

Localization in Wireless Sensor Networks

YOUSSEF CHRAIBI



**KTH Signals
Sensors and Systems**

Master's Degree Project
Stockholm, Sweden 2005

IR-RT-EX-0523

Abstract

Similar to many technological developments, wireless sensor networks have emerged from military needs and found its way into civil applications. Today, wireless sensor networks has become a key technology for different types of "smart environments", and an intense research effort is currently underway to enable the application of wireless sensor networks for a wide range of industrial problems. Wireless networks are of particular importance when a large number of sensor nodes have to be deployed, and/or in hazardous situations.

Localization is important when there is an uncertainty of the exact location of some fixed or mobile devices. One example has been in the supervision of humidity and temperature in forests and/or fields, where thousands of sensors are deployed by a plane, giving the operator little or no possibility to influence the precise location of each node. An effective localization algorithm can then use all the available information from the wireless sensor nodes to infer the position of the individual devices. Another application is the positioning of a mobile robot based on received signal strength from a set of radio beacons placed at known locations on the factory floor.

This thesis work is carried out on the wireless automation testbed at the S3. Focusing on localization processes, we will first give an overview of the state of the art in this area. From the various techniques, one idea was found to have significant bearing for the development of a new algorithm. We present analysis and simulations of the algorithms, demonstrating improved accuracy compared to other schemes although the accuracy is probably not good enough for some high-end applications. A third aspect of the work concerns the feasibility of approaches based on received signal strength indication (RSSI). Multiple measurement series have been collected in the lab with the MoteIV wireless sensor node platform. The measurement campaign indicates significant fluctuations in the RSSI values due to interference and limited repeatability of experiments, which may limit the reliability of many localization schemes, especially in an indoor environment.

Contents

1	Background on Wireless Sensor Networks	4
1.1	Introduction	4
1.2	Our field of interest	4
1.3	Problem definition	5
1.3.1	Literature survey	6
1.4	Research approach	8
1.5	Thesis outline	9
2	Theoretical study of localization methods	11
2.1	Previous work	11
2.2	Range methods	12
2.2.1	Received Signal Strength	12
2.2.2	Time of flight	12
2.2.3	Using both: Calamari	13
2.3	Range-free methods	15
2.3.1	Do you hear me ?	15
2.3.2	Multi-hop	16
3	Development of a new algorithm	18
3.1	Purpose	18
3.2	The Bounding box	18
3.2.1	Description	18
3.2.2	Modelling	19
3.2.3	Results	21
3.3	Improvement	24
3.3.1	Results	24
3.4	The Circles Intersection	26
3.4.1	Description	26

3.4.2	Modelling	27
3.4.3	Results	27
3.5	The Use of Mobility	29
3.5.1	Modelling	30
3.5.2	Results	30
3.6	Conclusions	32
4	The IEEE 802.15.4 Standard	34
4.1	Network topology	34
4.2	Physical layer	35
4.3	Specificities for sensor networks	36
5	Moteiv Tmote rev.B	38
5.1	Introduction	38
5.2	Creating applications	39
5.3	Communication between motes	40
5.4	Emitting power	41
6	RSSI experiment	42
6.1	Overview	42
6.1.1	Limitations for RSSI approach	42
6.1.2	Application to Telos	43
6.2	Measurement campaign	43
6.3	Measured values: RSSI and LQI	44
6.4	Experimental settings	45
6.4.1	The application	46
6.4.2	Post-processing	48
6.4.3	The map	49
6.4.4	First results	49
6.5	Results	51
6.5.1	Normal environment	51
6.5.2	Quiet environment	53
6.6	Influence of batteries	54
6.7	Influence of emitting power	56
6.8	Influence of radio interference	57
6.8.1	Using the Aegis application	57
6.8.2	Due to other sources	59
6.9	Influence of the motes themselves	61

Chapter 1

Background on Wireless Sensor Networks

1.1 Introduction

Following the example of many technological developments, wireless sensor networks were an intense field of activity for military purpose. Today, smart environments are deployed everywhere, and sensor networks can be used in many different scenarios.

Wireless sensor networks are particularly interesting in hazardous or remote environments, or when a large number of sensor nodes have to be deployed. The localization issue is important where there is an uncertainty about some positioning. If the sensor network is used for monitoring the temperature in a building, it is likely that we can know the exact position of each node. On the contrary, if the sensor network is used for monitoring the temperature in a remote forest, nodes may be deployed from an airplane and the precise location of most sensor may be unknown. An effective localization algorithm can then use all the available information from the nodes to compute all the positions.

