2008

Dynamic Event Positioning using Wireless Sensor Networks

ING. MAN HUNG WONG

THESIS HOGESCHOOL ANTWERPEN

14-10-2008

HOGESCHOOL ANTWERPEN

DEPARTEMENT INDUSTRIEEL WETENSCHAPPEN ELEKTRONICA-ICT

Paardenmarkt 92, 2000 Antwerpen, België

Dynamic Event Positioning using Wireless Sensor Networks

Scriptiebank editie

Thesis voorgedragen tot het behalen van de titel
Master of Industrieel Ingenieur Elektronica-ICT.

door: Man Hung Wong
Promotor: Ing. J. Doggen

Academiejaar 2007-2008

VOORWOORD

De volledige thesis van "Dynamic Event Positioning using Wireless Sensor Networks" bestaat uit verschillende onderdelen die samen op een CD-ROM wordt geplaatst. Zo kunt u de volgende zaken terugvinden: thesis contract, wetenschappelijke artikel, voortgangsverslagen, logboek, planning, resultaten van de testen, broncode van het gemaakte programma, handleidingen van het programma, PowerPoint presentatie van het project, verslag over internationaal contact, literatuurlijst, relevante websites, poster en andere relevante materialen.

Alle onderdelen van het onderzoek worden in het Engels geschreven wegens de internationalisering van onze afdeling Industrieel Ingenieur Elektronica-ICT binnen de Hogeschool Antwerpen. Dit heeft het voordeel dat de informatie kan gebruikt worden door mensen uit het buitenland.

Dynamic Event Positioning using Wireless Sensor Networks scriptiebank editie is de aangepaste versie van de volledige thesis. Hier wordt enkel de meest essentiële informatie van het onderzoek besproken namelijk de informatie uit de wetenschappelijke paper van Dynamic Event Positioning using Wireless Sensor Networks en de conclusie van de testresultaten.

Voor extra informatie, raadpleeg de website http://www.kennywong.co.cc/ of email naar manhung.wong@gmail.com .

AUTHOR

Man Hung Wong was born in Hong Kong and moved to Belgium when he was 11. He received his Professional Bachelor degree in Applied Computer Science at Karel de Grote University College, Belgium in 2005. Later he studied Electrical Engineering and received his Academic Bachelor degree in Applied Engineering: Electronics-ICT at University College of Antwerp, Belgium in 2007 and got the title Master Electrical Engineering in Applied Engineering: Electronics-ICT at University College of Antwerp, Belgium in 2008.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the support of the following people for verifying the text. They add more professional value.

Justine Venneman and Joris Onghena.

ABSTRACT

One of the main purposes of Wireless Sensor Networks is monitoring. A simple monitoring system consists of a network with wireless sensors which can gather information from the monitored area and which can report problems by unusual events. A more complicated system is a tracking system. Besides, this one can track the moving mote. And a more advanced system is equipped with a mobile mote, that we can send to the problem area for a detail analysis with possible rescue and recovery functions to solve the problems. Sending a mobile mote to the problem area is not that difficult. We just need to know the position of the problem area and the mobile mote, then the back-office operator can run the mobile mote manual via tracking of the mote on the screen. The real problem comes when we will build an entire automated system where the mobile mote goes automatically to the problem area, without human intervention. Dynamic Event Positioning is a project about how we can bring a mobile mote to a certain mote without any human intervention, without knowledge of the orientation of the mobile mote, without the help of other reference points (anchor motes) and the position of the destination mote, where the mobile mote goes does not matter. The only thing we need is a direct connection between the mobile mote and the destination mote.

SLEUTELWOORDEN

Wireless Sensor Networks, Try-Angle, Telos, RSSI, Events Positioning, Localisation.

LIJST VAN AFKORTINGEN

- AOA: Angle of Arrival
- Init: Initialisation
- LQI: Link Quality Indicator
- nesC: network embedded systems C
- OS: Operating System
- Org: Origin
- RSS: Received Signal Strength
- RSSI:Received Signal Strength Indicator
- RSSI_org: RSSI origin
- TDoA: Time Difference of Arrival
- TRA: Try-Angle
- TRA-R: Try-Angle Reverse
- WSN: Wireless Sensor Networks
- WSNDEP: Dynamic Event Positioning using Wireless Sensor Networks

INHOUDSOPGAVE

Inhoud

BIJLAGEN

4. Bijlage 4: Test en resultaat verslag 1 - Het testen WSNDEP-realtime programma in labo
5. Bijlage 5: Test en resultaat verslag 2 - Het verzamelen van meetresultaten

CHAPTER 1: INTRODUCTION

Technologies make life for us easier than ever. Not so long ago people came up with the idea of wireless, which brings new possibilities to the surface. Think about the mobile phones that give us more freedom. We can call our friend when and wherever we want. Ten years ago when we made an appointment to meet with our friends, we needed to be on time. If something unexpected happened on the way, like a delay of a train, our friends were not able to be informed. The chance that they will leave without waiting for us was probable. Another trend is minimal sizing. Nanotechnology makes it possible to develop chips in very small sizes. In the meantime, scientists are also working on development of sensors for different fields. Combining all of those techniques, we get Wireless Sensor Networks as a result. This is a wireless network consisting of small autonomous devices with sensors for environment monitoring.

Wireless Sensor network (WSN) has features that traditional wired sensor network cannot deliver. It can be deployed much faster and easier. It can also be used in places where wiring is difficult or not possible to realize. Think about places where people do not have permission to pierce into the wall like museums or locations with very high temperatures where wire can't go through. Proceeding from the economical viewpoint, people also prefer WSN above wired sensor network, because it is more cost-effective. That is why there are so many studies about Wireless Sensor Networks.

The remainder of this paper is structured as followed: First, we talk about the goals of our project at chapter 2, then the case at chapter 3 and then we give a short enumeration at chapter 4 about the materials which are needed. To know more about localisation, we opened a topic at chapter 5. In chapter 6 we introduce our own algorithm Try-Angle. Signals play an important role at the communication. Fading and shadowing of signals lead to incorrect localisation, more about this at chapter 7. Chapter 8 discusses the scarcity element "power" at WSN. Try-Angle seems simple, but there is more to it. More about the qualities of Try-Angle at chapter 9. Chapter 10 is about the worst case scenario of Try-Angle. One of the most interesting chapter is chapter 11 which talks about how can we use Try-Angle to get the localisation information of an network without anchor motes. We compare the Try-Angle also with another localisation algorithm, this one you can find in chapter 12. Two elements which are essential for the research are tests and measurements were taken, more about this in chapter 13. And finally, we sum everything up with remarks in the conclusion and give an introduction about the future work in chapter 14.

CHAPTER 2: GOALS

Dynamic Event Positioning using Wireless Sensor Networks is just as the name says, it uses Wireless Sensor Networks technology [1]. This project is focussed on the problem of how we can bring a mobile mote [2] to its destination by "point to point" connection without any human intervention. Not like trilateration [3][4] where 3 anchor motes are needed, not like the location tracking system where there is an operator needed to run the mobile mote and also the position of the destination mote does not matter. The only thing we need is a direct connection between the mobile mote and the destination mote.

In this document, we introduce a new algorithm called Try-Angle, which makes this possible. Here, we take use of the RSSI signal with the reasoning the further the two motes are removed from each other, how weaker the signal.

Note: Mobile mote is a mote which can move freely through the networks. Examples of mobile mote are "Cotsbot"[2](see fig.1) or "Mindstorm"[5].

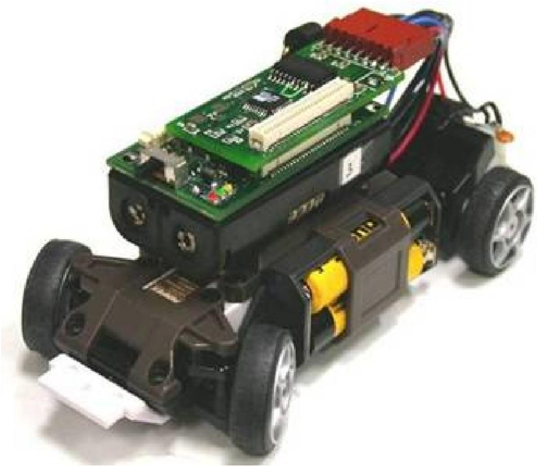


Figure 1: Cotsbot, an example of mobile mote

CHAPTER 3: CASE

A simple monitoring system consists of a network with wireless sensors which can gather information of the monitored area and which can report the problems which occur by unusual events. A more complicated system is the tracking system. Besides, this one can track the moving mote. And a more advanced system is equipped with a mobile mote [2], that we can send to the problem area for a detailed analysis with possible rescue and recovery functions to solve the problems. Sending a mobile mote to the problem area is not that difficult. We just need to know the position of the problem area and the position of the mobile mote, then the back-office operator can run the mobile mote manual via tracking of the mote on the screen. The real problem comes when we will build an entire automated system where the mobile mote goes automatically to the problem area, without human intervention. We are focussing on that last aspect and try to find a solution with our algorithm. We believe that our algorithm brings new possibilities expert on WSN and automated systems. This way, we can make an entire automated monitoring system against the (half) automated tracking system which needs human intervention. It is much better to have a system which discovers a problem, reports it and at the same time tries to work it out itself, than have a system which can only give a report and which needs an operator who has to be there 24/7 to solve upcoming problems.

CHAPTER 4: MATERIALS USED

4.1 HARDWARE

WSN focuses on the development of low-cost, low-power [6]} and multi-functional sensors in small sizes which can communicate with each other to share information. Such sensor nodes are also known as motes. Motes are like small computers, but with very limited processing and memory capabilities. They typically have the following structure: a microcontroller for processing, flash ram for data-saving, an antenna for communication, optionally with sensor(s) for measuring and a power supply unit using battery. The motes, which we have used for this project, are TelosB (see figure 2 and 3) motes from Crossbow which consist of TI MSP430 microcontroller with 10kB RAM, an integrated onboard antenna with 250 kbps High Data Rate Radio and optional integrated temperature and humidity sensor.

Figure 2: TelosB mote

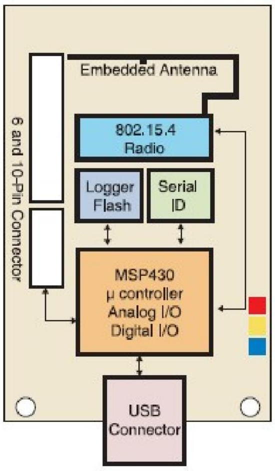


Figure 3: Structure of TelosB mote

4.2 OPERATING SYSTEM

Due to the characteristic of motes and the special requirements of sensor network applications, specific operating systems are made. TinyOS is one of the operating systems which is specifically designed for Wireless Sensor Networks. The reasons that people prefer using TinyOS is the fact that it is an open-source operating system with a component-based architecture. In addition, it has been ported to over a dozen platforms and numerous sensor boards.

4.3 PROGRAMMING LANGUAGE

The programming language that is used on TinyOS is called nesC (network embedded systems C). NesC is comparable with C and C++. The advantage of nesC is that the developer can make an application by wiring different components together [7].

CHAPTER 5: STUDY ABOUT LOCALISATION

5.1 INTRODUCTION ABOUT LOCALISATION

When we want to send a mobile mote from one place to another, we will come in contact with the topic "localisation", because we need to know the position of the mobile mote and the position of the place to go to. There are two important things to keep in mind when talking about localisation: the localisation techniques and the localisation algorithms.

5.2 LOCALISATION TECHNIQUES

5.2.1 RECEIVED SIGNAL STRENGTH (RSS)

RSS [4], [8], [9] estimates the distance between the motes by assuming a known rate of signal attenuation over distance. This is the most popular technique designed by WSN, that is because with the IEEE 802.15.4 standard, the radio receivers are bound to measure the received signal strength of arriving frames.

5.2.2 TIME DIFFERENCE OF ARRIVAL (TDOA)

Here, we measure the distance between the two motes by using the arrival time of the signal [10]. This is possible when we know the velocity of the signal. And the further the distance, how longer the signal takes to arrive. Synchronize is an important part here.

5.2.3 Angle of Arrival (AoA)

Measure the distance through the angles of arrival. This is not possible with the on-board antenna of TelosB.

5.3 Localisation algorithm

5.3.1 Trilateration

Trilateration is a method of determining the relative positions of objects using 3 or more reference points. Imagine that we have a network with 3 anchor nodes “A”, “B” and “C” (see Fig.4).

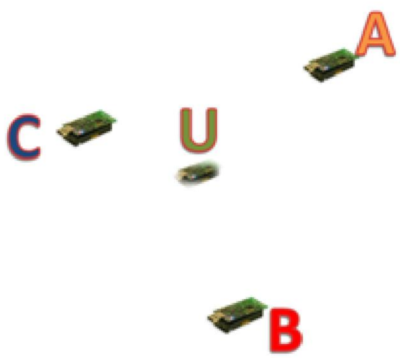


Figure 4

Anchor nodes are motes that have knowledge about their position. Mote “U(nknow)” is the mobile mote that doesn't know its position. “U” asks “A” where it is and receives the answer from “A” that it is 8 meters from “A”. We draw a circle with radius of 8 meters. “U” musts be somewhere on the circle (see figure 5).

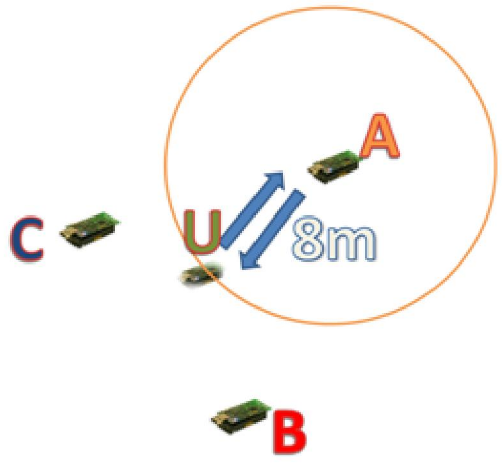


Figure 5

"U" asks "B" the same question and gets the answer that it is 3 meters from "B". We draw the circle "B" with radius of 3 meters. "U" musts be somewhere between the intersection of circle "A" and "B" (see figure 6).

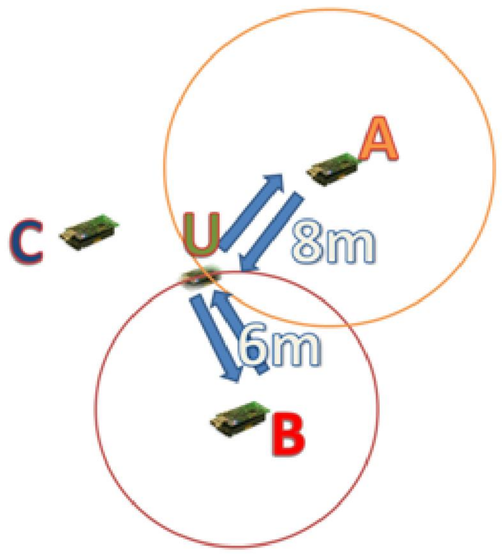


Figure 6

There are still two possibilities left. So "U" also asks "C" the same question. And get the answer it is 4 meters from "C". We draw again a circle. The interchapter of the 3 circles provides the exact location of "U" (see figure 7).

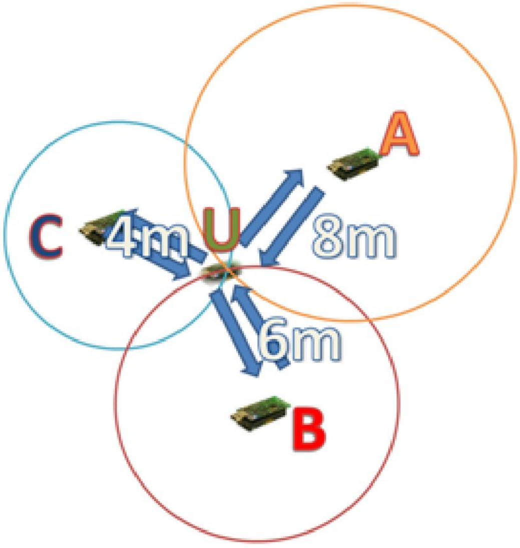


Figure 7

5.3.2 Triangulation

Triangulation is another method for localisation. It uses the measurements of radial distance, or the direction, of the received signal [10]. By measuring the relative time delays in the signal from the source set to three or more different base stations, it is possible to analytically compute the source location. Triangulation is sometimes used in cellular communications to pinpoint the geographic position of a user. However, this method is not very useful for WSN due to need of powerful antenna, which is expensive and not cost-effective.

5.3.3 Why not using Trilateration and Triangulation?

First of all we need at least three anchor motes as reference points. As a second step, trilateration and triangulation are complex calculations. To compute the position of the mobile mote requires calibration and/or calculation time and a powerful processor for synchronizing. In the last step, the orientation of the mobile mote will also cause a problem.

CHAPTER 6: TRY-ANGLE CONCEPT

6.1 BASIC IDEA

As we have mentioned before, Try-Angle (TRA) uses the method RSSI. In this part we will provide the basic idea of TRA with an example (see Fig. 8).

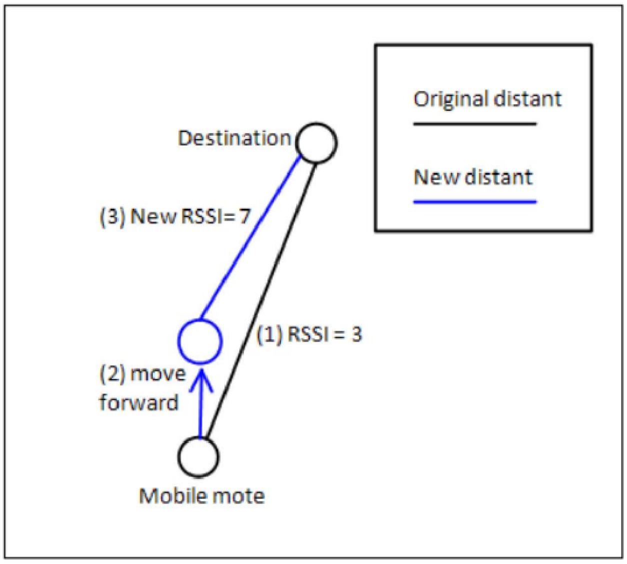


Figure 8 illustrates the begin situation.

At first the signal strength between the mobile mote and his destination using Received Signal Strength Indication [11] needs to be determined. We save this information as an original RSSI-value. Then we command the mobile mote to drive a bit ahead (recommended more than 1 metre). We measure the RSSI signal again and save it as a new RSSI-value. Next we compare the new value with the original value. If the new value is higher than the original one, the mobile mote is riding in the good direction. Because, a higher value means it has better signal strength. The signal strength increases when the distance between the mobile mote and its destination is shorter. Now we save the new RSSI-value over the old RSSI-value.

We let the mobile mote drive ahead and measure the signal strength again for a new comparison. We repeat those steps until the new RSSI-value is lower than the old one.

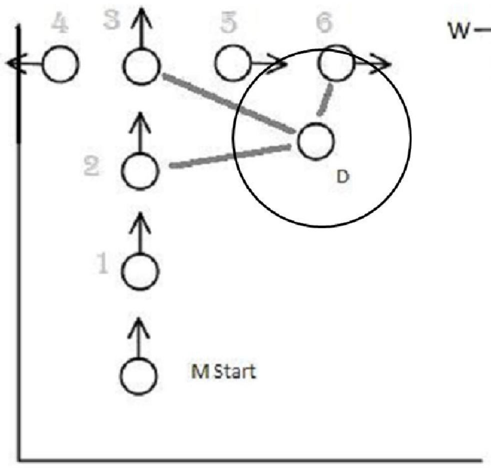


Figure 9 illustrates the whole process.

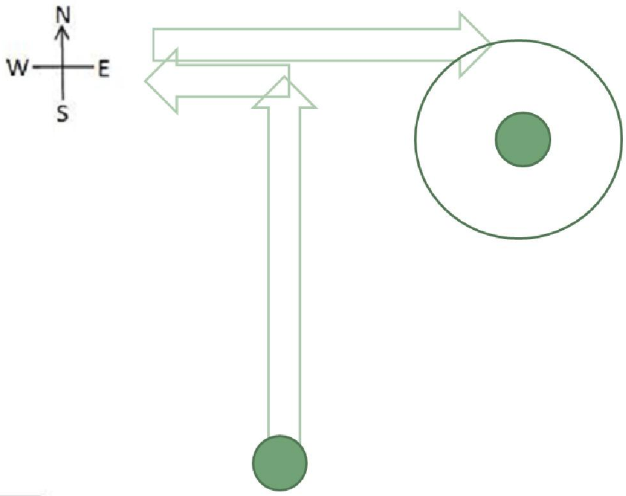


Figure 10 illustrates the road which the mobile mote had taken-off.

When this happens (step 3 by fig.9), it means the mobile mote can't go further ahead, because the signal strength will only get lower and lower. Trying another direction is needed. We let the mobile mote turn to the new direction and measure the new signal. If the new RSSI-value is higher than before, the mote may go ahead again. Otherwise, it needs to try another direction. Repeating this a few times, the mote will reach its destination (see fig. 9 and 10).

6.2 The execution: A simple idea with a solid logic as background

Now we got through the main idea of Try-Angle, we can finally get started with the real algorithm. The main idea of TRA seems simple. That is also the intention, because we want to build a simple algorithm which is quick to work with so that it is also workable with slower processors. However, there is a similar study done with logic and result.

