an angle of 45 degrees from the orientation of the receiving mote and assuming North to be the 0th degree of reference) the signal strength dropped sharply. It was also observed (not shown in Figure 5) that for a given linear direction on the map, the same signal strength occurred on two different points on the line. This brings a new dimension to the triangulation equation as my original assumption was that there was only a one to one mapping between signal strength and distance. However this is not the case and that would mean more than one contour map could be drawn for a fixed signal strength. This would make calibration of the motes difficult as at some point a mote would have to “guess” which of the distances it is presented given a known signal strength.

A brute force method would be to just randomly guess what that distance would be. Thus given a set of distances the probability of choosing the right distance would be 1 in *n* where *n* is the number of distances given for that signal strength. However as mentioned previously in this report a Kalman filter could be made to use additional information of standard deviation and packet loss to improve the mote's guess. For example if the standard deviation and packet loss is high, then the system could bias the probability to choose a distance more towards the longer end of the spectrum.

Experiment 3:- Dividing the contour map into quadrants

c. *Using four motes with the receiving antennas focused in North, South, West and East directions, try and discern in which of the four quadrants the transmitting object is located.*

Outline:

In this experiment, four receiving motes were placed in such a manner, that the orientation of the receiving antennas were roughly perpendicular to each other. It was hoped that the additional information that the extra motes provided would enable the system to better predict the direction and distance the tagged mote was from the base station. There were many difficulties that were faced in the making of this experiment. Because the serial port on the PC is a limited resource it did not seem feasible find a PC with enough serial ports to support the proposed system.

The solution to this dilemma was quite straightforward. The motes come with 8Kb of program memory and 512 bytes of data memory. Consequently it was seen that the data memory could be used to act as a buffer and store readings from the tagged mote. This required a small modification to the existing generic_base code. Instead of sending the raw signal strength data gathered in the field straight to the serial port, an array of the packets stored 20 or so readings on the mote. After a threshold was reached, a flag was set. In another part of the code, a clock event was initiated that called a function at regular clock intervals. When the flag was set, the messages stored in the array were redirected to the serial port. The code was structured in a manner so as to make the messages broadcasted cyclic. Each message was also incorporated with a reading number so that the data processing program on the PC would be able to discern some semblance of order with the readings.

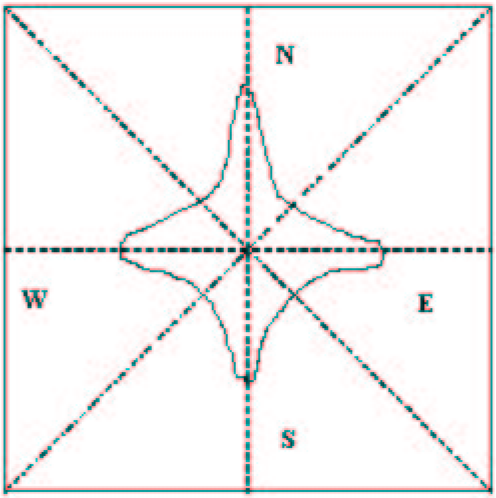

Figure 6: Contour map using 4 motes for base stations for signal strength 580

Figure 6 shows the results of the experiment. The contours were made using the following algorithm. Of the four readings, chose the two maximum readings and get their corresponding directions from the motes that shot emitted the two readings. This already gives you a general direction (which of the four quadrants) that the tagged object lies. Based on observation that if the top two readings are within 10 arbitrary units of each other, it usually means that the tag is located in one of those quadrants and not on N,S,W,E lines of the map, the map can be further divided into octants, thereby increasing the accuracy of the system.

Although this system is a better improvement on the earlier system of just having a single mote, new overheads are introduced.

- 4 new motes are required, which means the physical and processing overhead is more.
- The modified program of generic_base which allows readings to be stored directly on the chip means that more component object files have to be stored on the chip along with memory to store the readings.
- Although I have not implemented the algorithm above, conceivably it should be easy to produce. This would however more complexity to the system.