1.2 Our field of interest

The Automatic Control Group at the S3 Department at KTH is involved in many different areas of research, and is setting up a working group on

Wireless Sensor Networks. Several research topics started earlier this year, involving both PhD and master thesis students. From the current research fields, we can cite: Self-organizing Scheduling, by Pablo Soldati; Distributed Spatio-Temporal Filtering, by Manuela Cippitelli; Estimation in WSN, by David Pallassini; Change detection, by Victor Nieto.

My work comes as part of this group, the main field of my Master Thesis being the Localization Processes in those networks.

The scope being that a large amount of sensors (motes) are being deployed randomly, and only a few of them (anchors) are aware of their location (by GPS, for example). The purpose is to determine the best way to allow all nodes to deduce their own position.

1.3 Problem definition

The aim of this thesis is to develop an algorithm for localization of nodes in a sensor network. The algorithm should be distributed and executed in individual nodes; schemes that pool all data from the network and perform a centralized computation will not be considered. Since the algorithm should be run in individual sensor nodes, the solution has to be relatively simple, and demand limited resources (in terms of computation, memory and communication overhead). The goal is to be able to position nodes with a given accuracy, or to classify a nodes as being "non-localizable" (if it does not have enough, or accurate enough, information to perform the localization, for example). The performance of localization algorithms will depend on critical sensor network parameters, such as the radio range, the density of nodes, the anchor-to-node ratio, and it is important that the solution gives adequate performance over a range of reasonable parameter values.

One should bear in mind that localization in radio networks has been an intense area of research for quite some years, for military, civil (e.g. cellular networks) and sensor networks, so the development of a new algorithm will be far from trivial.

1.3.1 Literature survey

The paper [3] presents a global overview of the sensor networks. It describes the protocol stack as being divided in a physical, data link, network, transport and application layers; and gives characteristics and issues of each of them. It is focusing on enhancing route selection and lists some open research issues, as enhancing existing protocols or developing new ones with better scalability properties and increased robustness for frequent topology changes.

Another global survey of research issues in sensor networks, [11], points out several additional interesting aspects, such as the importance of pre-processing, as the devices have severe power constraints, limited storage, and since communication is the most expensive operation. (the use of micro-sensors and MEMS give the subsystem almost the same energy profile as the processor).

The localization methods could be divided into range methods, that would compute an estimation of the distances between two nodes, or range-free methods, that would not.

Range methods

The range methods exploits information about the distance to neighboring nodes. Although the distances cannot be measured directly they can, at least theoretically, be derived from measures of the time-of-flight for a packet between nodes, or from the signal attenuation. The simplest range method is to require knowledge about the distances to three nodes with known positions (called anchors or beacons depending on the literature), and then use triangulation. However, more advanced methods exist, that require less severe assumptions.

A relative complete description of ad hoc positioning systems is given in [10], comparing DV-Hop (Distance Vector), DV-Distance and Euclidian propagation methods. The first one computes an estimation for the size of one hop, while DV-distance measures the radio signal strength and is propagated in meters. The Euclidian scheme propagates the true distance to the anchor (a similar idea has been exploited in [7]). DV schemes are suitable in most cases, while Euclidian is more accurate, but costs much more communications.

MDS-MAP, an algorithm using connectivity information for computing the nodes' localization is presented in [12]. It consists of three main steps: First, using connectivity information to roughly estimate the distance between each pair of nodes. Then, multidimensional scaling (MDS) is used to find possible node locations that fit the estimations. Finally, it is optimized by using the anchors' positions. The first part of the algorithm can be enhanced by knowing the distances between neighboring nodes (even if with limited accuracy). It requires less anchor nodes and is meant to be more robust, especially if the nodes are quite uniformly deployed.

Range-free methods

A description of an ad-hoc localization system is given in [5]. Until now, the devices were individually tuned (built-in calibration interface or original long-life calibration). In SN, as a large number of sensors is used, that cannot be the case. The authors present *Calamari*, an ad-hoc localization system they developed that also integrates a calibration process. Regarding localization, it uses fusion of RF received signal strength information and acoustic time of flight.

There is an interesting definition of a distributed algorithm for random WSN in [6]. The minimal density of known nodes is presented. The main objective of their algorithm is to broadcast a request ("*Do you hear me ?*") and compute the estimated localization by the interpretation of the answer of all the known nodes.

A related method, called APIT, is suggested in [9]. In this approach, nodes test if they are inside or outside a triangle (there is one triangle for every combination of three distinct beacons), and then attempts to reduce the area as much as possible. Even if the method will produce a localization region, rather than a unique estimate, the authors of APIT claim it to be the best known range-free algorithm. One main drawback is that it needs a big anchor-to-node ratio and anchor number.

Challenges and improvements

The authors of [4] describe the history of the research in SN, but more interesting is the overview on the technical challenges and issues is presented, from which we can cite several relevant items: WSN working in a harsh environment; the knowledge of the network (at least the neighbors); the network control and routing; querying and tasking (should be as simple and intuitive as possible); and also security issues (low latency, survivable, low probability of detecting communications, high reliability).