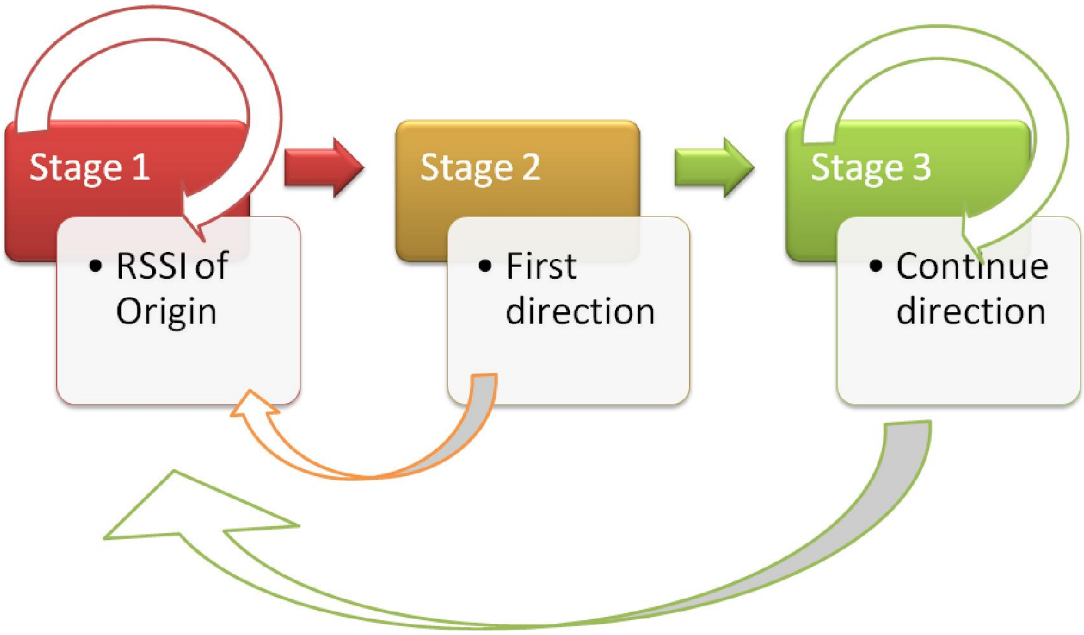


Figure 11: Try-Angle stages

Altogether there are 3 stages with Try-Angle: TRA-Stage1: Initialisation stage, TRA-Stage2: First direction, TRA-Stage3: Continued directions (see fig.11).

6.2.1 TRA-STAGE1: INITIALISATION STAGE

This is the most important stage of TRA, because all the steps which will follow are based on the information we get by this stage. At the initialisation stage, a directly connection between the mobile mote and the destination mote is made.

Pay attention: without intercession of other motes, it is a point to point connection.

Once we have connection between the mobile mote and the destination mote, we can begin collecting the RSSI values. We will sample 10 RSSI values, the average of these samples is our first RSSI value which we call "RSSI_X"(see fig.12).

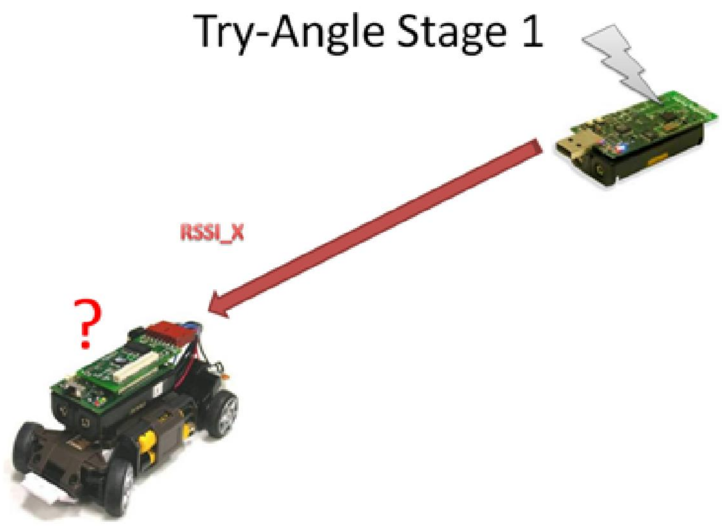


Figure 12: RSSI_X = average of 10 RSSI values

Because this step is very important, we are going to do several sampling to eliminate faults as much as possible. By this we are going to sample another 10 RSSI values for the second time and we become the result that we call "RSSI_Y" (see fig.13).

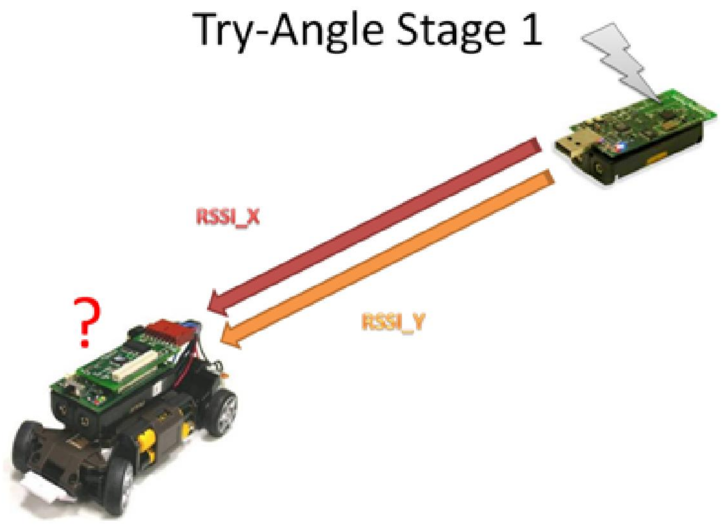


Figure 13: RSSI_Y = another average of 10 RSSI values

We compare the difference of "RSSI_X" with "RSSI_Y". When the difference is not that big (this is the threshold that you can choose, we call this "Init(Threshold)". See more in chapter 9: More about Try-Angle), we take the average of "RSSI_X" and "RSSI_Y", and this becomes also the definite RSSI value of the origin namely "RSSI_org".

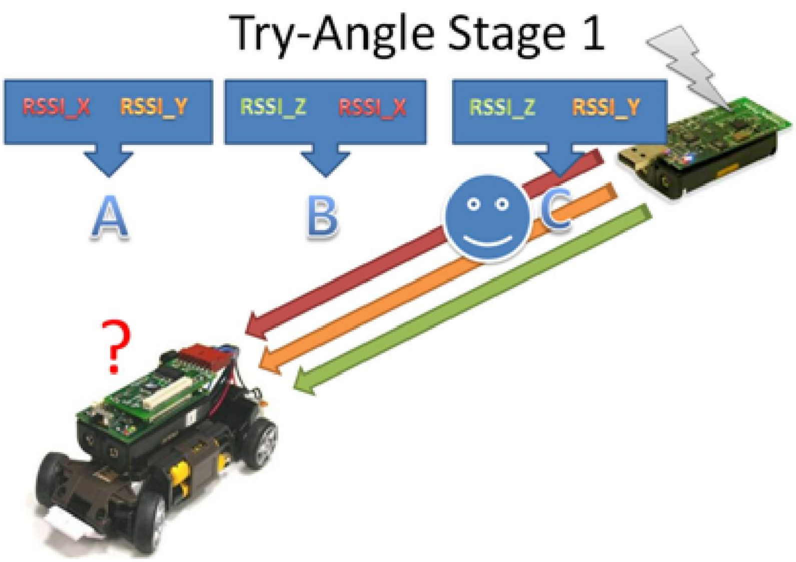


Figure 14: Stage 1 complete

When the difference between "RSSI_X" and "RSSI_Y" is big, we need to do a third measurement. We sample again 10 RSSI values, and the average of these results is "RSSI_Z" (see fig.14). We compare "RSSI_X" with "RSSI_Z", when the difference is inside our given limit, we take the average of both as "RSSI_org". When the difference is big, we compare "RSSI_Y" with "RSSI_Z" and the average of this we call "RSSI_org". But when the difference between "RSSI_Y" and "RSSI_Z" is also bigger than our given limit, it means that there is too much noise during the measurements. By this, TRA starts again the initialisation stage and with lots of hope that it will work better now. When we have completed two initialisation stages and still no good measurements are taken, there are two possibilities:

1) We take the risk and we go further with the following steps. By this we take the average of "RSSI(X, Y or Z)", the one which has the smallest difference will be our origin "RSSI_org".

2) We stop the process and we start this process again another time. The process which will start another time can give a better or more stable result. Certainly on an area that is very busy. During the peak hours there will arise more noise, than the less busy hours where are less moving objects or other obstacles.

6.2.2 TRA-STAGE2: FIRST DIRECTION

Once we have the RSSI of the origin, we can go further with the TRA-stage2. In this stage, we determine the first good direction, namely the RSSI which has a better (or higher) RSSI than the RSSI of the origin. All the directions we are testing are NOT just some random directions, but they are in a certain order to avoid superfluous measurements and to take off distances as much as possible. Altogether there are 4 possibilities of directions. The order is as follows: 1) Forwards, 2) Backwards, 3) Left and 4) Right. We shall explain it more in detail with an example(fig.15).

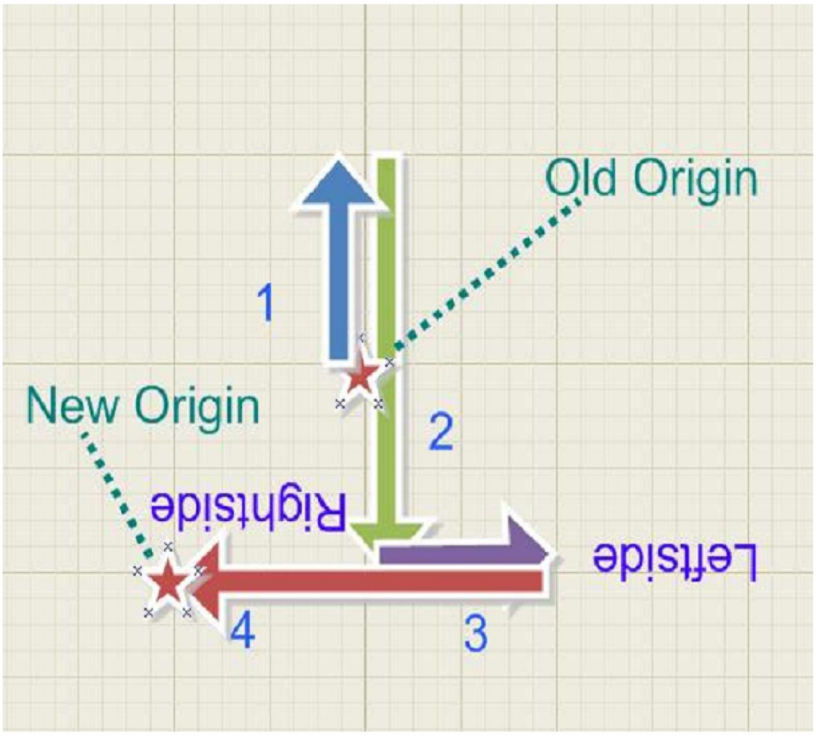


Figure 15: Stage 2 possibilities of directions

1) Step 1: Forwards
   - The mobile mote is going to walk a step forwards (every step the mobile mote takes is 1 metre. For explanation see chapter 9: More about TRA). Again, we collect 10 RSSI values from the new position. First we compare this value with the threshold value of destination to see if the mobile has reached his destination or not. If not, we will compare the obtained average "RSSI_new" with the RSSI of the origin "RSSI_org". If this new value is higher than the one of the origin, then it means we are in the good direction. A higher RSSI value means closer to our destination mote. We save the current RSSI as the old RSSI value for the next steps "RSSI_old = RSSI_new". We leave the TRA-stage2 and go to TRA-stage3. Otherwise, it means we are going in the wrong direction. It has no sense to move further forward, because the further the mobile mote moves, how larger will be the distance between both motes. In this case we go to Step 2: Backwards.

2) Step2: Backwards
   - Backwards does not mean that the mobile mote rides backwards, but he turns around and then rides forwards. Pay attention: After turning back he does not ride 1x forwards but 2x, so 2 metres. Why not moving 1 metre? The reason is simple, when he rides 1 metre the mobile mote is back to his original place (see Fig. 3 TRA-Stage2). It has no sense to measure it again to compare.
   - We measure the new RSSI value and compare it with the threshold of destination. If we don't get a good result, the algorithm will go on. We compare the new RSSI value with the RSSI value of the origin. When it is better, we save the information and leave this stage. Otherwise, we are going further with Step 3: Left.

3) Step3: Left
   - Here the mobile mote turns 90 degrees to the left and goes a step forwards. Next he measures the new RSSI value again which he compares it first with the threshold of destination.

And if he get a bad answer. The algorithm will go on to compare it with the RSSI value of the origin. In case that the obtained result is better he goes further with Stage3; else he changes to Step 4: Right.

4) Step4: Right
    - Right does not mean that the mobile mote goes to the right. But like in Step 2, he turns back and moves 2x forwards. Why do we call this right (see Fig. 3 TRA-Stage2)? If you look from the view by step2, is this the right-side. Again he measures the new RSSI and compares it first with the threshold and then with the "RSSI_org". In case the new RSSI is better than "RSSI_org", he goes to Stage3. In other case, he reports that there is a problem with the measurements. Because he already has tested all the possibilities, none of these directions gives a better RSSI value with regard to the origin. This can happen by the (temporary) noise of the environment. Try-Angle will demand go back to Stage1. But instead of using RSSI value of the original origin as reference value, we measure a new "RSSI_org" from the current (new) position. This has an advantage to avoid a similar problem with the previous measurements because TRA begins from another position. But in the case it still doesn't find a better result after the second time by Stage2, TRA reports this problem and stops the algorithm temporary (and restarts it after a time) or stops definitely.

6.2.3 TRA-STAGE3: CONTINUED DIRECTIONS

Stage3 works the same as Stage2, but there are only 3 possibilities: 1) Forwards, 2) Left, 3) Right. Backwards is not available anymore. This is because we come from the back-side. In case the mobile mote turns back and moves forwards, he moves to the path where he comes from. That is not the intention. Figure 16 makes it easier to understand.

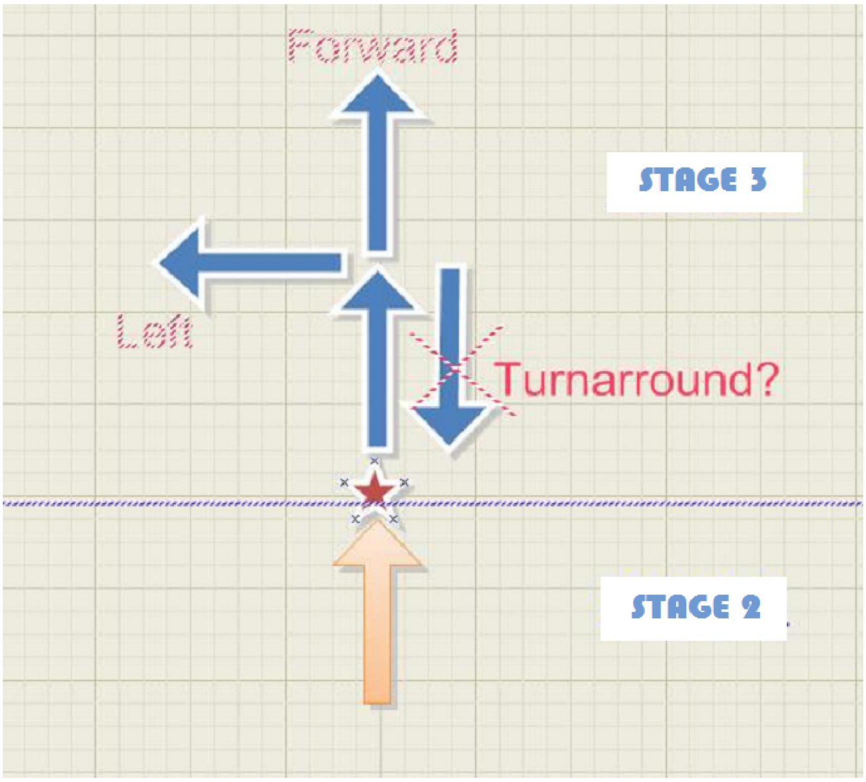


Figure 16: Stage 3 possibilities

Another difference of Stage3 with regard to Stage2 is that repetition of the steps is possible.

1) Step1: Forwards
   - Ø So we try in step 1 to move forwards until he reaches his destination or when he cannot move further forwards by the situation where the new RSSI value is weaker (or lower) than the old RSSI value ("RSSI_new < RSSI_old"). Then he goes further to step 2: Left.

2) Step2: Left
   - Ø The mobile mote runs 90 degrees to the left and moves a step forwards. Here, he measures the new RSSI value to compare. Again, it will compare first with the threshold. If it is not good, the algorithm will go further. Pay attention: If the new RSSI value is better than the previous RSSI value, it is going to the good direction. The mobile mote goes back to step 1) and moves further forwards until it is not possible anymore, and

then he changes to step 2. In other case, he goes to step 3: Right.

3) Step3: Right
   - Here the mobile turns 180 degrees and moves two steps forwards. Then, he measures the new value to compare. First with the threshold and then with the previous RSSI value. In case the new value is better, he goes further in this direction (he changes to step 1: forwards). In other case he reports a problem. Because we have tried all the possibilities, TRA goes back to Stage1 and the RSSI of current position will be the new RSSI origin.

6.2.4 TRA-STAGES SUMMARY

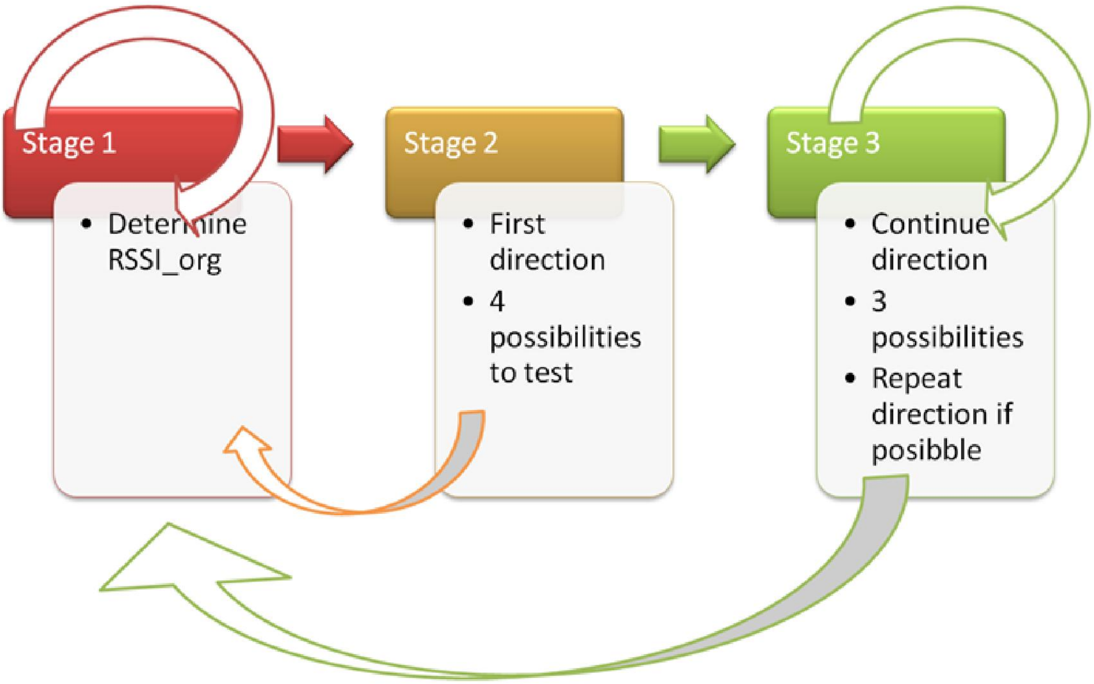


Figure 17: TRA-Stage summary

6.2.4 IMPROVEMENT OF THE ROBUSTNESS AND THE RELIABILITY: WHEN DOES THE ALGORITHM STOP?

1) Within the range of destination mote (Threshold of destination)
    - Ø The algorithm stops when the mobile is within the range of the destination mote in any kind of phase (1, 2, and 3). The range is the threshold value where the user can establish. This value will be discussed in detail later.

2) At Stage1
    - Ø Try-Angle stops completely or temporary when he cannot get out Stage1 two times after each other.

3) At Stage2
    - Ø TRA goes back to Stage1 when he tried out the four directions and got no result. To arrange that Try-Angle does not get into a loop, TRA can try Stage1 only maximum 6 times in total. In other words, after going 6 times to Stage1, TRA stops and reports a problem to the operator of the system. The operator can decide if he is going to reset the algorithm and execute it again on another time. If he stops the process entirely, he can call the mobile mote back to his original situation (via TRA-Reverse).

4) At Stage3
    - Ø Try-Angle goes back to Stage1 in case he doesn't get another good result. To occur that TRA does not get stuck, Stage1 can only tried out maximum 6 times.

6.2.5 THE ROUTE: FASTER GO BACK!

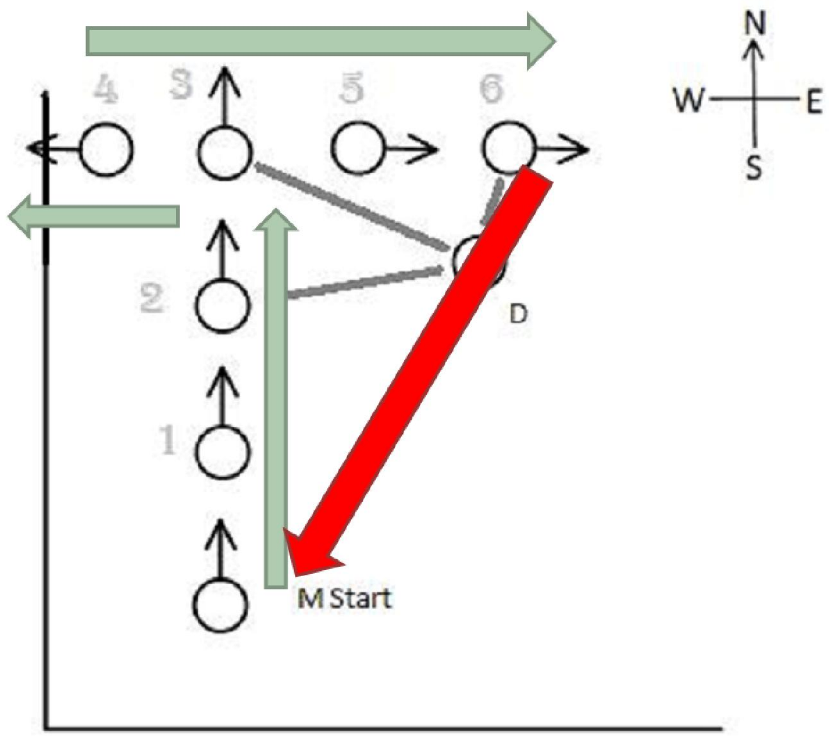


Figure 18: Try-Angle Reverse

The figure 4 illustrates a situation of TRA. The green line is the route that M has taken off. It is always the rectangle of the triangle when he turns. Therefore the name "TRYANGLE ALGORITHM (TRA)". At first glance, it seems there is not much advantage, because it has taken off a longer route. However, that is not entirely true! This is because we can use TRA without the need of 3 anchor motes for references. There is no need to know the position of the mote M, which means we don't need the time to calculate it as with "trilateration" and "triangulation". We are not concerned about the orientation of the mobile mote.