There are considerable limitations to this system though. Increased accuracy means adding more motes to the system. Such an approach is not feasible especially since the tradeoff between increased accuracy and increased amount of hardware does not give practical returns. For example to detect a tagged object on a 2 dimensional plane 3 base stations would be required. Each of these three base stations would require 12 motes to make an operable triangulation system. Also the contours are now more ‘starfish’ like in shape. This added symmetry makes it easier for one to define mathematical geometrical equations that would determine location based on triangulation. However there is a serious drawback to this system. Since each of the readings are stored on the mote itself it can only take a small sampling at a time. In the most primitive scenario after all samplings were taken, then each of the motes would have to be connected to the serial port where the data would be polled and aggregated before a reliable determination of location could be made. The increased number of motes still do not take care of the many to one relationship between distance and signal strength. It was observed at many points on the same linear axis how different distances from the receiving mote had the same corresponding signal strength value.

Section 3: Building a location sensing system

Using three base stations and a tagged object obtain signal strength data from the object perform aggregation and use the corresponding values to build a possible set of distance locations. Finally from the resulting set deduce a location that would be a good approximation of the objects actual distance.

One way to improve the system would be to give each of the four motes making a group a group id and each individual mode a sense of it's direction. After a certain, fixed sampling of data is taken, the data is forwarded to a base mote connected to the PC. Because each packet of data containing the data also incorporates the group id and line of orientation of the mote as well as signal strength, the PC on receiving all this information is able to discriminate between data received and systematically build up a topology of where the mote is located. Because of the low connectivity of the mote with increasing distance the motes and base station would have to be in close proximity of each other to make this system effective. Figure 7 illustrates this concept.

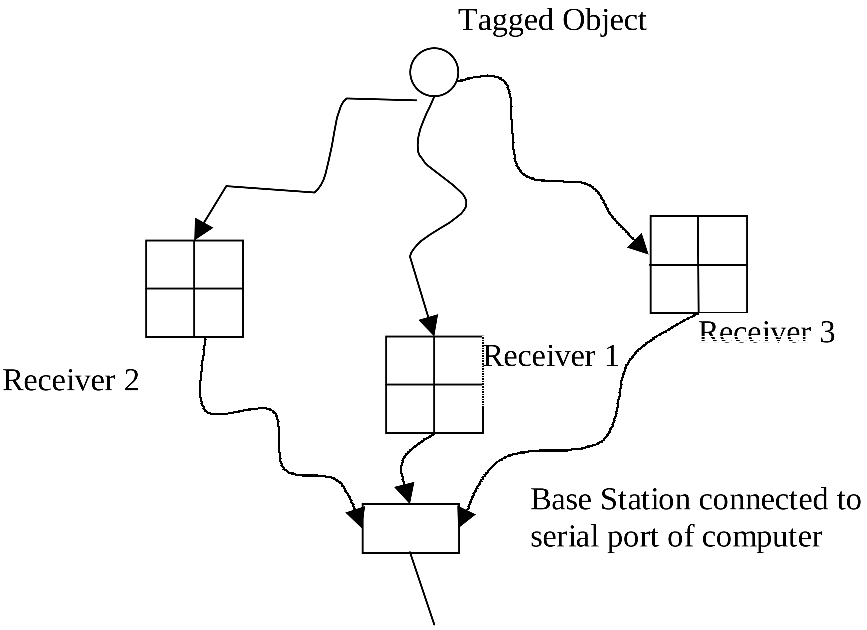

Figure 7: Building up a triangulation system

The base station mote will receive 12 different sets of data from each of the four motes from each group. It will then assimilate this data and figure out based on a quick, internal comparison of the four motes the general direction of the tagged object should be determined.