The influence of noise can be important, as [7] shows (flip and flex ambiguities). To minimize it, the authors define Robust Quadrilaterals and Clusters. But the computation complexity increases as it is extended to large-scale WSN, which is a big inconvenient.

Localization schemes that exploit the additional information that can be obtained when some nodes are mobile are described in [8]. Three schemes are possible: static nodes - moving seeds, moving nodes - static seeds, or both moving. An interesting local algorithm is presented here, based on changes of neighborhood, using the principle of *insider* and *outsider* nodes).

A comparison of Ad-hoc positioning, Robust positioning and N-hop multilateration is given in [13]. The three algorithms have common structure: determining of the anchor-to-node distance, deriving from this a position for each node, refining the estimations using positions and range of neighboring nodes. As one expects, their conclusion is that there is no absolute algorithm to be used, the choice depending on the required utilization's conditions (range errors, connectivity, ANR...).

1.4 Research approach

For studying the localization issue in sensor networks, different approaches could be imagined. The first division is between range and range-free algorithms. The thesis will present a discussion of the main range techniques in sensor networks, as the use of the received signal strength or time of arrival, and of some performing algorithm based on them (Calamari, for example).

Their purpose will be exposed, as well as an overview of their main advantages and drawbacks for sensor localization. In this context, we will put particular focus on limitations due to current sensor network technology.

An other aspect is the use of a static or dynamic model. Static algorithms, also called geometrical, are taking advantage of the configuration of the network, while dynamic models will use an external item to be able to localize each member of the network.

The evaluation of various algorithms and methods is done by extensive simulations, studying the accuracy and repeatability of the results. An extra study about different parameters that could affect the measurements accuracy could describe their different effects as for example to illustrate the case of a sensor mote running in low battery.

Each of range and range-free techniques could be used, depending on the network situation. Even if we think that range-free is the best way according to our motes and our environment, this thesis could not be complete without a more extensive study of range methods. This could then be done by creating a real mapping of the received signal strength in a specific environment (the floor of the Automatic Control Group, for example) and then analyze those results. Mapping the acoustic time of flight on the same way could be interesting too, but this cannot be achieved on our sensor motes.

1.5 Thesis outline

The Thesis will start by a theoretical study of the localization methods, presenting a state of the art of the different technical solutions and selecting the most promising algorithms, in Chapter 2.

We will then elaborate further on one of the ideas in Chapter 3 and develop a simple algorithm that extends the method. We study the performance of the approach in different conditions and compare its efficiency to other methods.

As the Department's research activities are based on Telos motes, Chapter 4 presents the IEEE 802.15.4 standard, while the motes themselves are described on Chapter 5.

Experiments using the received signal strength are described in Chapter 6, in order to study the practical feasibility of some of the proposed algorithms.

Chapter 2

Theoretical study of localization methods

2.1 Previous work

Schemes for localization in WSN have been developed in the last 20 years, mostly being motivated by military use. Numerous studies have been performed since then for a civil use [11]. Researchers have pointed out the influence of noise on the localization process [7], and the importance of various system parameters on the accuracy and efficiency of the localization process [6], but there is no consensus of a single best algorithm for localization in sensor networks. This indeed depends on the environment and the specifications of the used motes.

Others showed that all the variations (transmitter frequency, acoustic hardware, etc...) lead to errors up to 300 % in the distance estimates and thus insisted in calibration issues instead of developing specialized hardware. A combination of the different range techniques has also been done, and led to *Calamari* [5], a quite successful algorithm.

The use of mobile nodes has also been studied [8], should it be used for beacons or for nodes. A very interesting algorithm that *simulates* mobility has been developed [9], and presents good results, even if it needs a large amount of nodes. Indeed, localization is still a very active and largely open field of research [3].

2.2 Range methods

All algorithms can be classified in either range or range-free methods. The range methods being the first ones to appear. Their principle in localization is mainly to estimate the distances between node pairs, and then to compute the position of individual nodes in the global network. Triangulation is for example the most basic approach for computing the position. Before giving the details of different range methods, we will discuss how one can estimate inter-node distances.

2.2.1 Received Signal Strength

The energy of the radio signal, viewed as an electromagnetic wave, decreases as it propagates in space. By knowing the original emitted power and comparing it to the received signal power, one can estimate the attenuation g and deduce the distance via, for example, a free space path-loss model:

$$g = d^{-\alpha} \quad (2.1)$$

In this scheme the exponent α is around 2 in an open-space environment, but its value increases if the environment is more complex (walls, etc.) or less suitable for radio waves (metallic devices...).