Yet, there is more, because the sides and the angle of triangle are known, the calculation for a return route is much faster see the arrow backward! We call this "Try-Angle Reverse (TRA-R)"(fig.18). More about it in chapter 11.

CHAPTER 7: STUDY OF SIGNALS

7.1 CC2420

CC2420 [12] [4] plays a very important part in this test, because this is the internal antenna which is built in the TelosB mote. CC2420 operations in 2.4 GHz ISM band with an effective data rate of 250 kbps. In total there are 16 channels (from 11to 26) with each channel occupying a 3 MHz bandwidth with a centre frequency separation of 5 MHz for adjacent channels. Besides CC2420 provides us with two useful parameters for measurements: RSSI and LQI.

7.2 RSSI

RSSI is the estimate of the signal power. It is calculated over 8 symbol periods and stored in the RSSI VAL register [13]. Below is the formula to compute the received signal power (P) in dBm:

P = RSSI(VAL) + RSSI(OFFSET), where RSSI(OFFSET) is about -45.

Note: The RSSI value we get by our program is Raw RSSI data. When we need the dBm value, we can simply calculate it to subtract the Raw Data with 45.

According to many previous tests, the Received signal strength indicator (RSSI) is known as a bad estimator of the link quality, because RSSI has for a given link, a very small variation over time. However, Kannan et al. [13], shows that RSSI can be a good estimator too in the situation when RSSI is above "-87 dBm". But CC2420 has a better parameter for link quality, namely LQI.

7.3 LQI

LQI [12] [13] [4] stands for link quality indicator. It is a new parameter of CC2420. It measures the error in the incoming modulation of successfully received packets (packets that pass CRC check).

LQI is a better estimator of the link quality, because it varies over a wider range over time for a given link. LQI can be viewed as chip error rate and is calculated over 8 bits following the start frame delimiter (SFD). The values are between 110 and 50, and correspond to maximum and minimum quality frames respectively.

Note: Single packet LQI is NOT a good indicator of intermediate links

7.4 Signal fluctuation

7.4.1 Causes

There are several causes which can cause the signals fluctuation. The instability of the signal leads to incorrect localisation determination. This problem occurs strongly with indoor environment like the office.

1) Antenna orientation
   - Ø By changing the antenna orientation between the sender and the receiver motes, there can appear a fault which varies from 1 to 3 metres. The average localisation error of one orientation can be 200 percent higher than another [14].

2) Foreground obstacles
   - Ø Tsung-Han Lin et al. [14] has done a test with four circumstances: (1) mote on chair, (2) mote held by hand, (3) human standing still next to the mote, and (4) human wandering around the mote. When we move before the mote, the average localisation error changes from "1.22 - 2.71 metres". And the fault can be even 141% big when we hold the mote in our hands. The study shows us that the localisation error increases significantly if the nearby obstacle moves.

3) Effect of background obstacles
   - Ø Background obstacle like furniture in the building, noises generated from the use of electronic devices can also lead to signal fluctuation. But the effect is rather limited. The fault is approximately "1,52m - 1,58m". This is negligible according to [14].

7.4.2 MULTIPATH FADING EN SHADOWING

We have discussed the several causes, we can now begin to discuss the fluctuation more in details. Fluctuation of the signal happens by fading (multi-path fading) and shadowing. They can be subdivided in a scale from small to big according to Daniele et al. [15].

- Ø a small-scale: the level of attenuation of the signal changes substantially if the position of the receiver or the transmitter is varied by about half a wavelength of static multipath fading
- Ø a large-scale: due to large obstacles, which create shadow zones that cause deep fades if a receiver happens to enter them of shadowing

7.4.3 Calculate Fading

Fading does not directly depend on time, and only depends on the position of the nodes and the topology of their surroundings. According to [15], fading can be calculated (in levels) if the position of the terminals and the geometry of the environment where the network is deployed, are known at all times.

The two most important techniques to calculate the fading are: Rayleigh-distributed and Ricean distribution

- Rayleigh-distributed: is used with absence of a dominating component
- Ricean distribution: by presence of a dominant static component. A typical phenomenon at a line-of-sight path.

However, according to Daniele Puccinelli et al. [15], these two models did not describe the realistic situation. They were started with a realistic model which combines the two methods. But the obtained model is complicated and a processor with a hard calculation capacity is a must. More details we refer to their paper [15].

7.4.4 Signals through the wall

Matthias et al. [16] indicates that when we let the signals go through the wall, there is a certain weakness, but that the weakness through the second wall is much smaller than the first wall.

7.4.5 Fluctuation with outdoor

Multipath fading is strongly present by indoor environment, but outdoor can also have the problems. Radio waves still get reflected of buildings and other landscape features. Another problem outdoor system can have is the weather. Benji et al. [17] indicates that the signal fade dues to weather by outdoor.

7.4.6 HOW CAN WE AVOID FLUCTUATIONS AS MUCH AS POSSIBLE?

- Using an antenna of uniform radiation pattern
- Do the test and place the mote in a way that we can use it later to avoid re-calibration.

CHAPTER 8: POWER CONSUMPTION BY WIRELESS SENSOR NETWORKS

Power is the most scarcely element at WSN. By this we have done also a study about the possible effects. When we let the motes communicate with each other, the transmission uses the most signal. This is much more than the power that the microprocessor uses. Michael et al. [18] indicates also that the reception costs may be more than the transmission costs. Many people think the cost will rise if the distance increases. But that is not the case. Because according to [18] two links of different lengths will need to transmit at the same power level, therefore the cost to transmit over different distances can be equivalent.

That is why Michael et al. [18] concludes that the transmission cost is no longer dependent on the distance between two nodes but rather the chip specifications, packet length and sending power level.

1) How far can we adjust the power at Tinyos?
    - In Tinyos, we can adjust the sending power level of the packets. Here we make use of the command CC2420Packet.setPower().

2) Effects of discrete power levels
    - Michael et al. [18] have done a test with 8 distinct power level settings. It showed that at power level 3 the mote can easily communicate with motes within 10 metres. And with power level 11, it can go easily until 20 and 25 metres.

CHAPTER 9: MORE ABOUT TRY-ANGLE

9.1 TRA PARAMETERS

9.1.1 THE EFFECT OF STEP SIZE (WHY 1 METRE?)

In the chapter about the TRA concept, we told that the step size of the mobile mote is 1 metre. Now, why is this 1 metre and not 2 metres? This has to do with the localisation technique RSSI, the exactness of TRA, the velocity of TRA, the range of the destination (threshold destination mote) and the chance on an extra side path.

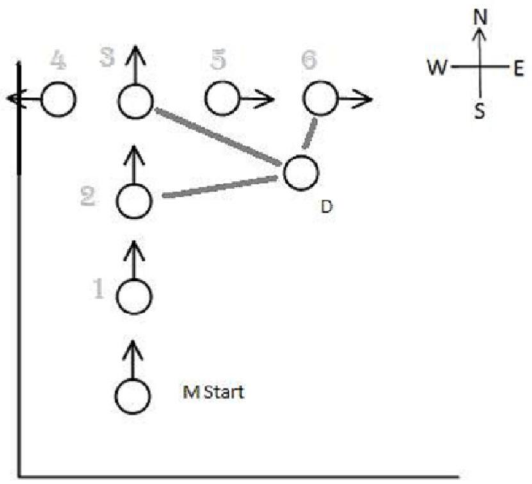


Figure 19 (A)

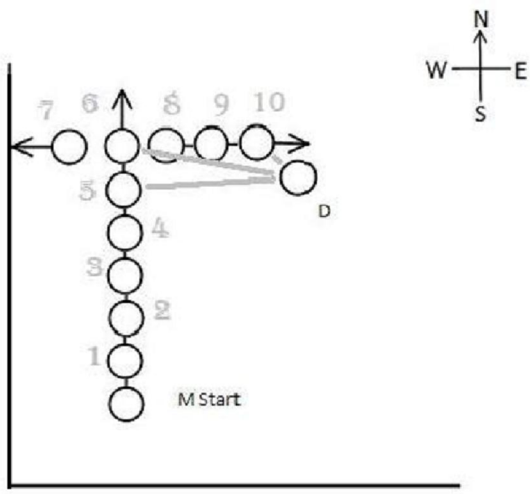


Figure 19 (B)

Here we see the mobile mote by fig. 19 (A) stops further of the destination than by fig.19 (B).

1) RSSI characteristic

- Like we said in the chapter "Study about signals", there are lot of causes to let the signals fluctuate. Especially by the indoor environment, it is a problem. By choosing the right step size (not too big, our suggestion is 1 metre due to the other parameters), the fading and shadowing can be detected faster. By this TRA can also take the required action quicker.

2) The exactness
   - We will indicate it by showing an example. The figure 5 shows the effect of the step size where the starting point of the mobile mote and the position of the destination mote on both figures are the same and the destination range is 2 metre. Fig. 19(A) illustrates the situation with the step size of 2 metre and fig. 19(B) the step size of 1 metre.

3) The velocity
   - At fig.19 (A), TRA has only stopped and measured 9 times, however at fig.19 (B). He had to do it more than once. But that is not always the case, because it can also take longer. This will be discussed at "chance on an extra side path".

4) Destination range
   - The size of the destination range has a direct influence on the size of the step. When we use a smaller destination range, the size of the step has to be smaller too. The step size has to be smaller or equal to the destination range. For example when the destination range has a radius of 1 metre, the step size per step cannot be bigger than 1 metre. This avoids the chance on an extra side path.

5) A chance on an extra side path

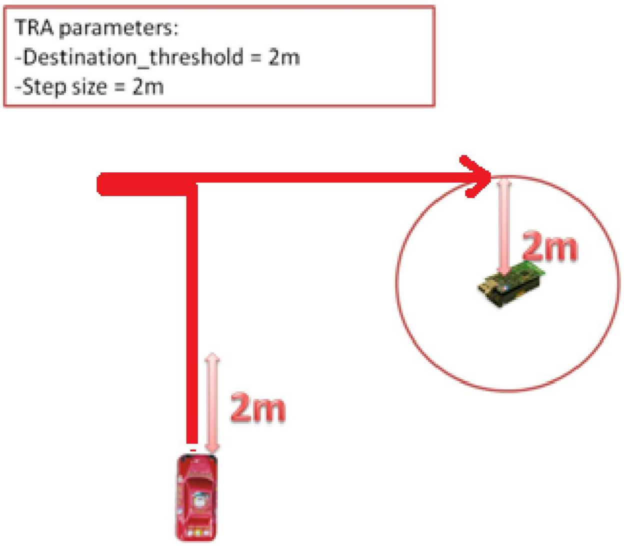


Figure 20

- When we enlarge the step size, we enlarge also the chance on an extra side path. When the destination range is changed to 1 metre, but the step size does not change (still 2metre). We see the final step(fig. 21 and 22) not close enough to the destination mote.

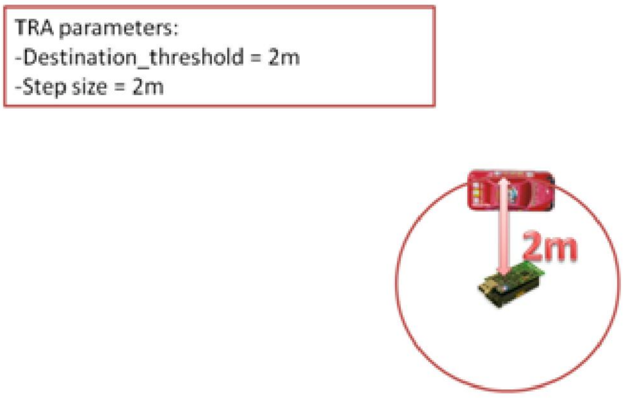


Figure 21

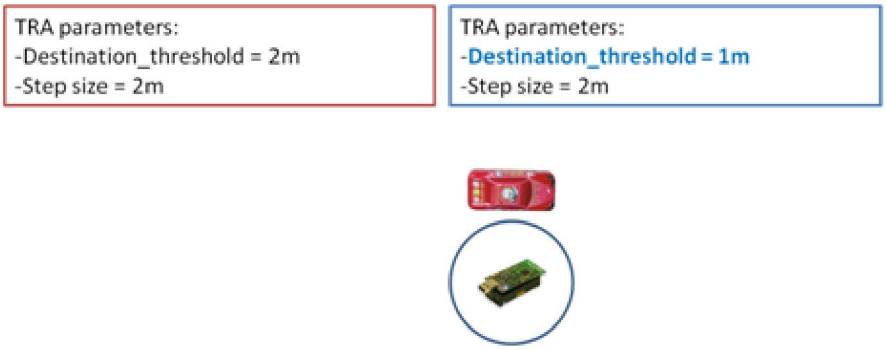


Figure 22

- Therefore the mobile mote has to move further.

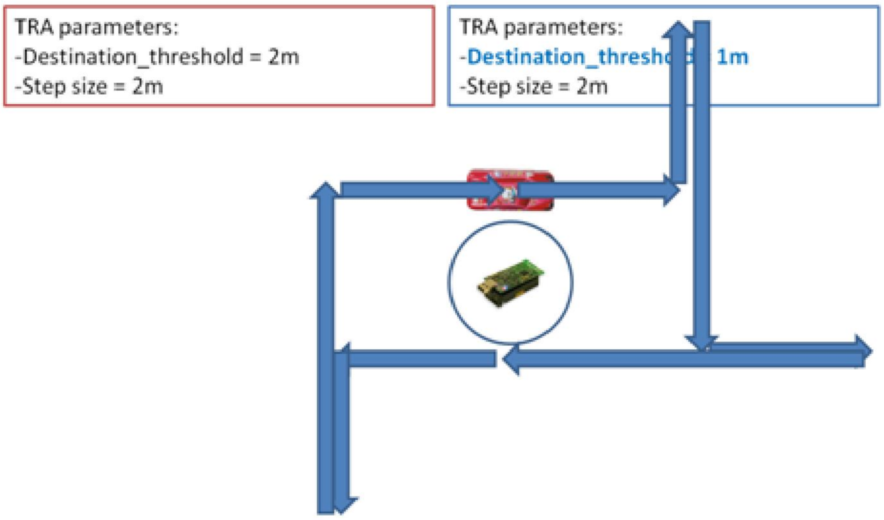


Figure 22: Out of control

9.1.2 INIT(THRESHOLD)": THE THRESHOLD BY TRA-STAGE1

1) What is Init(Threshold)?

- Ø It is the threshold we use for the exactness with TRA-Stage1 (initialisation stage). We have told that the TRA-Stage1 is very important, that is why we sample it several times. The sample value we become are "RSSI_X", "RSSI_Y" and if necessary "RSSI_Z". We have told that the difference between "RSSI_X" and "RSSI_Y" cannot be too big. So the difference between "RSSI_X" and "RSSI_Y" cannot be bigger than "Init(Threshold)".

2) What is a good value for "Init(Threshold)"?
   - Ø When the distance between the mobile mote and the destination mote is smaller or equal to 10 metres, the fluctuation of the signal is limited (this is what our measurement of 100 RSSI values said). By this we can decide that the difference between the average of the first 10 RSSI values (RSSI_X) and the average of the second 10 RSSI values (RSSI_Y) has to be in a range between [-2, 2]. To eliminate the "-" sign, the difference between "RSSI_X" and "RSSI_Y" has to be raised to a square before they can compare it with "Init(Threshold)". So we can say that "Init(Threshold) = 4" is a good Threshold value. When the distance between the two motes is bigger, we can increase the "Init(Threshold)" value.

9.1.3 THRESHOLD OF DESTINATION MOTE

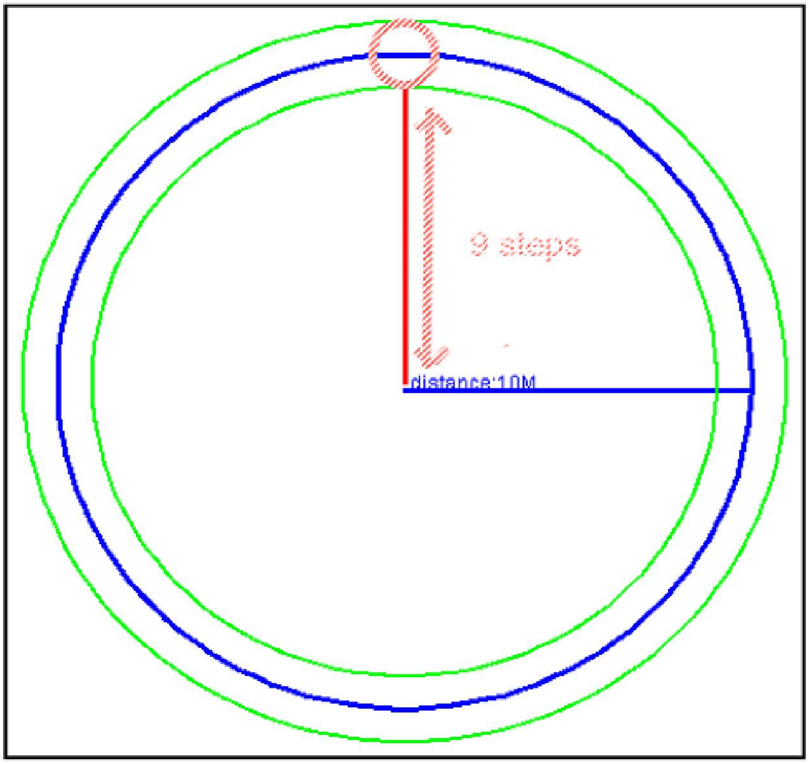
Threshold of destination mote determines how close the mobile mote has to be to the destination mote. TRA stops when the mobile is in this range.

Note: Since that the mobile mote has better equipment than the static mote (anchor mote), it is not necessary to reach the position 100 percentage exactly. As long as it is near enough to the destination mote, it will be fine for the mobile mote for further detail analysis.

CHAPTER 10: WORST CASE SCENARIO BY NORMAL NETWORK

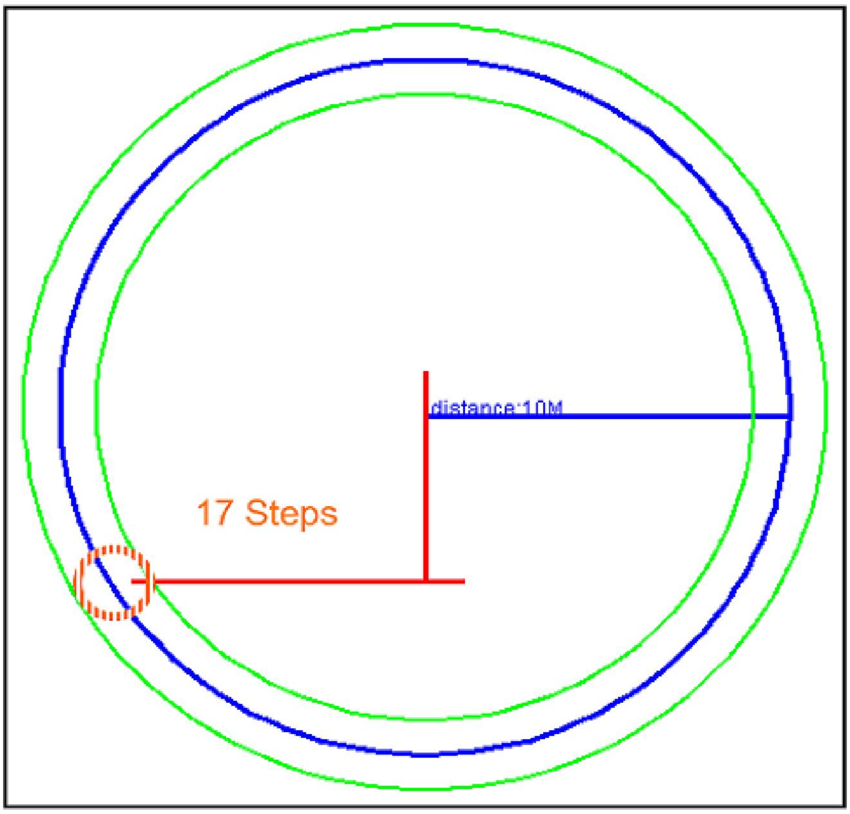
How big can the difference be between best and worst scenario at TRA? We present the following. The destination mote is 10 metres further than the mobile mote. The range of the destination mote is 1 metre. The step size is also 1 metre.

1) Best scenario:



- The best situation takes place when the destination is 0 degrees from the mobile mote (something between "-5" or "5" degrees, because the range of destination is 1 metre). The total steps is "9".

2) The worst scenario:



- The bad situation occurs when the destination mote is at "180 degrees + 45 degrees = 225 degrees" of the mobile mote clockwise. The number of steps is "17". It is 80% more than the best case. When the distance between the mobile mote and the destination mote is 5m. Best = 4 steps, Worst = 10steps: thus the worst case can up to 150% more than the best case.