Given the relative positions of each of the base stations using a Cartesian coordinates on the plane a mathematical equation can be used to express the shape of the contour given signal strengths. For example if the signal strength corresponded to a unique distance (Figure 3), a circular contour could be drawn signifying areas on the Cartesian plane where the mote could lie.
The equation is given by :-

$$(x + a)^2 + (y + b)^2 = r^2$$

where x and y are relative Cartesian coordinates of the base system which is used as the reference point. *a* and *b* are the negated displacement distances from the reference point. Since we now have described the resultant circles a solution for intersecting points can be found. If circle A and B intersect they yield a pair of intersecting points. If circle is introduced and intersects both A and B, then 3 pairs (6 points) of intersecting points are produced. Ideally there should be an intersecting Cartesian coordinate that is common in all three pairs and this would be the exact coordinate where the tagged object lies on the plane. However since inaccuracies in the system would probably not generate a common point, an alternative method would be to compose a triangle of the three points that determine the boundaries of the intersected areas of all three triangles and the tagged object would lie within the area of that triangle. Although I have mentioned three circles, the system would produce more than one contour map for a given signal strength. The system would have to choose based on standard deviation of the readings and other relevant external data which of the contour maps is the most likely to have the tagged object located on its set of boundary points.

a. System Implications and Future Improvements:-

The system described here only provides a small framework in which a location sensing can be designed. Further improvements on it can be made quite easily. For instance since the contours of the mote map show that signal strength is strongest when the antennas of the motes are aligned parallel to each other. This would give you a directed line on which the tagged object lies. Given another mote or a couple of motes, similar lines could be drawn across the plan and a set of intersection points would be produced. Finding the area on the map where most of the lines intersect would yield an approximate location of the tagged object.

If this location sensing model can be improved and implemented to within a high degree of accuracy, the next obvious step would be to use it to track, mobile tagged objects. For this feedback would sent to the serial port would have to be processed really quickly. At this point using the existing triangulation system would not make this a very feasible. While more motes improve the accuracy of the prediction they introduce more complexity too in the systems and require a larger processing overhead. Furthermore it was noticed that when the system was disrupted either by a sudden movement, obstruction or interference it took a long time for it to reach equilibrium again, thus make it a very impractical prospect
Other improvements to the system could include research could include

- Doing triangulation with sensors at arbitrary positions and have them calibrate themselves with respect to a set of reference transmitters at known positions
- Doing triangulations with sensors that are mobile themselves. Only sensors that are close enough to the target need to participate and this has to be determined dynamically by group formation.
- Do triangulations with sensors that are "unreliable", i.e., messages may get lost etc.
- Extend triangulation to three dimensions using 4 motes (one mote to determine altitude)

b. Conclusion

In this paper I have presented a step by step build up of how a working location sensing system can be made possible using TinyOS motes. Although this system is admittedly not robust and inaccuracies are within rather unacceptable boundaries of error, never the less I feel that it is a good reflection of how the real world works. Although conceptually quite simple, external independent variables of noise, low connectivity of the motes and other environmental factors are to be taken into consideration to make a reliable working model of the problem at hand.

c. References

1. Jeffrey Hightower and Gaetano Borriella, *Location Systems for Ubiquitous Computing*,
2. TinyOS, http://webs.cs.berkeley.edu/tos/
3. BATS, http://www.uk.research.att.com/bat/
4. Cooksey, D., Understanding the Global Positioning System (GPS). 2000. http://www.montana.edu/places/gps/understd.html
5. Applications. In Proceedings of *Human Factors in Computing Systems: CHI 9*9. Pittsburgh, PA: ACM Want, R., A. Hopper, V. Falcão, and J. Gibbons, The active badge location syste*m. ACM Transactions on Information System*s, 1992. 10(1): p. 91-102.
6. Aetherwire. http://www.aetherwire.com/
7. Scott Klemmer, Sarah Waterson and Kamin Whitehouse, *An Emperical Analysis of TinyOS RF networking.. and Beyond!*, http://guir.berkeley.edu/projects/location/
8. Kalman, R.E., A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME-Journalof Basic Engineerin*g, 1960: p. 35-45.