10.1 TRA ERROR RATE

Just like other localisation algorithms, TRA can fail too. This is because TRA is based on the data of RSSI. When we are in an area where has lots of noises, it can lead to a very big fluctuation of RSSI signals. This occurs in indoor environment where are lots of movements (many objects which move), so that the topology varies constant (the RSSI signals varies constant). In this case TRA cannot function well, so it has to re-initialise a couple of time (max. 6 times). However this is a problem that appears at every localisation algorithm.

10.2 TRA RELIABILITY

To rise the reliability of TRA, we split up TRA in 3 stages. This has a reason. There are 4 possibilities (4 directions) by TRA-Stage2. When TRA tried all 4 possibilities and still do not get a good result, he begins at stage 1 again with a new "RSSI_org". Now, the second origin is different than the first one. This has an advantage to avoid certain problems. What happen in this situation, does not happen in the new one. Because we start from a different starting point with a different "RSSI_org".

Note: In totality, stage 1 can have 4 different origins (positions), when stage2 fails every time and go back to stage 1.

CHAPTER 11: USING TRY-ANGLE TO LOCATE THE UNKNOWN NETWORK (AUTOMATIC)

In this chapter we are going to discuss how you can do an entire determination of the location of the network, where you know nothing about it, by using only a mobile mote. An unknown network is a network that only exists of motes which doesn't know anything about their location. There are no anchor motes present to do a reference.

WE ILLUSTRATE IT WITH AN EXAMPLE.

Imagine we have a destination mote and a mobile mote.

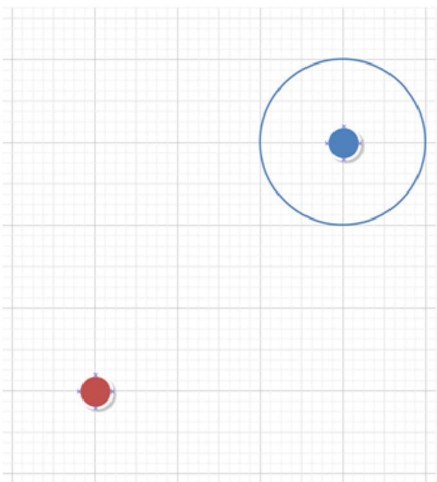
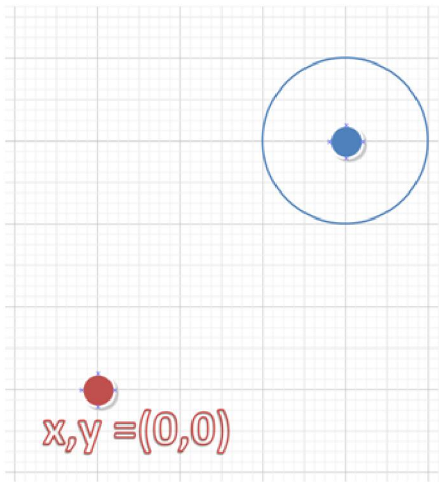


Figure A

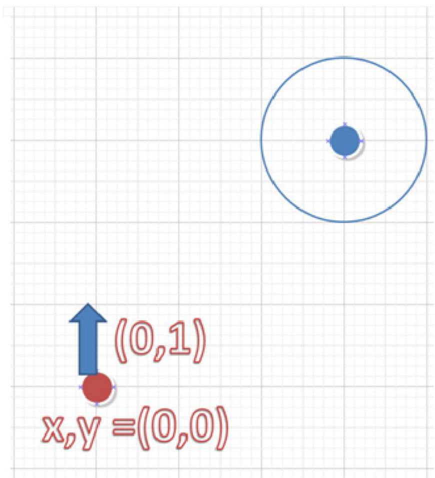


Figuur B

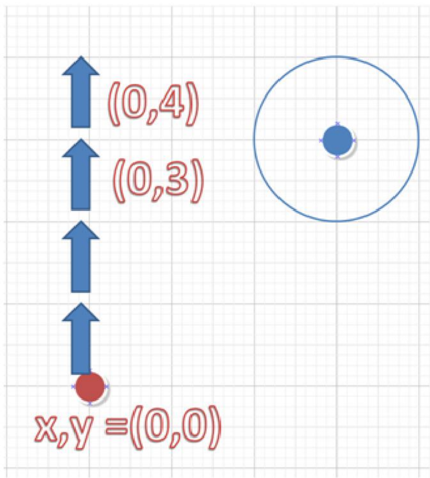
The red point is the mobile mote, the blue one is the destination mote and the blue circle is the range of the destination (fig. A). We have no idea where the location is of the mobile mote and the destination mote in our network. The only thing we have is the connection between the mobile mote and the destination mote. We want to determine the position of the destination mote with the position of the mobile mote as startingpoint and the range of the destination as endingpoint. We work out our algorithm Try-Angle. The startingpoint of our mobile mote is the origin of our system of coordinates. X refers to the X-as and Y refers to Y-as.

The mobile mote goes a step fowards, each step is 1 meter. Figure C illustrates it. The value of Y rises because the mobile mote goes up. The value of X stays unchanged. The result: X,Y = (0,1).

Note: If the mobile mote goes down, the value of Y goes down too.



FigureC



Figuur D

The mobile mote drives further untill the new RSSI-value is lower than the old one (principe of Try-Angle). Figure D illustrates it: the mobile mote drove 4 steps forwards => X,Y = (0,4).

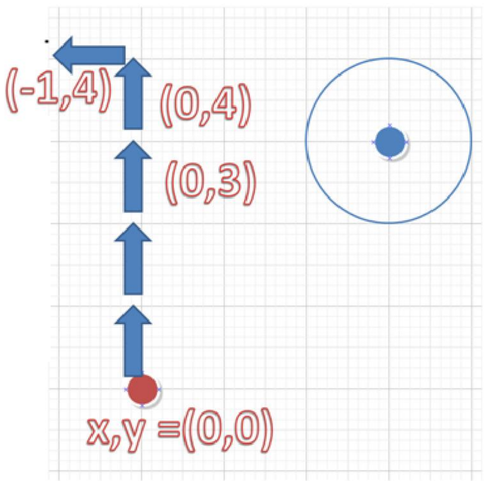
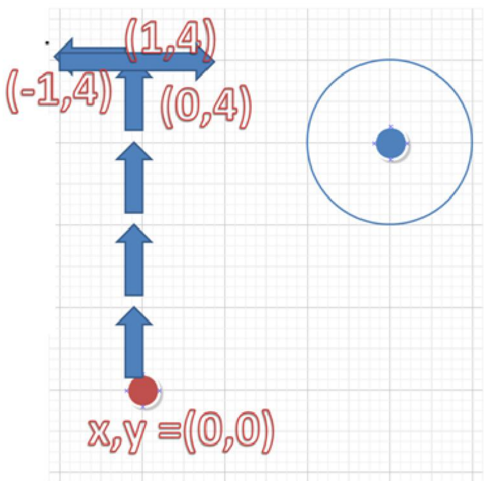


Figure E



Figuur F

Now the mobile mote has to go to another direction, according to Try-Angle, it has to go to the left. The value of X goes down => X,Y = (-1,4) (see figure E). But the new RSSI-value is worser than the old one. It means the mobile mote went to the wrong direction. It has to try another direction by turning back and moving 2 steps further => X,Y = (1,4).

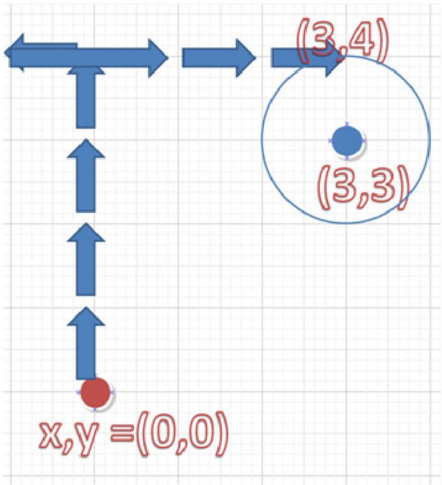
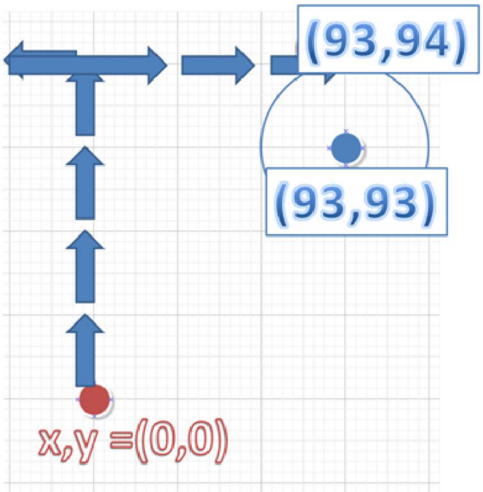


Figure G



Figuur H

The mobile mote is driving further until it has reached its destination, this is the situation where it is in the blue circle(see figure G). We have as result that the destination mote is X=3 and Y= 4 with the startingpoint of our mobile mote as the origin. But the real position of our destination mote is X=3 and Y= 3.

IS IT BAD?

- No, it is an acceptable when the mobile mote stays in the range of the destination mote(look for the reason by chapter6: Try-Angle concept).
- And if the real position of the destination is changed to X= 93, Y= 93, than we get as result: X=93, Y=94. This looks much better than before(figure. H).

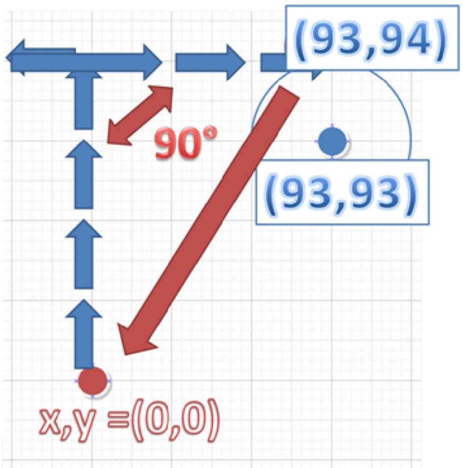


Figure I

MORE ABOUT TRY-ANGLE REVERSE

Since we can determine the X-value and the Y-value, we know that Try-Angle always turns 90 degrees. We can determine a fast way back, for the mobile mote to go back to his origin after it finished his task by drawing a tryangle. That's why we call it Tr"y"-angle (Tryangle). We can draw this by looking at the characteristics of the triangle. The only thing we have to say to the mobile mote is the amount of degrees he has to turn, next he just have to drive forwards without any roundabout. We can also determine the distance of the roundabout by using using Pythagoras

If there are other motes who are unknown in our network, we can locate them with our algorithm Try-Angle. We can do this from the point of the destination mote but also from the origin.

A possible exercise of this is the construction of a fast WSN network where we have no time to make an entire analyse of an area. So without beforehand determining what a specific mote has to install (different measurements). We can spread the mote anywhere we want in the area we want to monitor. And we let the mobile mote do his job to locate the whole network. Meanwhile the operator can go drink a coffee, when the mobile mote do his job. Can it be easier?

CHAPTER 12: TRY-ANGLE VERSUS TRILATERATION

12.1 SUMMARY

Before we end this paper, we shall compare Try-Angle with one of the most used localisation algorithm "Trilateration".

	Trilateration	*Try-Angle*
Localisation techniques	*RSSI*	*RSSI, LQI (maybe in the future)*
Requirements	*Needs at least 3 anchor motes as reference points*	*Needs no help of anchor motes, as long as he have connection with the destination mote*
Orientation mobile mote	*Must be known beforehand*	*Does not matter*
The route	*No superfluous route*	*Superfluous route is possible*
Complexity of algorithm	*Needs calibration (the reference information from the "anchor motes cross rarely through 1 point)*	*Simple algorithm, Works also with a slower processor.*
Type of WSN System	*Half-automated system (Tracking)*	*Fully Automated system*

12.2 ERROR RATE

As they both use RSSI as localisation technique they suffer the same problem namely the signal fluctuation. We cannot give the exact error rate between both algorithms. What we can conclude is that trilateration will has as much as Try-Angle when fading and shadowing take place. As soon as one of the anchor mote provides wrong information, the circles will not cross through one point.

And in the worst case: Because trilateration uses 3 reference points, when one of the three points does not work anymore (wrong information), the algorithm will fail. Even if the destination mote sends correct information. Is trilateration maybe more vulnerable than Try-Angle?

CHAPTER 13: TESTS AND RESULTS

13.1 MEASUREMENTS OF RECEIVED SIGNAL STRENGTH INDICATOR (RSSI): WSNDEP DATA COLLECTOR

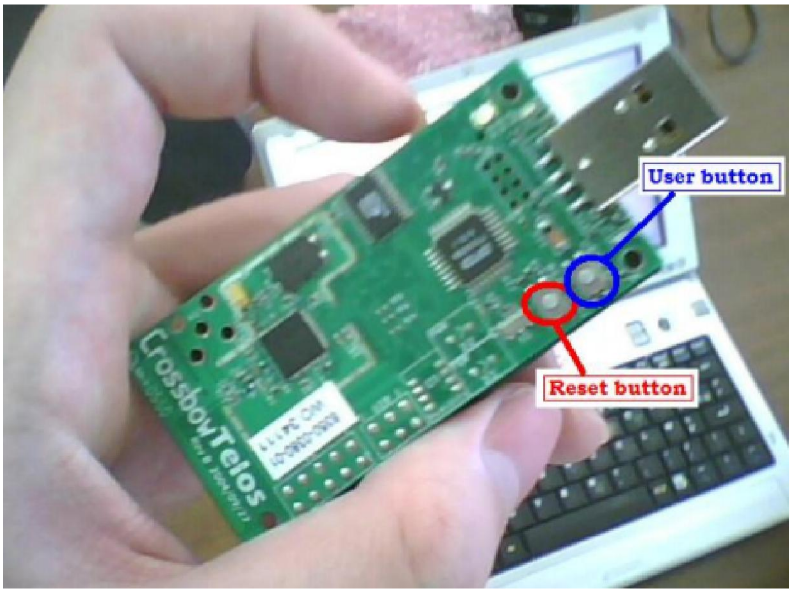
To know more about the characteristic of RSSI by TelosB, we have done several measurements at different places. To make it more pleasant, we have designed a special program "WSNDEP Data Collector" to help us collect the measurements. This program is written in nesC and works under TinyOS 2.0 environment. This program can collect in total 100 RSSI values with an interruption by each 10 RSSI values. At the end of the test, we get all the RSSI values together with the average RSSI values on screen.

The measurements will be done as followed.

Foto 1: Viewpoint of TelosB - During measurement by the parking Carrefour Merksem, Belgium

Foto 2: Viewpoint of UMPC - During measurement by the parking Carrefour Merksem, Belgium

For the measurements we need 2 (TelosB) motes. The Anchor mote has the task to constantly send packets to Blind mote. And we connect the Blind mote with the UMPC (Ultra Portable). When the Blind mote receives a packet, he calculates the RSSI value. Next, he sends the RSSI value to our UMPC, which shows the result on our screen. We lay down the Blind mote (UMPC) and the Anchor mote from each other.

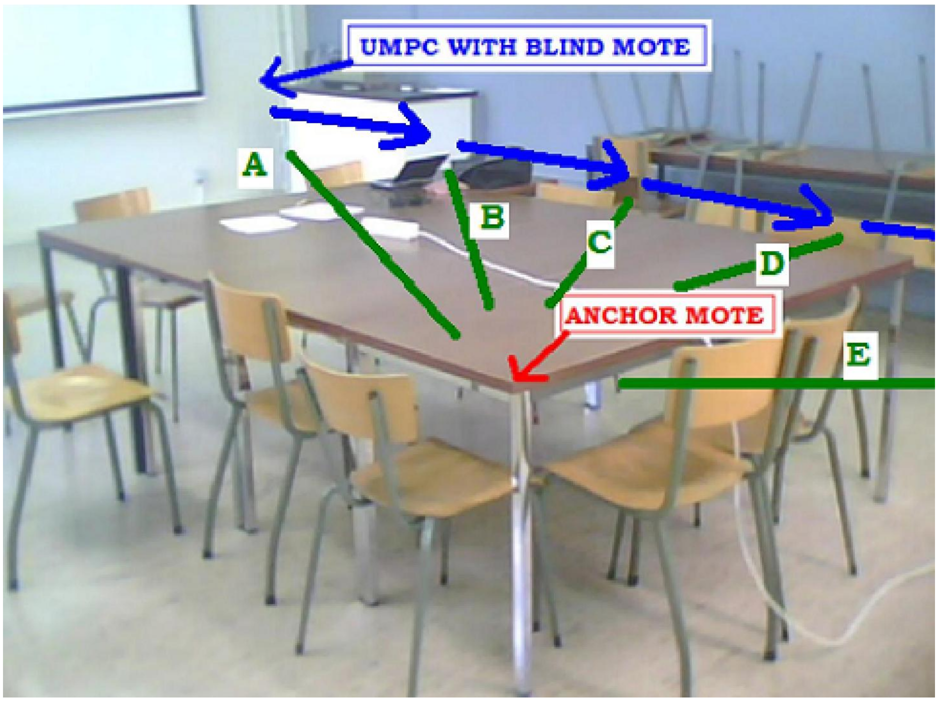


By clicking on the "Reset Button" at Blind mote, the progress starts. There will be 10 RSSI values collected. The 10 values are together with the average value reflected on the screen. Then we push on the “User Button" to collect 10 new RSSI values. Again, there will be an average calculated. Each time we push on the User Button, again 10 RSSI values are measured. This step we repeat the calculation 9 times until we have collected 100 RSSI values. In the end we get the average of 100 RSSI values reflected on the screen and a report which tells us that the test is finished. If we want to collect again, the only thing we have to do is push the reset button.

13.1.1 WSNDEP DATA COLLECTOR VERSION 2

The new version of Data Collector can not only collect RSSI values but can also collect LQI values. You can see that the latest TEST we have done at Carrefour has also LQI values. However, I have only based on the RSSI value. But the LQI value can maybe help us to improve the TRA algorithm. The value we already have collected at Carrefour shall be a good reference towards the future.

13.2 REAL-TIME TEST VAN TRA: WSNDEP REAL-TIME DEMONSTRATION PROGRAM



To demonstrate the working of Try-Angle algorithm, we have written a program in nesC calls "WSNDEP". The program is tested with TelosB motes from manufacturer Crossbow and worked under TinyOS 2.0 environment. "WSNDEP" also requires the "PRINTF" function, which is not included by the TinyOS 2.0 standard distribution. But, one can get it by browsing the TinyOS website [12]. There are two modules by "WSNDEP". One is called "ANCHOR NODE" and the other is called "BLIND NODE". Anchor node plays the role of an anchor mote (D for destination) inside a problem area. It sends constant RSSI messages to the Blind node. Blind node is a modified version of "BASE STATION". It can transfer the information from sensors network to the terminal via "SERIAL FORWARDER" function. It plays here the role of mobile node (M for mobile).

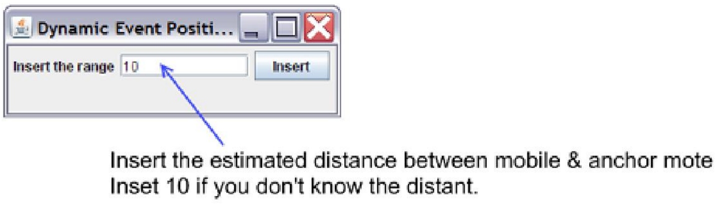
To simulate the situation we have put the anchor node somewhere far away from blind node inside the E-lab room of the University College of Antwerp's Department of Applied Engineering. Blind node is connected with our UMPC (Ultra Mobile PC) via USB-port.

13.3 TRY-ANGLE SIMULATOR AND VISUALISATOR FOR WSNDEP

To understand Try-Angle better, we have also designed a simulator to simulate the TRA-algorithm. So we can try out the different situations with the motes.

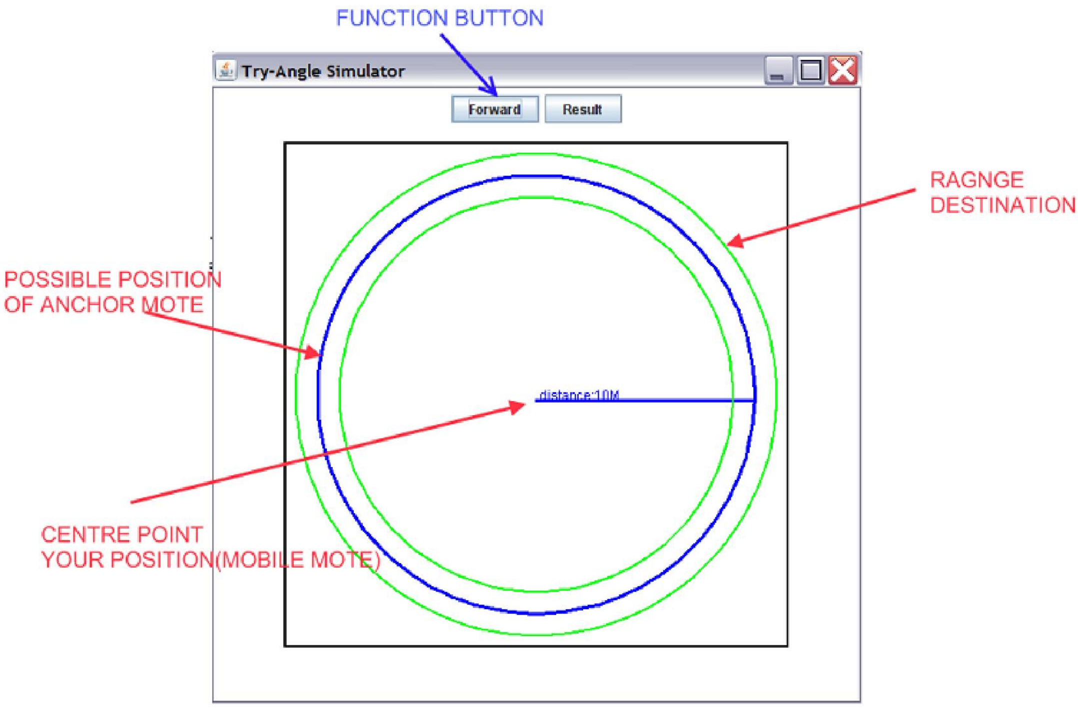
1) The requirements
   - The program is written in JAVA with Turtle Graphic. To use this program you need the JAVA editor as Eclipse and the Turtle graphic library.

2) Before beginning



   - After you have started the program, you will receive a window asking for the range value. In the case that the distance between the mobile mote and the destination mote is 10 metres, we fill in 10 (if you don't know the distance between both motes, just fill in 10). After answering the question, the next screen will appear.

3) The simulation test



- The blue circle indicates the possible place where the destination mote can be, the central point of the circle is our mobile mote. And the green circle is the threshold of the destination mote (the range of destination). Somewhere above inside the frame, we have several buttons. At the moment, we have only two buttons, this is because of the TRA algorithm. (For more details, see part theory.) Note Reset button: The result button is normally meant to reflect the path that is taken off and TRA-R. But this is still under construction. We go further with our test. We have now the possibility to go forwards, we push on forwards. And then an extra button is added. We can choose between forwards and turn back. This is the same like the theory of TRA-Stage2. When we push on forwards, we leave Stage2 and we come into Stage3. We notice that we cannot go back, but we can turn to the left jus as the theory says.

4) In the future: merge with WSNDEP Real-time Demonstration Program
   - This program is not connected with WSNDEP Real-time Demonstration Program yet. But towards the future, it will be plugged in together to build up an automatic real-time visualisation for tracking and prediction of the following steps.

5) Use for visualisation
   - Ø However at the moment we can still use it as a support for the visualisation. We work as followed. We start the WSNDEP Real-time Demonstration Program and also the simulator. We begin with the test by clicking on the reset button at the Blind mote. And if everything is running great, it asks us to move a step forward. We walk a step forward and mark our movement at the simulator by clicking on the forwards button. Now we are in Stage2 and we push on the user button. Then we get a new command of the test program. This can be forward or backward. And we do the same with the simulator. We get a visual picture of our travelling way.

13.4 SUMMARIZE TESTS

1) Indoor
   - Ø The measurement shows us that there are with indoor environment lots of fading. When the mote is near the wall, we see there is increment of RSSI strength. That has to do with reflection of the wall. Also closing and opening of the door has effect on the RSSI strength, but the effect is limited. More details see the report in the thesis's CD-ROM.

2) Outdoor
   - Ø We see there is less fading and other effect by outdoor environment. Most of the measurements are like in the theory. The further the motes are separated from each other, how weaker the signal.

CHAPTER 14: CONCLUSION AND FUTURE

14.1 CONCLUSION

At last we sum up the things we have done at our research. We have done a study on Wireless Sensor Networks. One of the main purposes of WSN is monitoring. A simple monitoring system consists of a network with wireless sensors which can gather information from the monitored area and which can report problems by unusual events. To add a mobile mote which can move through the entire network to test the details provides more advantage. But we have seen that the most WSN monitoring systems are only half automated systems. The mobile mote navigates by the operator behind the screen via tracking. In our test we focus on the problem, how we can bring a mobile mote to a certain mote(point to point) without any human intervention, without knowledge of the orientation of the mobile mote, without the help of other reference points " anchor motes" and the position of the destination mote, where the mobile mote goes does not matter. We have developed a new algorithm which we called "Try-Angle" with the requirement: a direct connection between the mobile mote and the destination mote.

To know more about the localisation principle under WSN, we have done a study about the possible localisation techniques and the most popular localisation algorithms inside the WSN area. We have done also a study about the signals like the different causes of signal fluctuation as orientation of the antenna, fading and shadowing, effect of the signals through the wall and how can we avoid fluctuation? As power is one of the most scarcely element at WSN, we have written a chapter about it. Try-Angle seems simple, however there sits a big logic behind it. Here, we have discussed the several parameters, error rate and the reliability of TRA. We think it is also important to place our algorithm next to another localisation algorithm. And finally, we discuss the different measurements and results we have calculated.

We believe that our algorithm leads to new possibilities at WSN, localisation and automated systems.

14.2 THE FUTURE

Wireless Sensor Networks is a big domain. There are still so many things we can test and improve. At this we are working on Level 2 of Try-Angle algorithm (TRA-L2). These increases much more the reliability by detecting the faults quicker. Here we use the characteristics of a circle where we preclude the impossible situation so that the tested area gets smaller and smaller after each step.

We have also planned to go more on the details of LQI. At the moment there is not much study about how we can use this technique with localisation. But LQI brings new information that RSSI does not have. A combination of both techniques to improve TRA is maybe possible. To be continued.

REFERENCES

[1] T. Z. Kazem Sohraby, Daniel Minoli, WIRELESS SENSOR NETWORKS: Technology, Protocols, and Applications. A John Wiley and Sons, INC, Publication, 2007.

[2] S. Bergbreiter and K. Pister, "Cotsbots: An off-the-shelf platform for distributed robotics," Proceedings of the 2003 IEEE/RSJ,Intl.Conference on Intelligent Robots and Systems,Las Vegas, p. 6, 2003.

[3] V. K. M. P. Seapahn Meguerdichian, Sasa Slijepcevic, "Localized algorithms in wireless ad-hoc networks - location discovery and sensor exposure," p. 11.

[4] Z. Yang and Y. Liu, "Quality of trilateration: Confidence based iterative localization," Master's thesis, Hong Kong University of Science and Technology, Hong Kong.

[5] "Nxtmote website," 2008. [Online]. Available: http://nxtmote.sourceforge.net/

[6] M. A. M. G. N. N. Javier Alonso, Sergio Gmez, "Experimental measurements of the power consumption for wireless sensor networks," Master's thesis, Computer Architecture Department,Universitat Politcnica de Catalunya, 2006.

[7] R. v. B. David Gay, Philip Levis, "The nesc language a holistic approach to networked embedded systems," p. 11. [Online].Available: http://nescc.sourceforge.net

[8] W. M. Bart Scheers and B. Lauwens, "Developments on an ieee 802.15.4-based wireless sensor network," Journal of telecommunications and information technology, p. 8, 2008.

[9] D. T. Frank Reichenbach, Jan Blumenthal, "Comparing the efficiency of localization algorithms with the power-error-product," p. 6, 2008.

[10] G. J.-P. D. S. L. Ottoy Geoffrey, Van Nieuwenhuyse Anneleen,"Indoor localisation techniques: Comparison between the use of rss and toa," Master's thesis, KaHo Sint Lieven Department Industrieel Ingenieur, G.Desmetstraat.1, B-9000 GENT, 2007.

[11] Y. O. M. M. Masashi Sugano, Tomonori Kawazoe, "Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard," Master's thesis, School of Comprehensive rehabilitation and Graduate School of Information Science and Technology, Osaka Prefecture University,.

[12] R. S. Joseph Polastre and D. Culler, "Telos: Enabling ultra-low power wireless research," Master's thesis, Computer Science Department,University of California, Berkeley, CA 94720, 2005.

[13] K. Srinivasan and P. Levis, "Rssi is under appreciated," p. 5.

[14] S.-Y. L. K.-M. C. P. H. Tsung-Han Lin, I-Hei Ng, "A microscopic examination of an rssi-signature-based indoor localization system," Master's thesis, Department of Electrical Engineering, Graduate Institute of Networking and Multimedia, Graduate Institute of Communication Engineering, National Taiwan University, 2008.

[15] D. Puccinelli and M. Haenggi, "Multipath fading in wireless sensor networks - measurements and interpretation," p. 6, 2006.

[16] I. F. Matthias Lott, "A mulit-wall-and-floor model for indoor radio propagation," p. 5.

[17] B. Capsuto and J. Frolik, "A system to monitor signal fade due to weather phenomena for outdoor sensor systems," p. 2.

[18] S. H. Michael Mallinson, Patrick Drane, "Discrete radio power level consumption model in wireless sensor network," p. 6.

LIST OF RELEVANT WEBSITES

This page provide you extra information about the topic Wireless Sensor Networks

TinyOS Homepage

- Here you can find the newest information about Wireless Sensor Networks.
- http://www.tinyos.net/

TinyOS Documentation Wiki

- The new Wiki site for the TinyOS Documentation Working Group.
- http://docs.tinyos.net/

Xubuntos

- XubunTOS simplifies the installation of TinyOS by using a Linux live CD. The bootable live CD contains a working TinyOS environment and offers the option to perform a full installation. XubunTOS is built from Xubuntu and TinyOS 2.x Debian packages (plus the TinyOS 1.x CVS repository).

- http://toilers.mines.edu/Public/XubunTOS

TinyOS Tutorials

- The tutorials are intended to get you started with TinyOS. They show you the basics of writing, compiling, and installing TinyOS applications. They introduce the basic TinyOS abstractions: computation, communication, sensing, and storage. The later tutorials go a little deeper into some of the more advanced areas of TinyOS, such as handling interrupts, power management, and how platforms are organized.
- http://docs.tinyos.net/index.php/TinyOS_Tutorials

Cotsbots website

- The CotsBots are inexpensive and modular mobile robots built entirely from commercial off-the-shelf components. These robots provide a convenient platform on which to investigate algorithms, cooperation, and distributed sensing in large (> 50) robot networks. Each robot is small (13cm x 6.5cm base) and costs under $200.
- This website of Berkeley tells you how to build your own Cotsbots.
- http://www-bsac.eecs.berkeley.edu/projects/cotsbots/index.html

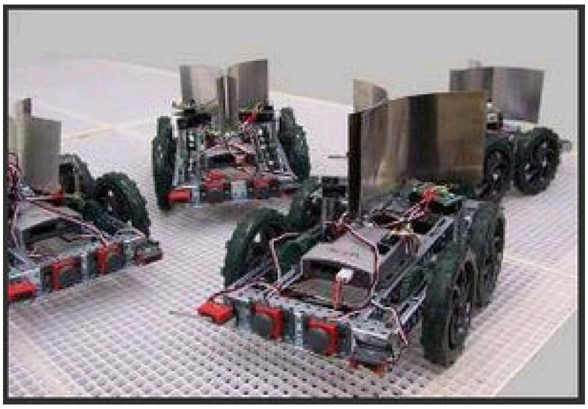
Octopus website

- Octopus is an open-source visualization and control tool for sensor networks in the TinyOS 2.x environment. Octopus provides users with a graphical user interface (GUI) for viewing the live sensor network topology. It also allows the user to control the behavior of one, many, or all sensor nodes, such as the sampling period, the radio duty cycle, or triggering mode.
- http://csserver.ucd.ie/~rjurdak/Octopus.htm

CC2420

- The CC2420 is a true single-chip 2.4 GHz IEEE 802.15.4 compliant RF transceiver designed for low-power and low-voltage wireless applications. CC2420 includes a digital direct sequence spread spectrum baseband modem providing a spreading gain of 9 dB and an effective data rate of 250 kbps.
- Architecture Radio Stack:
  - http://www.tinyos.net/tinyos-2.x/doc/html/tep126.html
- Datasheet
  - http://focus.ti.com/lit/ds/symlink/cc2420.pdf

Crossbox TelosB website

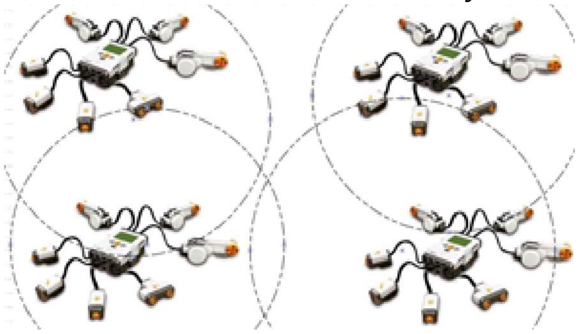


- Here you will get the information about the newest activities about “TelosB”.
- At this moment they are doing research about Swarm Navigation with TelosB Motes

- http://blog.xbow.com/xblog/telosb/index.html

NXTMOTE website

This is maybe an alternative mobile mote for our project. In place of building Cotsbots, we can use this. However they are still in development phase.



- TinyOS is a small operating for small (wireless) sensors. LEGO MINDSTORMS NXT is a platform for embedded systems experimentation: The combination of NXT and TinyOS is NXTMOTE.
- http://nxtmote.sourceforge.net/

Bijlagen

2008

Dynamic Event Positioning

using Wireless Sensor Networks

International Project v1.0

Reporter: Man Hung, WONG
Applied Engineering: Electronics-ICT
University College of Antwerp
6-4-2008

PREFACE

SUBJECT: TIMEPATH

- Version 1.0
- Date: 6 April 2008

REPORTER:

- Man Hung(Kenny), WONG
- Email: manhung.wong@gmail.com

PROMOTOR:

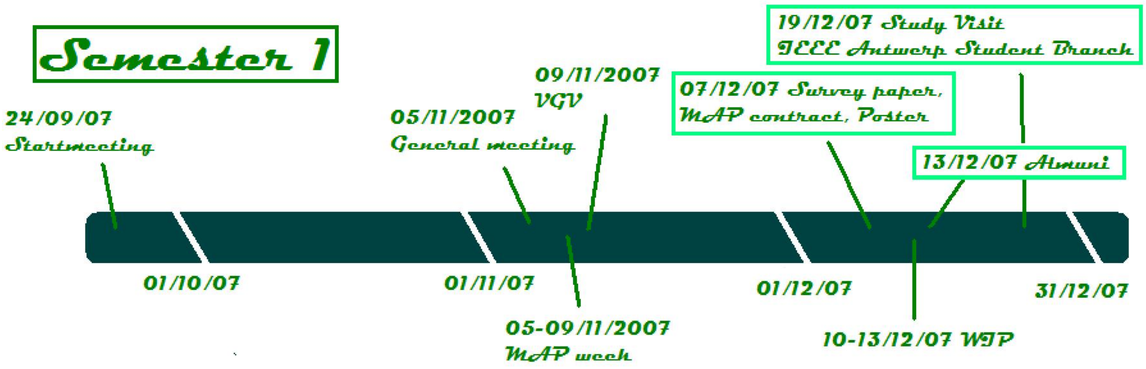
- Jeroen, DOGGEN
- Email: j.doggen@ha.be

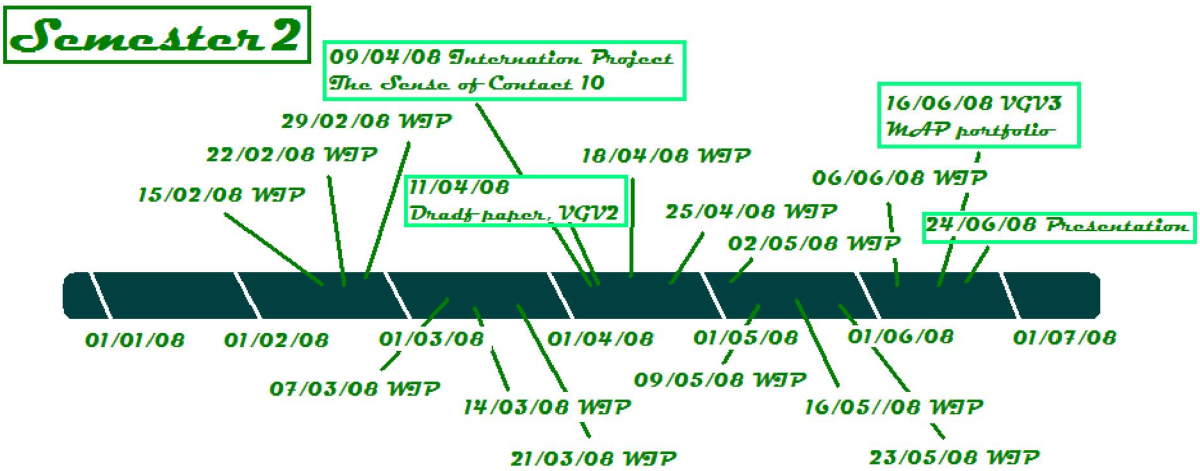
PROJECT:

- Dynamic Event Positioning using Wireless Sensor Networks
- Abbreviation: WSN-DEP

TIMEPATH

TIMEPATH OF DYNAMIC EVENT POSITIONING





2008

Dynamic Event Positioning

using Wireless Sensor Networks

Manual WSN-DEP v1.0

Reporter: Man Hung, WONG
Applied Engineering: Electronics-ICT
University College of Antwerp
15-6-2008

PREFACE

SUBJECT: MANUAL WSN-DEP

- Version 1.0
- Date: 15 June 2008

REPORTER:

- Man Hung(Kenny), WONG
- Email: manhung.wong@gmail.com

PROMOTOR:

- Jeroen, DOGGEN
- Email: j.doggen@ha.be

PROJECT:

- Dynamic Event Positioning using Wireless Sensor Networks
- Abbreviation: WSN-DEP

MANUAL WSN-DEP

MANUAL FOR DYNAMIC EVENT POSITIONING

INDEX

This manual is splits into 3 parts:

- Part 1 "Preparation" gives an overview of software's and hardware's for the installation.
- Part 2 "Testing TinyOS 2.0". Here we will test the correctness of installation.
- Part 3 "WSN-TRA". Here you will be instructed to install WSN-TRA into TelosB mote.

REQUIREMENTS

Software

- TinyOS 2.0
- Xubuntos
- FTDI driver
- VMware workstation/player

Hardware

- 2x TelosB motes
- PC with USB ports

PART 1 PREPARATION

This part gives you an overview of software's and hardware's for the installation.

INSTALL VMWARE

First you need a virtual machine. I prefer using VMware player or VMware workstation.
You can get VMware from this website http://www.vmware.com/
Note: VMware Player is free.

INSTALL XUBUNTOS AND TINYOS

If you are newbie, it is not easy to install TinyOS on your Windows. I prefer using "Xubuntos", because it simplifies the installation of TinyOS by using a Linux live CD. The bootable live CD contains a working TinyOS environment and offers the option to perform a full installation. Xubuntos is built from Xubuntu and TinyOS 2.x Debian packages (plus the TinyOS 1.x CVS repository).

You can get the Xubuntos iso from this website:
http://toilers.mines.edu/Public/XubunTOS

INSTALLING THE FT2DXX USB DRIVERS

TinyOS uses the FTDI VCP drivers which cause the TelosB USB to appear as an additional COM port available to the PC. However to reprogram the mote, the msp430-jtag tool uses FTDI D2XX drivers. The D2XX drivers allow the tool direct access to the TelosB USB device through a series of DLL function calls.

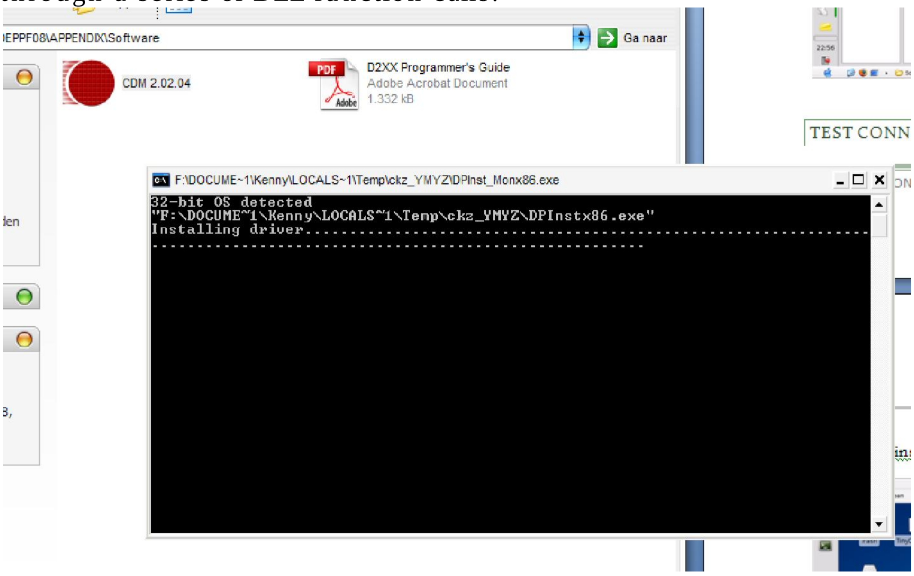


Photo taken during installation

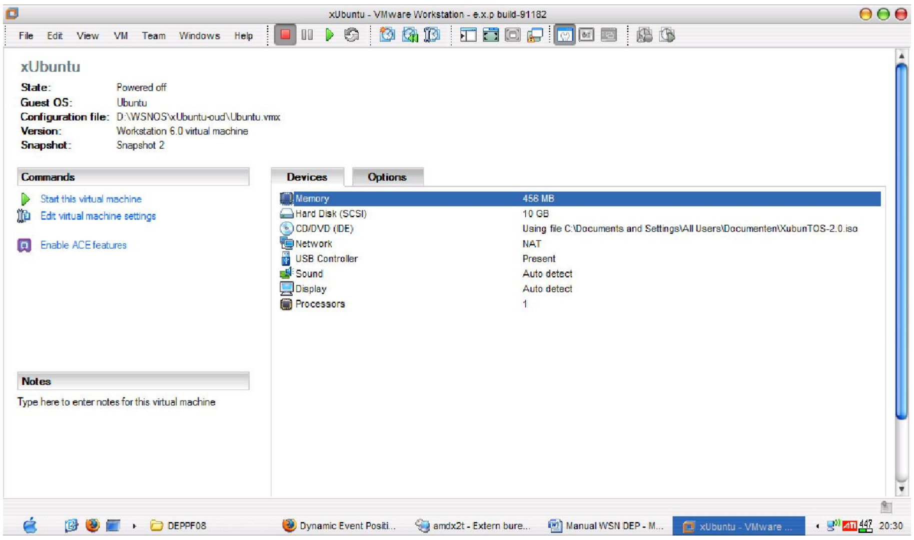
You can install it (CDM 2.02.04.exe) from the directory "X:\DEPPF08\APPENDIX\Software". You will also find a manual about D2XX Programmer's Guide in the same folder.

Note: If you still get problem after installing the driver, you can consult this site
http://www.cs.utah.edu/~vchakra/

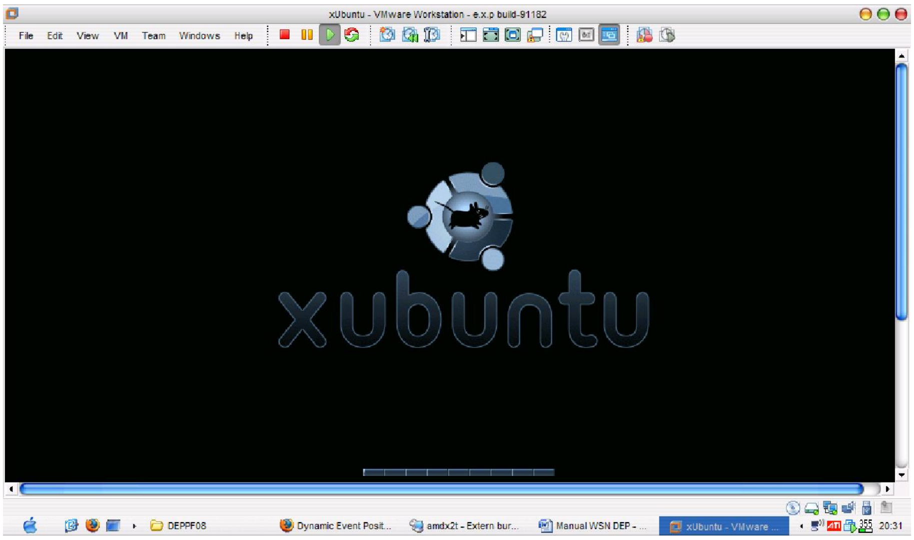
PART 2 "TESTING TINYOS 2.0"

Here we will test the correctness of installation.

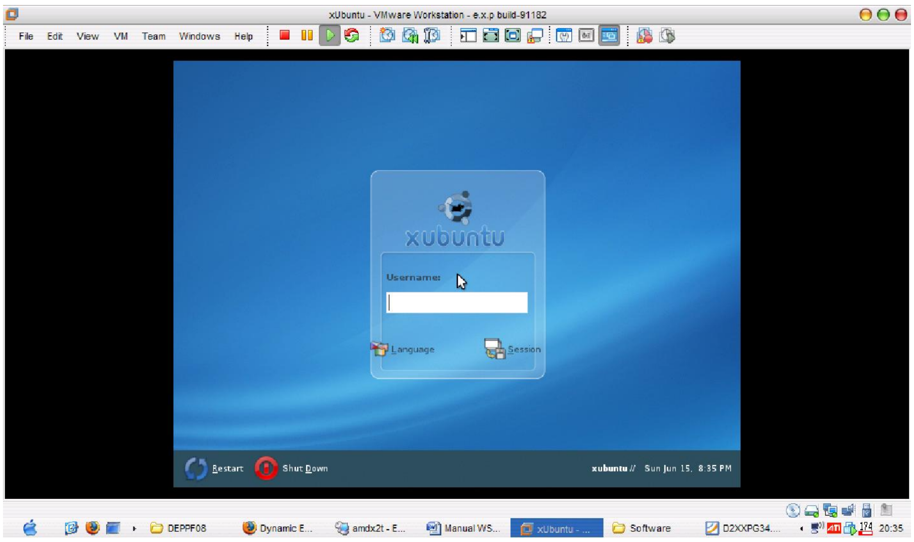
START UP XUBUNTOS IN VMWARE



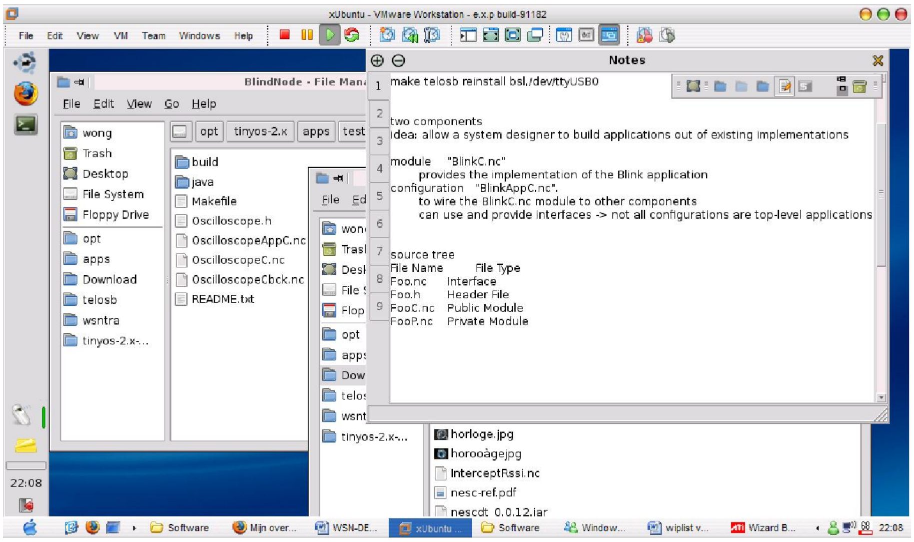
VMware with Xubuntos



Loading screen of Xubuntos inside VMware



Login screen of Xubuntos

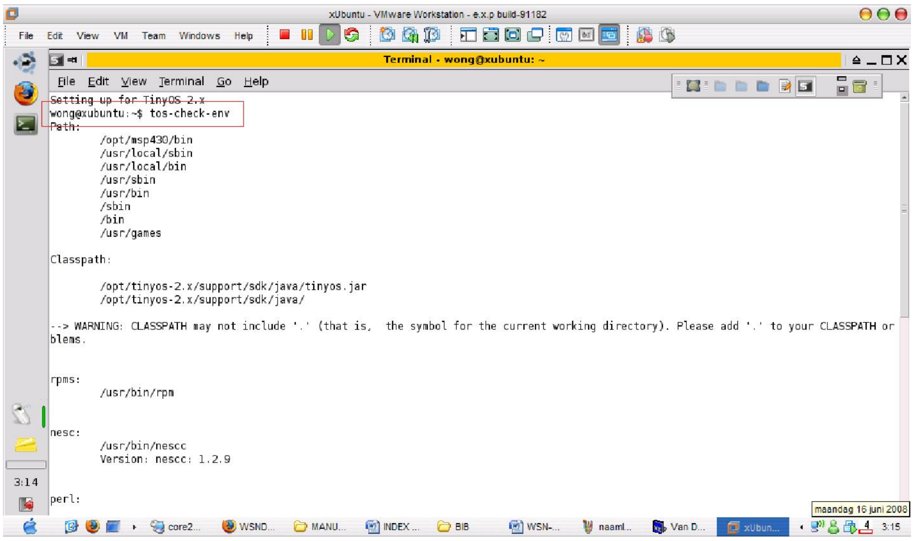


Xubuntos Operating System

Testing TinyOS installation

To test the correctness of TinyOS 2.0 installation type the following command by terminal.

```
$ tos-check-env
```



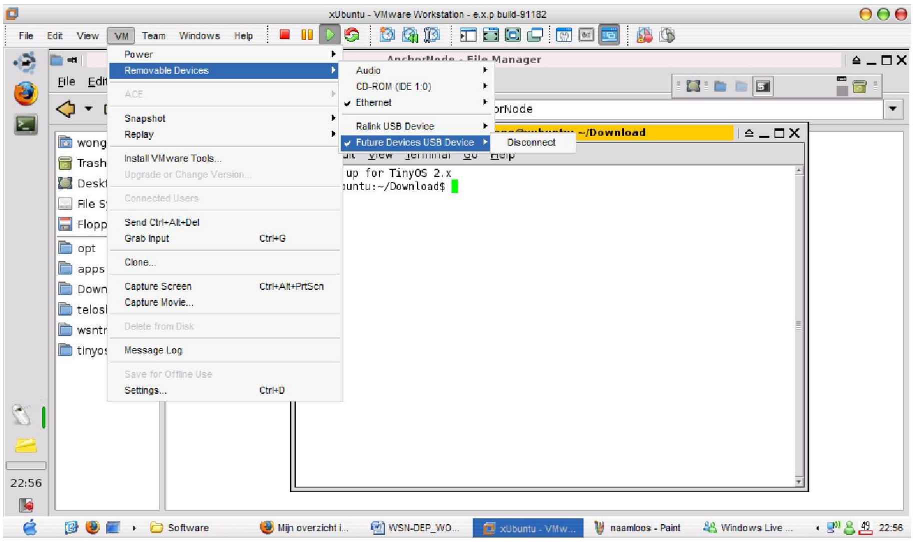
You will get a list of information about the installed software's and paths.

Note: TinyOS works perfect with JAVA JDK/SDK 1.42 or 1.5. If you use JAVA JDK6.0, you can get warning. And some programs won't work.

CONNECTION TELOSB WITH VMWARE

To use TelosB need connect it with VMware. You can do it as follow:

- VM->removable devices->future device usb->connect



TEST CONNECTION TELOSB INSIDE TINYOS

To test the connection between TelosB mote and TinyOS, you type this in the terminal:

```
motelist
```

It will show you the port of the mote.

```
Setting up for TinyOS 2.x
wong@xubuntu:~$ motelist
Reference  Device           Description
---------- ---------------- ---------------------------------------------
XBP4HMK6   /dev/ttyUSB0     XBOW Crossbow Telos Rev.B
wong@xubuntu:~$
```

PART 3 “WSN-DEP”

Here you will be instructed to install WSN-TRA into TelosB mote.

To demonstrate the working of Try-Angle algorithm, we have written a program in nesC calls WSNDEP. The program is tested with TelosB motes from manufacturer Crossbow and worked under TinyOS 2.0 environment.

INSTALLING PRINTF

WSNDEP also requires the "PRINTF" function, which is not included by the TinyOS 2.0 standard distribution. But you can get it by browsing the TinyOS website http://www.tinyos.net/.

Installation
You have to apply the patch “tinyos-2.0-printf.patch” in order to allow the printf library to compile correctly for atmega128x based platforms (i.e. mica2, micaz):

You can find it under “X:\DEPPF08\APPENDIX\Printf”.

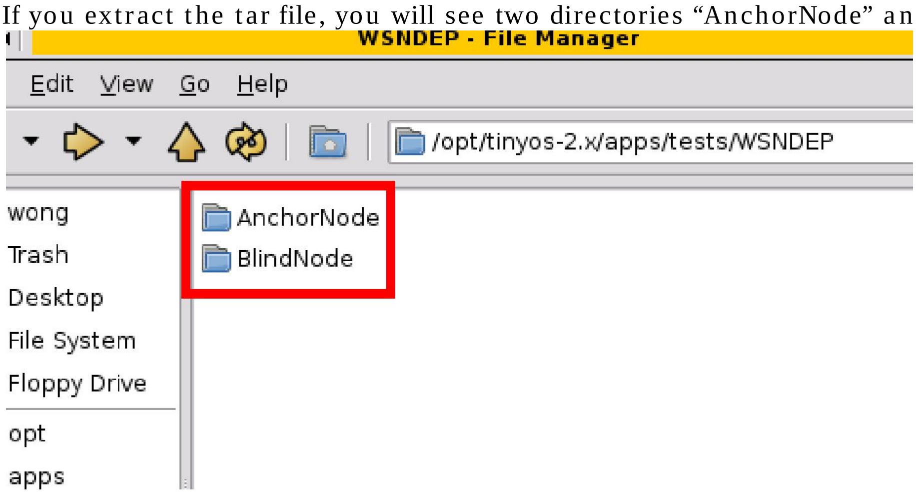
```
cp tinyos-2.0-printf.patch $TOSROOT/..
cd $TOSROOT/..
patch -p0 < tinyos-2.0-printf.patch
```

Copy the directory “Printf” into “opt/tinyos-2.x/apps/tutorials”
The Prinf directory can be found on this CD under
“X:\DEPPF08\APPENDIX\Printf\Printf”

For extra information about installation and tutorial of “Printf”, you can consult this page
http://docs.tinyos.net/index.php/The_TinyOS_printf_Library

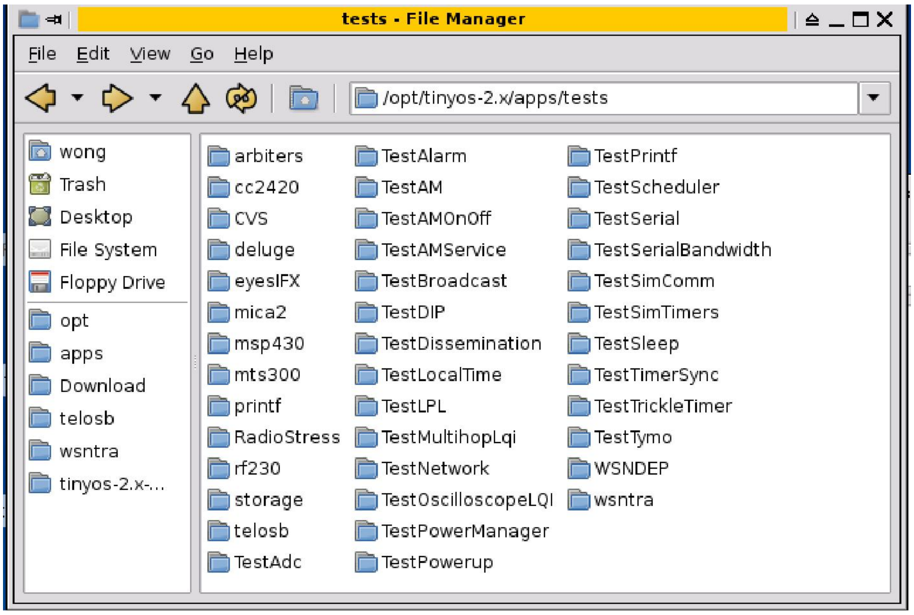
THE SOURCE CODE

If you extract the tar file, you will see two directories “AnchorNode” and “BlindNode”.



There are two modules by WSNDEP. One is called ANCHOR NODE and the other is called BLINDNODE. Anchor node plays the role of an anchor mote (D for destination) inside a problem area. It sends constant RSSI messages to the Blind node. Blind node is a modified version of BASE STATION. It can transfer the information from sensors network to the terminal via "SERIAL FORWARDER" function. It plays here the role of mobile node (M for mobile).

If you have installed “TinyOS2.O” correctly. You will have the following path: “/opt/tinyos-2.x/apps/”.

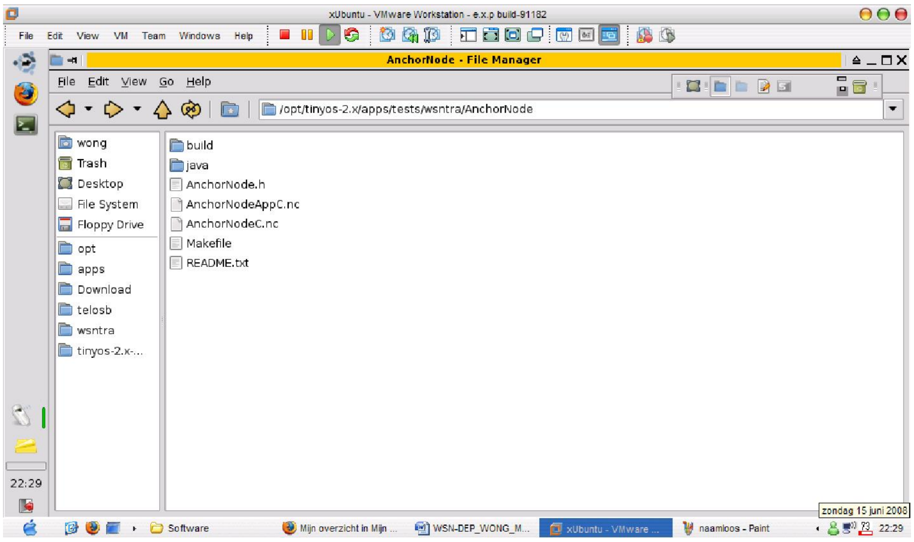


Go into directory “test” and make a map calls “WSNDEP”.

Note: Directory “test” is the place to test you own application.

You copy the both directories “AnchorNode” and “BlindNode” into “WSNDEP”.

COMPILE THE SOURCES



Inside the directory “AnchorNode” you will find different files.
File with extension
*.h : this is the header file
*.nc: is programs written in nesC

There are also two kinds of components:

1. Module "AnchorNodeC.nc"
   - provides the implementation of the AchnorNode application

2. Configuration "AnchorNodeAppC.nc".
    - to wire the AnchorNodeC.nc module to other components
    - can use and provide interfaces -> not all configurations are top-level applications

The idea of TinyOS:

- allow a system designer to build applications out of existing implementations

Note: more about the file extension see nesC programming guide.

COMPILE ANCHORNODE

To compile the program you type the following command

```
Make "platform"
```

Because we are using TelosB, we type this

```
Make telosb
```

Output

```
Setting up for TinyOS 2.x
wong@xubuntu:/opt/tinyos-2.x/apps/tests/WSNDEP/AnchorNode$ make telosb
mkdir -p build/telosb
   compiling AnchorNodeAppC to a telosb binary
ncc -o build/telosb/main.exe -Os -O -mdisable-hwmul -Wall -Wshadow -
DDEF_TOS_AM_GROUP=0x7d -Wnesc-all -target=telosb -fnesc-cfile=build/telosb/app.c -
board= -I/opt/tinyos-2.x/tos/lib/printf -DIDENT_PROGRAM_NAME=\"AnchorNodeAppC\" -
DIDENT_USER_ID=\"wong\" -DIDENT_HOSTNAME=\"xubuntu\" -
DIDENT_USER_HASH=0x828238d9L -DIDENT_UNIX_TIME=0x4855f9efL -
DIDENT_UID_HASH=0xd7fae66fL  AnchorNodeAppC.nc -lm
   compiled AnchorNodeAppC to build/telosb/main.exe
         21372 bytes in ROM
           864 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
   writing TOS image
wong@xubuntu:/opt/tinyos-2.x/apps/tests/WSNDEP/AnchorNode$
```

INSTALL THE PROGRAM INTO TELOSB MOTE

To install the compiled program into telosb mote, you type

```
make telosb reinstall
```

Output

```
wong@xubuntu:/opt/tinyos-2.x/apps/tests/WSNDEP/AnchorNode$ make telosb reinstall
cp build/telosb/main.ihex build/telosb/main.ihex.out
   found mote on /dev/ttyUSB0 (using bsl,auto)
   installing telosb binary using bsl
tos-bsl --telosb -c /dev/ttyUSB0 -r -e -I -p build/telosb/main.ihex.out
MSP430 Bootstrap Loader Version: 1.39-telos-8
Mass Erase...
Transmit default password ...
```

```
Invoking BSL...
Transmit default password ...
Current bootstrap loader version: 1.61 (Device ID: f16c)
Changing baudrate to 38400 ...
Program ...
21404 bytes programmed.
Reset device ...
rm -f build/telosb/main.exe.out build/telosb/main.ihex.out
wong@xubuntu:/opt/tinyos-2.x/apps/tests/WSNDEP/AnchorNode$
```

If this is the first time you compile the program, you type

```
make telosb install
```

If you have connected more than one telosb mote with you pc, you can define the port of the mote for installation.

```
make telosb reinstall bsl,/dev/ttyUSB0
```

In this example the port is USB0

You can also provide an “id” for your mote

```
make telosb reinstall bsl,3
```

in this example, the id is 3

You do the same with the blind mote.
You compile and install it in another mote.

SIMULATION

To test the working you put the Anchor node somewhere far away from Blind node (for example 10 meters). Blind node is connected with your laptop via USB-port.

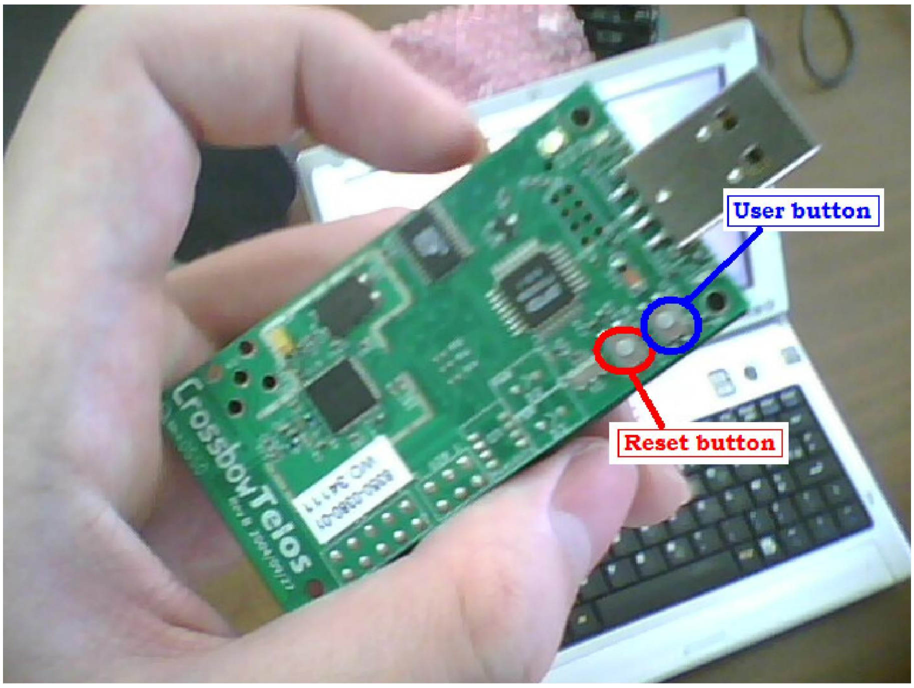
You need to start the serial forwarder of printf before the test, else you can’t see the notify message on screen.
Go to the directory of printf and type the following commands.
1. make (this build the java application, only need for the first time)

2. java PrintfClient -comm serial@/dev/ttyUSB0:telosb

Output

```
wong@xubuntu:/opt/tinyos-2.x/apps/tutorials/Printf$ java PrintfClient -comm
serial@/dev/ttyUSB0:telosb
Thread[Thread-1,5,main]serial@/dev/ttyUSB0:115200: resynchronising
```

You can begin the test now. To do it, you just need to push the reset button on the Blind node.



A message will notify you about the start process onscreen. At the same time M will calculate the RSSI-value through the message from D. When M is clear with measurement, a message will notify you to go one step ahead.

Output

```
BLnode restart
In-movement: FIRST!
RSSIval_filter_org = -8
RSSIval_filter_old = -8
1 x ahead!
```

You go one step ahead with the blind mote which is connected with your laptop (for example 1 meter = 1step). When you are clear, you press the user-button (not the rest-button). Mobile mote will measure the new RSSI value and compare it with the original one. Again you will get message about the next step.

Output

```
Calc new RSSI
In-movement: in try-ahead!
 RSSI old = -8
 RSSI new = -1
 Compare destination
VGL rssi - beter na vgl RSSI
RSSI is beter, good direction
1 x ahead!
```

Note: If you test it with a pc, you can also move the anchor mote in place of blind mote.

Other situation

Output

```
Calc new RSSI
```

```
In-movement: in try-ahead!
 RSSI old = -1
 RSSI new = -2
 Compare destination
VGL rssi - slechter na vgl RSSI
 RSSI is worst, wrong direction
Try Left
turn 90 degree to left!
1 x ahead!
```

After several step, you will get close enough to the anchor node. If this is the case, you will get a message “you reach your destination”.

Output

```
Calc new RSSI
In-movement: in try-left!
RSSI old = -2
 RSSI new = 23
Compare destination
Destination reached
```

Note: Number of steps is depending on situation and position between anchor and blind mote.

WSNDEP MANUAL

WSNDEP Data Collector

To know more about the characteristic of RSSI by Telosb, we have done several measurements at different places. To make it more pleasant, we have designed a special program "WSNDEP Data Collector" to help us collect the measurements. This program is written in Nesc and works under TinyOS 2.0 environment. This program can collect in total 100 RSSI values with a interruption by each 10 RSSI values. At the end of the test, we get all the RSSI values together with the average RSSI values on screen.

The measurements will be done as followed.

For the measurements we need 2 (TelosB) motes. The Anchor mote has the task to constantly send packets to Blind mote. And we connect the Blind mote with the UMPC(Ultra Portable). When the Blind mote receives a packet, he calculates the RSSI value. Next, he sends the RSSI value to our UMPC, which shows the result on our screen. We lay down the Blind mote (UMPC) and the Anchor mote 1 metre from each other. By clicking on the "Reset Button" at Blind mote, the progress starts. There will be 10 RSSI values collected. The 10 values are together with the average value reflected on the screen. Then we push on the "User Button" to collect 10 new RSSI values. Again, there will be an average calculated. Each time we push on the User Button, again 10 RSSI values are measured. This step we repeat the calculation 9 times until we have collected 100 RSSI values. In the end we get the average of 100 RSSI values reflected on the screen and a report which tells us that the test is finished. If we want to collect again, the only thing we have to do is push the reset button.

WSNDEP Data Collector version 2

The new version of Data Collector can not only collect RSSI values but can also collect LQI values. You can see that the latest TEST we have done at Carrefour has also LQI values. However, I have only based on the RSSI value. But the LQI value can maybe help us to improve the TRA algorithm. The value we already have collected at Carrefour, shall be a good reference towards the future.

The real-time demonstration program

To demonstrate the working of Try-Angle algorithm, we have written a program in nesC calls "WSNDEP". The program is tested with TelosB motes from manufacturer Crossbow and worked under TinyOS 2.0 environment. "WSNDEP" also requires the "PRINTF" function, which is not included by the TinyOS 2.0 standard distribution. But, one can get it by browsing the TinyOS website. There are two modules by "WSNDEP". One is called "ANCHOR NODE" and the other is called "BLIND NODE". Anchor node plays the role of an anchor mote (D for destination) inside a problem area. It sends constant RSSI messages to the Blind node. Blind node is a modified version of "BASE STATION". It can transfer the information from sensors network to the terminal via "SERIAL FORWARDER" function. It plays here the role of mobile node (M for mobile).
To simulate the situation we have put the anchor node somewhere far away from blind node inside the E-lab room of the University of Antwerp's Department of Applied Engineering. Blind node is connected with our UMPC (Ultra Mobile PC) via USB-port.

<u>TRA stage 1:</u>
To start the test, we just need to push the reset button on the Blind node. A message will notify us about the start process onscreen. At the same time M will calculate the RSSI-value through the message from D. Because RSSI is very sensitive to the environment, we have used a filter to get a more accurate result (More details see section TRA concept). When M is clear with measurement, a message will notify it to go one step ahead.

<u>TRA stage 2:</u>
We are now by stage2. After we have go one step ahead with the mote (1 metre). We press the user-button (not the rest-button). Mobile mote will measure the new RSSI value and compare it with the old one. If the new value is higher, TRA will command us to good one step forward and we will be in stage3. Otherwise, it will try out the other direction (backwards, left-side and right-side).

<u>TRA stage 3:</u>
Stage works the same as by stage 2. We just need to push the user button each time and we will get the instruction to go.

**** Note: If you are interested in the source code, you can ask the e-lab of University College of Antwerp for a copy. And if you want to know more about the working of the print library, there is also a tutorial you can consult[13].*

Try-Angle SIMULATOR AND VISUALISATOR FOR WSNDEP

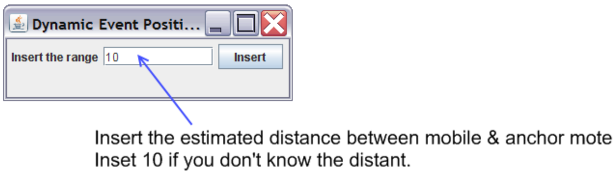
To understand Try-Angle better, we have also designed a simulator to simulate the TRA-algorithm. So we can try out the different situations with the motes.

The requirements

The program is written in JAVA with Turtle Graphic. To use this program you need the JAVA editor as Eclipse and the Turtle graphic library.

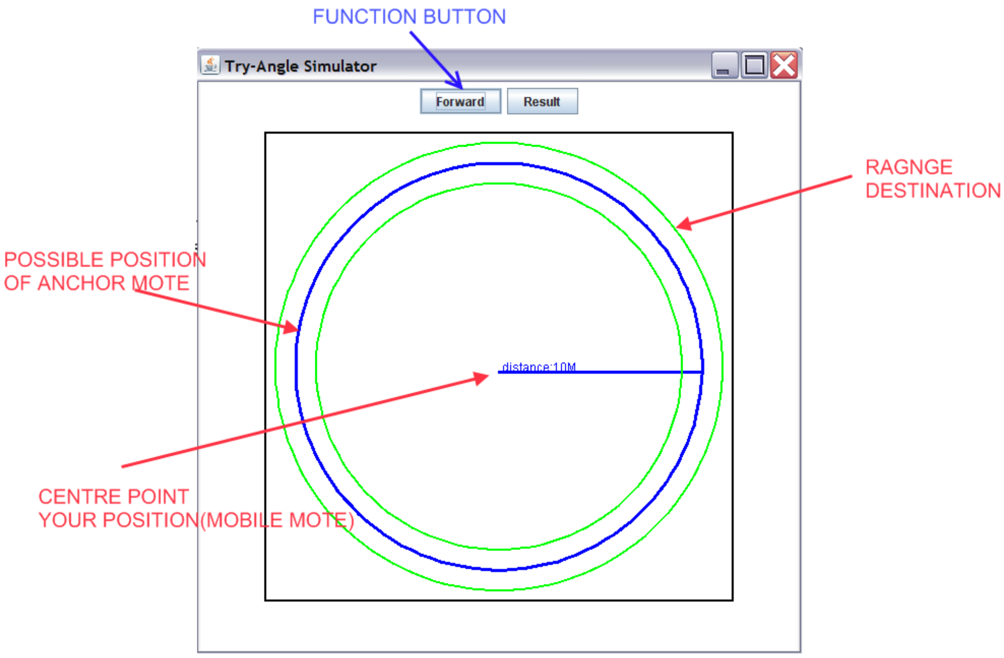
Before beginning

After you have started the program, you will receive a window asking for the range value. In the case that the distance between the mobile mote and the destination mote is 10 metres, we fill in 10 (if you don't know the distance between both motes, just fill in 10). After answering the question, the next screen will appear.



The simulation test

The blue circle indicates the possible place where the destination mote can be, the central point of the circle is our mobile mote. And the green circle is the threshold of the destination mote (the range of destination). Somewhere above inside the frame, we have several buttons. At the moment, we have only two buttons, this is because of the TRA algorithm. (For more details, see part theory.) *Note Reset button :The result button is normally meant to reflect the path that is taken off and TRA-R. But this is still under construction.* We go further with our test. We have now the possibility to go forwards, we push on forwards. And then an extra button is added. We can choose between forwards and turn back. This is the same like the theory of TRA-Stage2. When we push on forwards, we leave Stage2 and we come into Stage3 . We notice that we can not go back, but we can turn to the left jus as the theory says.



In the future: merge with WSNDEP Real-time Demonstration Program

This program is not connected with WSNDEP Real-time Demonstration Program yet. But towards the future, it will be plugged in together to build up an automatic real-time visualisation for tracking and prediction of the following steps.

Use as visulatisator

However at the moment we can still use it as a support for the visualisation. We work as followed. We start the WSNDEP Real-time Demonstration Program and also the simulator. We begin with the test by clicking on the reset button at the Blind mote. And if everything is running great, it ask us to move a step forward. We walk a step forward and mark our movement at the simulator by clicking on the forwards button. Now we are in Stage2 and we push on the user button. then we get a new command of the test program. This can be forward or backward. And we do the same with the simulator. We get a visual picture of our travelling way.

2008

Dynamic Event Positioning

using Wireless Sensor Networks

Test & Result v1.0

Reporter: Man Hung, WONG
Applied Engineering: Electronics-ICT
University College of Antwerp
15-6-2008

PREFACE

SUBJECT: TEST & RESULT

- Version 1.0
- Date: 15 June 2008

REPORTER:

- Man Hung(Kenny), WONG
- Email: manhung.wong@gmail.com

PROMOTOR:

- Jeroen, DOGGEN
- Email: j.doggen@ha.be

PROJECT:

- Dynamic Event Positioning using Wireless Sensor Networks
- Abbreviation: WSN-DEP

TEST & RESULT

TEST & RESULT OF DYNAMIC EVENT POSITIONING

REQUIREMENTS

For the test we need the following materials.

- A UMPC or laptop with TinyOS installed
- At least 1x free USB port
- 2 x TelosB motes

The used material

- Laptop:
  - Flybook: Intel Centrino ULV 1Ghz
  - 1 GB DDR Ram
- Operating system: Xubuntos (Xubuntu version with TinyOS installed)

PLACE FOR SIMULATION

E-lab "Lima Room" of University College of Antwerp Department Applied Engineering

E-lab dimension:

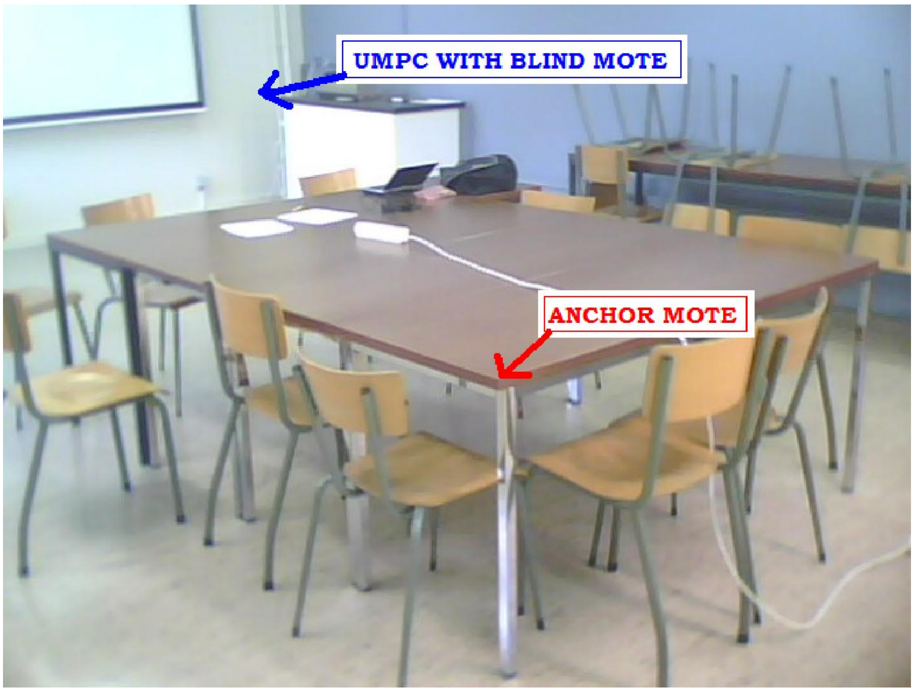
- length = 12 meter
- width= 10meter
- The result is based on the steps I have taken (1 step = +/- 1 meter)

Table:

- Length = 4 meter
- Width = 2 meter

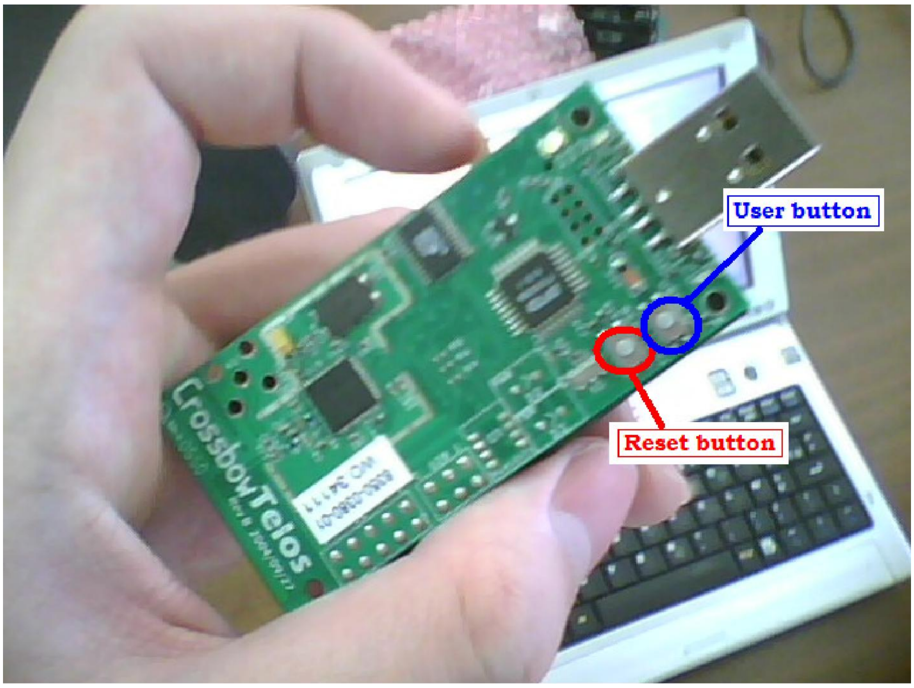
SIMULATION

To simulate the situation I have put the Anchor node at the right corner of the table.



And I stand with the Blind node at the left corner of the room. Blind node is also connected with our UMPC (Ultra Mobile PC) via USB-port.

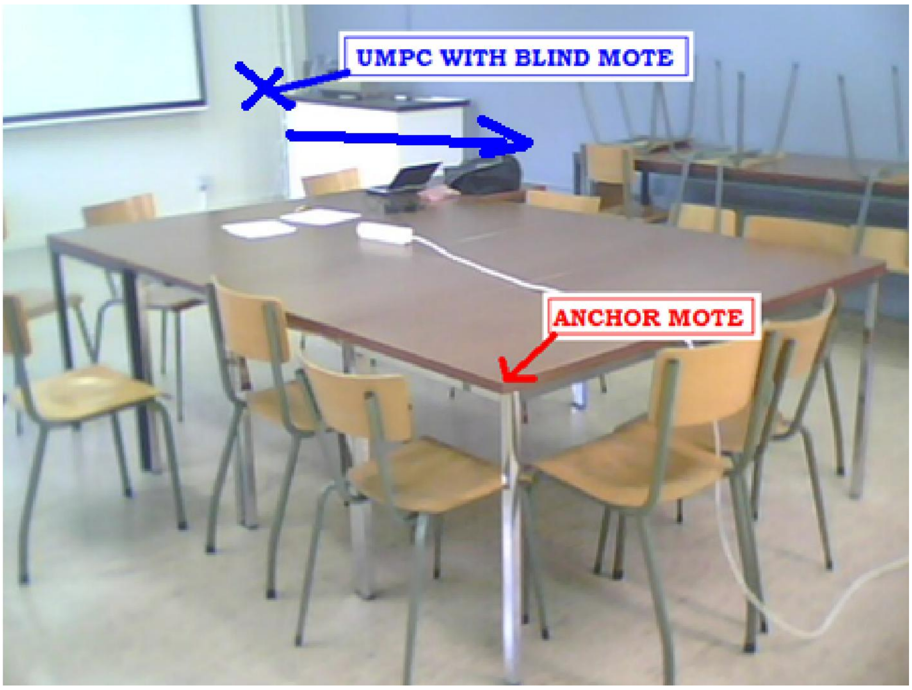
To start the test, I just need to push the reset button on the Blind node.



A message will notify me about the start process onscreen.
Output

```
BLnode restart
In-movement: FIRST!
RSSIval_filter_org = -16
RSSIval_filter_old = -16
1 x ahead!
```

At the same time M will calculate the RSSI-value through the message from D. Because RSSI is very sensitive from the environment, we have used a filter to get a more accurate result. To do this, mobile mote will measure the average RSSI-value through sampling of 10 RSSI-values. The average value will be saved automatically as the original value and old value. When M is clear with measurement, a message will notify it to go one step ahead.

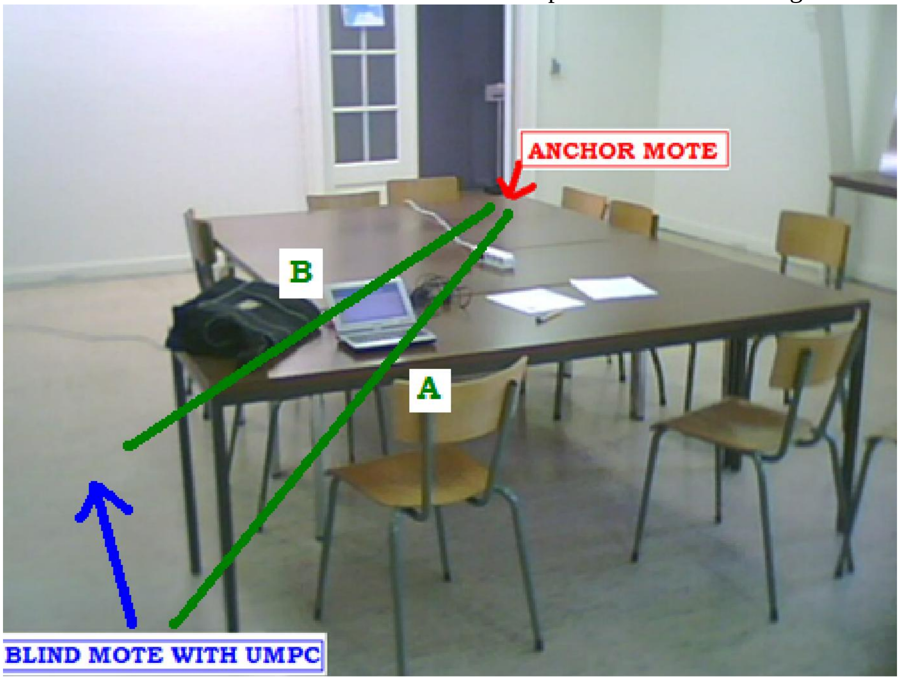


Due to the person who is responsible for the making of Cotsbot has delay. I will play the role of Cotsbot. So I go one step ahead with the mote (1 meter).

WSNDEP PHASES

There are two phases by WSNDEP. The first one is the "INITIAL PHASE" and the second one is the "PATH FINDING PHASE". The initial phase is intended to learn the environment. Here mobile mote will try out the four directions until the new RSSI-value is higher than the original RSSI-value.

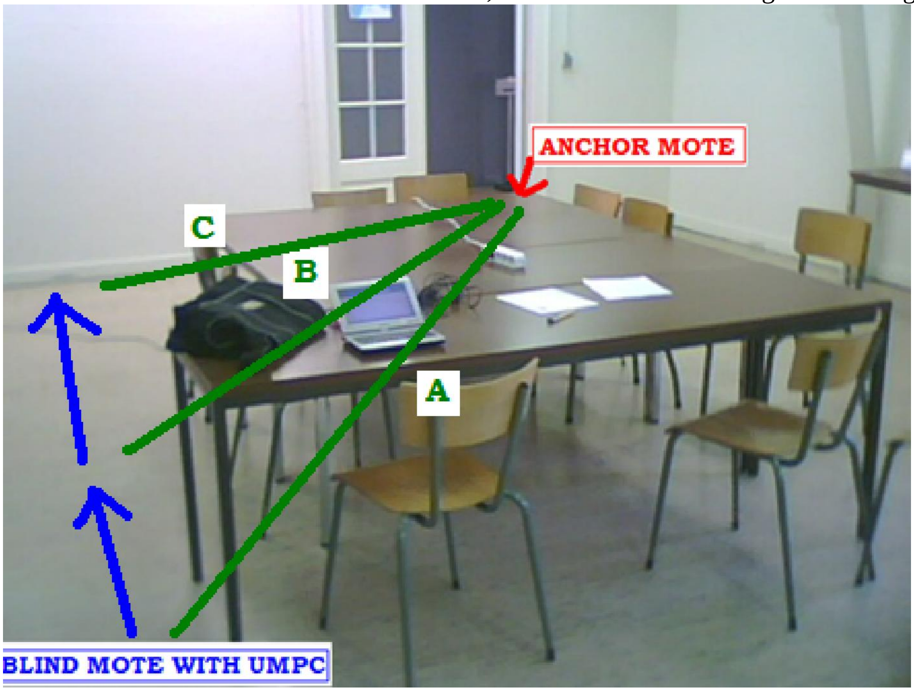
I'm now by the initial phase, so we press the user-button (not the rest-button). Mobile mote will measure the new RSSI value and compare it with the original one.

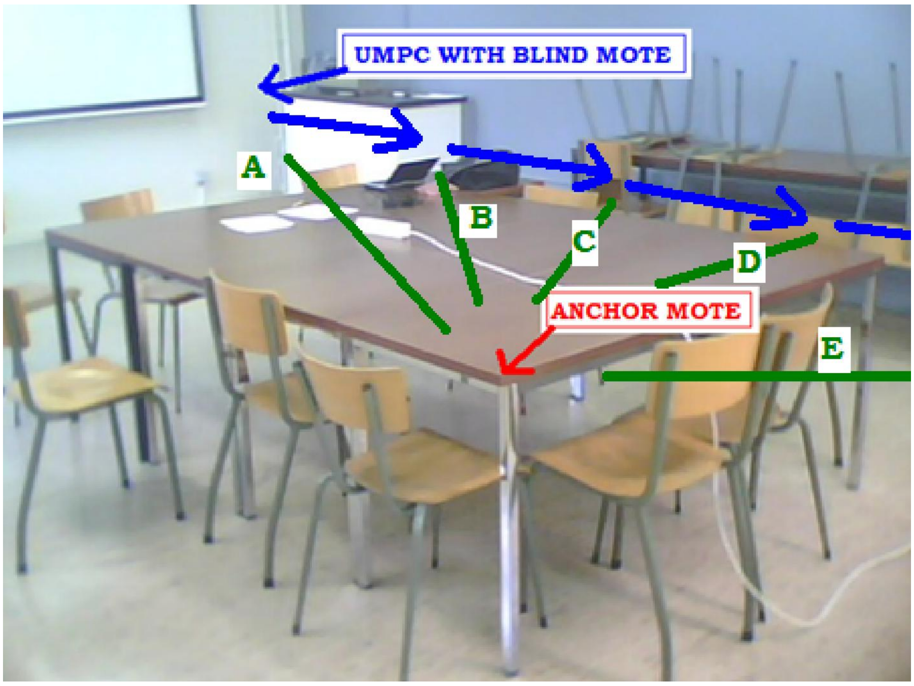


B is shortest than A, B has higher RSSI value than A.

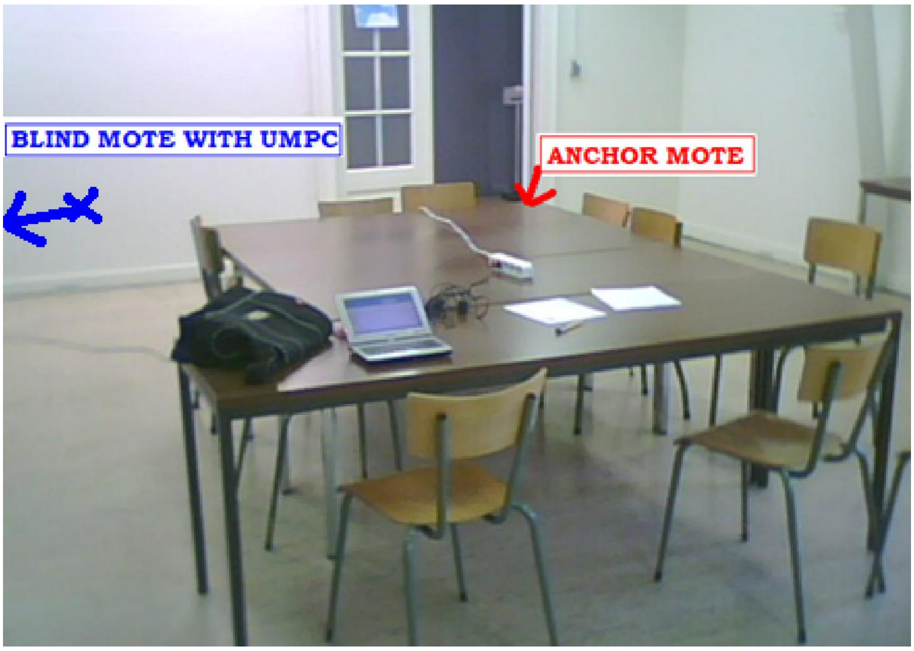
The new RSSI value is higher than before. Blind node will go to the next phase "PATH FINDING". In other case it will try out the other directions (backward, left-side and right-side).

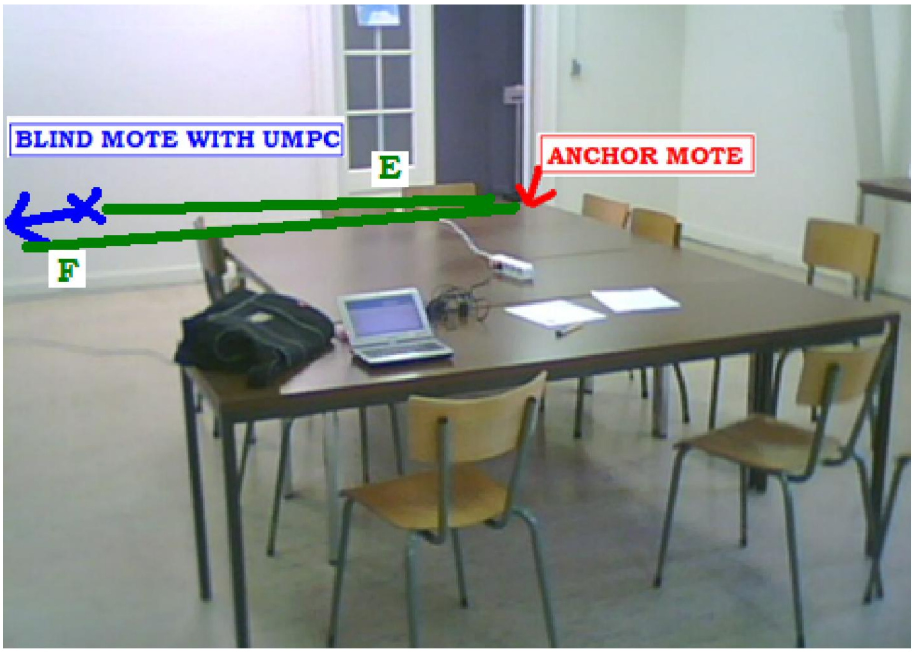
Because the RSSI value is better now, it commands me to go ahead again.



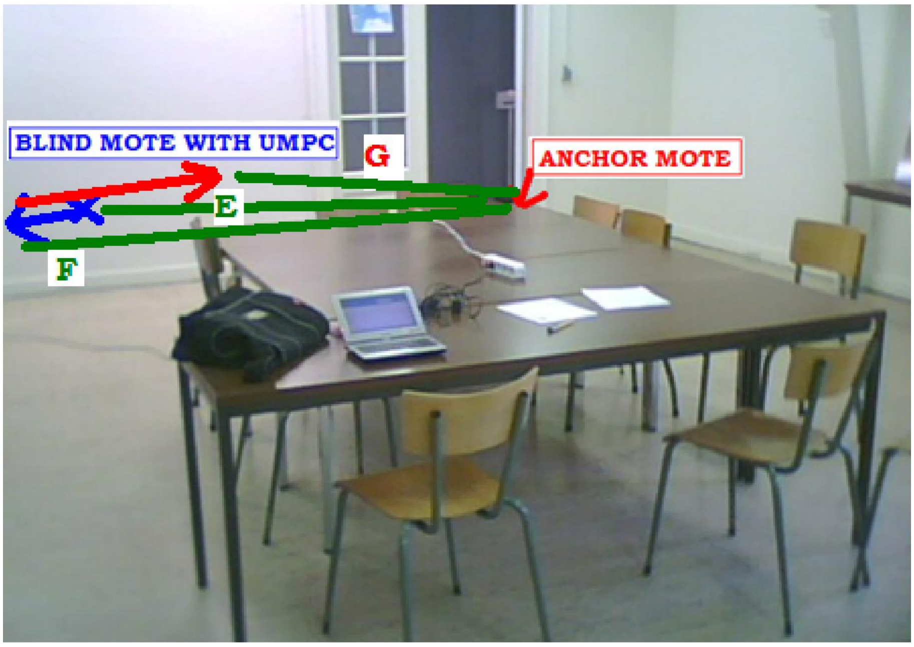


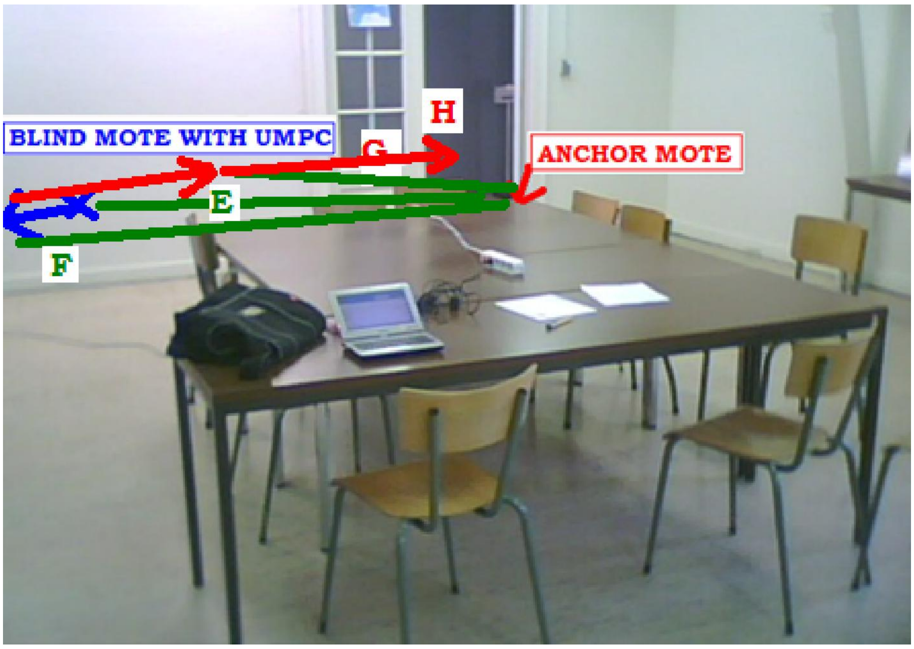
Blind node command me go ahead until step "E". Because the RSSI-value by step E is lower than by step D.





Because going ahead make no sense, Blind node command me try-left. So I turn one step to left. Unfortunately try-left is not a good choice. The new value "F" is lower than "E". So Blind node command me to try-right side, it mean turn around and go two steps ahead. At the same moment "the path finding phase" has excluded the direction "try-left". In the future I can always going ahead or turning right.





The new value is better. Blind node command to go one step ahead again and I reach the destination.

THE TEST PROVES THE FUNCTIONING OF TRY-ANGLE ALGORITHM.

Measurements Telosb RSSI

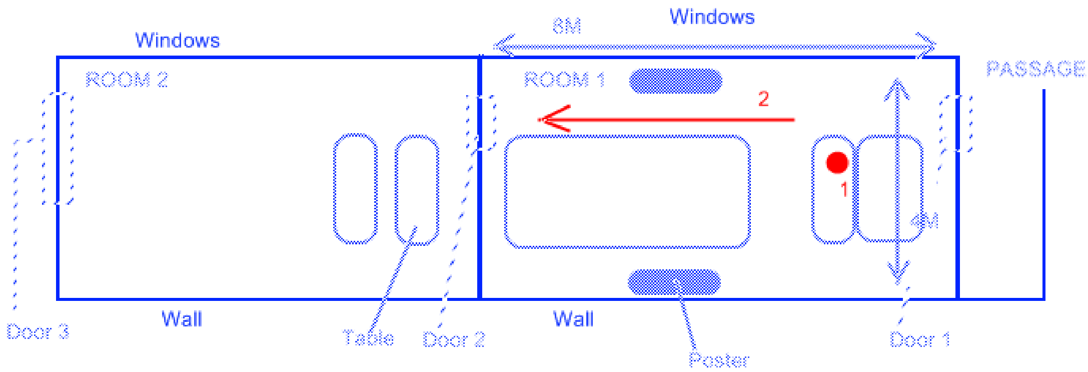
Date: 2008-08-27
Time: 11u30-16u
Place: Hogeschool Antwerpen Paardenmarkt

Sample RSSI

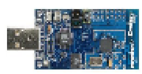
<u>Duration</u>: 6 secondes for 10 RSSI value
<u>Number of RSSI per meter:</u> For each distance there will be collected 10X10 = 100 RSSI value. Cause of the effect of our hand, the first RSSI value will be ignored. There will be collected per M 100 RSSI values, however in reality there only will be used 80 RSSI values for the result.

Measurement 1: measurering

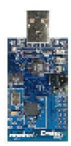
The measurement will be executed by placing an anchor mote on the table and fixing the mobile mote to the UMPC. The height of the table is approximately 80 cm. And the UMPC is hold by a human. The height of the UMPC is approximately 1m20 of the ground and with the telosB turned above.



(2) Mobile mote orientation



(1)Anchor mote orientation

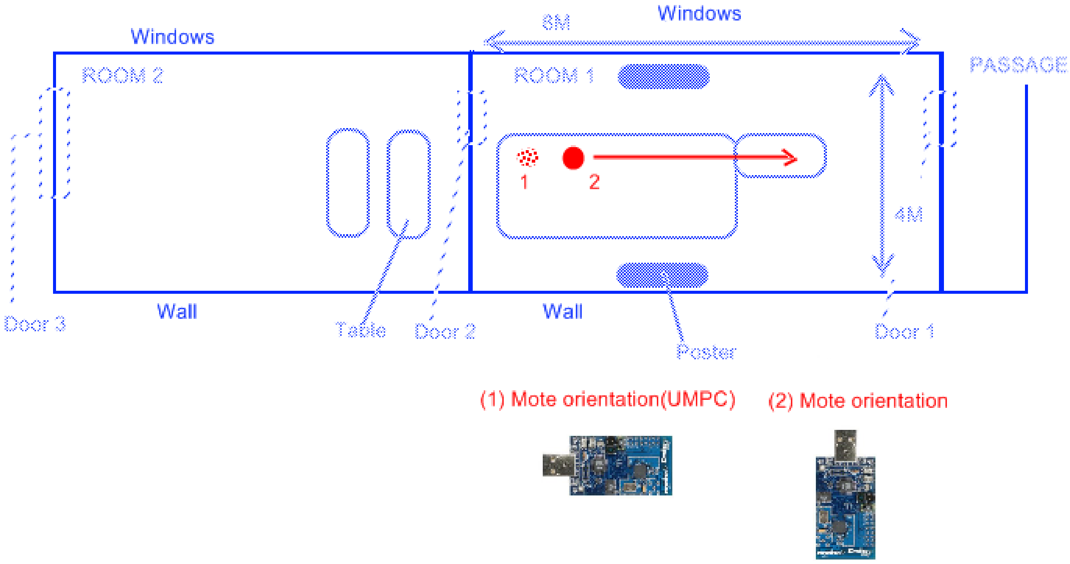


Result

Distance (measured from mobile mote)	Average of RSSI per 100 samples
1M	2
2M	-9
3M	-11
4M	-7
5M	-10
6M	-21
7M	-12
8M	Class is too small

(details, see measurements 27aug08 eblab.txt)

Meauserement 2: incomplete table, in the middle of the wall and the doors are open.



Condition

- Both motes stands on the table. Mote 1is combined with the UMPC and is fixed on the table. Mote 2 is moving.
- Both doors (doors 1&2) are open.

Result	
Distance(measured from mote 1)	Average of RSSI per 100 samples
1M	0
2M	-17
3M	-10(effect poster?)
4M	-16
5M	-20

Measurement 3: incomplete table, in the middle of the wall and the doors are closed.

Condition

- Both motes stands on the table. Mote 1is combined with the UMPC and is fixed on the table. Mote 2 is moving.
- Both doors (doors 1&2) are closed.

Result	
Distance(measured from mote 1)	Average of RSSI per 100 samples
1M	0
2M	-15
3M	-11
4M	-9
5M	-14

Conclusion

Measurement 3 is the same as measurement 2. Only mote 1 and mote 2 are changed of places. This shows that the RSSI signal change with a change of the topology.

Measurement 4: incomplete table, in the middle of the wall and the doors are open.

Condition

- Both motes stands on the table. Mote 1is combined with the UMPC and is fixed on the table. Mote 2 is moving.
- Both doors (doors 1&2) are open.

Result	
Distance(measured from mote 1)	Average of RSSI per 100 samples
1M	-11
2M	-13
3M	-17
4M	-19
4,5M-5M	-16

Measurement 6: Complete table, in middle of the wall and doors are open.

Condition

- Both motes stands on the table. Mote 1is combined with the UMPC and is fixed on the table. Mote 2 is moving.
- Both doors (doors 1&2) are open.

Result	
distance(measured from mote 1)	Average of RSSI per 100 samples
1M	-7
2M	Package error
2M(2^e time)	-10
3M	-13
4M	-18
4,5-5M	-18(0.5 before wall, reflection?)

Measurement 7: Complete table, in the middle of the wall through 2 classes (Effect of the wall)

Condition

- Motes are standing more in the middle of the table, it differs a litlle from the other measurements
- Mote2(Blindmote UMPC) is tight in Room1
- Mote1 is moved constantly untill in Room2

Result	
Distance(measured from mote 1)	Average of RSSI per 100 samples
1M	-14
2M	-13
3M	-13
4M	-22
4.5M(0.5M for wall class 1)	-11(effect reflection on the wall)
6M(l1M after wall class 2)	-18
7M	-17(door open and further away from the wall)
8M	-24

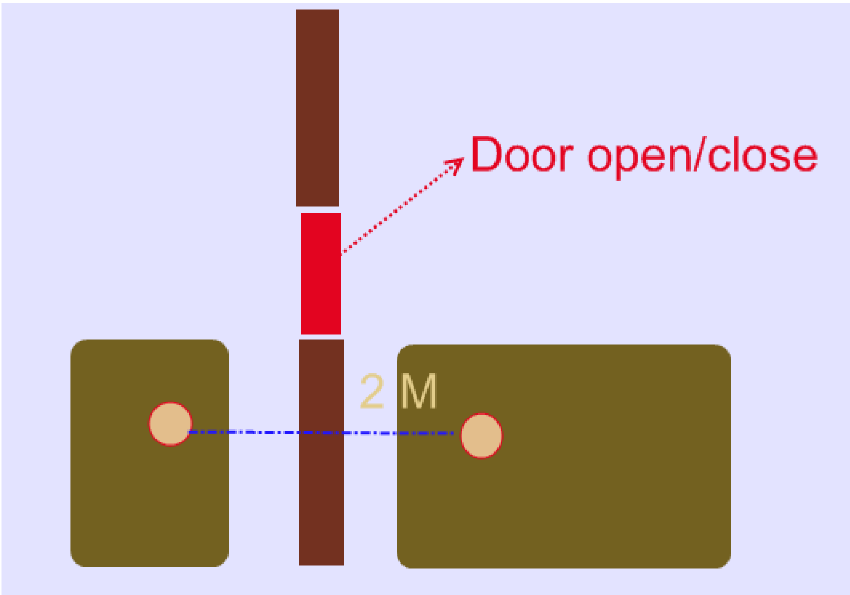
Conclusion

This measurement shows us that with indoor environment there are a lot of fadings. We have with 4,5 meters a sudden increase of the signal strenght. That has to do with the reflection of the wall.

Effect through the wall

We see at 6M a weaker signal than at 7M. That has to do with the fact that with 6M mote 2 is closer behind the wall. Almost every signal has to go through the wall. However with 7M, the mote is further away from the wall, and by this can reach certain sigals the other motes without going through the wall.

Measurement 8&9: Door effect and through the wall



<u>Condition</u>

- Both motes are standing on the table, mote 1 (UMPC) in Room 1 and mote 2 is in Room 2.
- The door is open/closed.
- Distance between two nodes is appr. 2M

Result	
Door open	-13
Door closed	-12

<u>Conclusion</u>

- The closing and opening of the door has an effect on the RSSI strenght, but this effect is limited.

Measurement 10: Outside measurement (Effect of an obstacle)

<u>Condition</u>

- Both motes are standing in the middle of the table.
- Mote 1 is combined with the UMPC and is fixed on the tbale. Mote 2 switches of table.
- Mote 1 standing on table 9.

Result

Table	Average of RSSI per 100 samples
1	-44
2	-38
3	-39
4	Battery flat
5	-35
6	-32
7	
8	

Note:

- *Is measured in the following order:table 3,2,1,6,5, 4(battery flat)*
- *Detais see measurement 10.tx*

<u>Conclusion</u>

The measurements are like in the theory. The further the motes are seperated from each other, how weaker the signal. Table 3&6 illustrates it.

Remark: Table 2&3

If we measure the distance, the distance between table 3 and table 2 is smaller. How is it possible that table 3 gets a weaker signal than table 2? The reason for this is that the signal has to go through table 6 (obstacle) to go to table 3, however at table 2 there is a line of sight connecting with our UMPC.