Another issue is that there is no unique path from the transmitter to the receiver. Any reverberations of the signal will influence the received strength, so it has to be measured at the appropriate moment. Some consider the first peak, whereas others prefer an average of the first periods.

2.2.2 Time of flight

When the environment is supposed coherent enough for the propagation of a signal being at constant speed, knowing the speed and measuring the time of propagation will give an estimation of the distance.

This is the basic principle of the Time of Flight (ToF), that could also be applied to a radio signal. Since the propagation speed of radio signals is very high (indeed, equal to the speed of light), time measurements must be very accurate in order to avoid large uncertainties. For example, a localization accuracy of 1m requires timing accuracy on the level of 3,3 nanoseconds.

In the case of the Global Positioning System, a synchronization of the atomic clocks in the satellites gives a great accuracy (thus depending on the clock of the receiver), but in the case of wireless sensor networks, the achieved accuracy is very poor: the Telos motes used at the Automatic Control Group have a time stamp of the radio packet with an accuracy of 1 millisecond.

Using an acoustic signal will decrease the propagation speed, and thus increase the accuracy. With a precision of 1 ms, the localization accuracy is 35 centimeters. Unfortunately, the motes we are using were not primarily designed for localization purposes, and have no acoustic transceivers nor receivers.

The NADA department at KTH is conducting researches with embedded sensor boards (ESBs), and their motes (manufactured and sold by ScatterWeb GmbH [21]) have both beepers and microphones. They could be used with acoustic time of flight algorithms, but we have been unable to conduct measurements on the ScatterWeb platform. One restriction could be on how much they are sensitive to their own signals.

An advantage of acoustic time of flight is the multipath avoidance, as the signal suffers less interaction with its reflections.

Both time of flight and acoustic time of flight are more expensive than received signal strength (hardware, power), but are more accurate and almost computation free.

2.2.3 Using both: Calamari

According to the description in [5], Calamari is a good compromise and a solution to the calibration problem. The authors showed that normal variations (in, for example, transmit frequency, acoustic hardware, etc) between sensor nodes from the same manufacturer may lead to an error up to 300% in the distance estimates. Although these errors could potentially be remedied via higher tolerances on hardware components, calibration would certainly be a much more cost efficient approach.

A traditional calibration technique would be to map the device response to the desired one. But this procedure has to be performed for every *pair* of

devices, thus it is order n^2 . This *pairwise calibration* is too expensive.

A first solution is **iterative calibration**. One transmitter is said to be the reference, and all the receivers are calibrated, and the other way round. The problem is that it is valid for a single frequency. As the frequencies may vary from a transmitter to another, this is still a valid scheme for a single pair, in a way.

Mean calibration avoids the pair problem by a simple assumption: the variations in the devices are normally distributed. Each receiver is then calibrated using all transmitters, but transmitters are not calibrated...

Joint calibration is used in Calamari, to calibrate each device by optimizing the *overall system response*. The ToF estimation is affected by hardware issues, mostly Bias (time for starting oscillating) and Gain (volume of the emitter, sensitivity of the receiver).

The sensor model is then:

$$r^* = B_T + B_R + G_T \cdot r + G_R \cdot r \quad (2.2)$$

Frequency and sensor orientation also affect the output, but are considered as included in the error term. This relation for each sensor pair will give $4n$ variables and $n^2 - n$ equations. There is no way to solve each of them separately (i.e to decide if the error is due to the transmitter or the receiver), but it can be solved globally. Detailed explanations and matrices are found in [5], section 5.2.

All the calibration process was with known distances. A proposition of **Autocalibration** for a completely uncontrolled environment is to take advantage of symmetric pairs ($d_{ij}^* = d_{ji}^*$) and triangle inequality ($d_{ij}^* + d_{jk}^* - d_{ik}^* \geq 0$, if i, j and k are connected).

If some anchor nodes are used, the known distances can replace the estimates in the above equations, thus reducing the estimation error.

In [15], there are more details about RSSI (section 6.2) and ToF (6.3), and just a few information about the mixed use in Calamari:

TinyOS' current default radio protocol measures signal strength with every sent message. And the message is time stamped with micro second accuracy. Both radio and acoustic messages are sent simultaneously. When the acoustic impulse is received, the processor is topped with a time stamp

with micro-second accuracy. The difference between the two stamps, multiplied by the speed of sound gives the distance. Technically, it seems that RSSI is not used, but only the time stamp included in it.

2.3 Range-free methods

Contrary to the first ones, those methods never compute the distances to the neighbors. They use hearing and connectivity information to identify the nodes and beacons in their radio range, and then estimate their position.

2.3.1 Do you hear me ?

This idea of only using the information of the immediate neighbors fits perfectly the distributed approach of the localization problem. In those type of schemes, every node only uses direct communications to refine his position estimates, and when it succeeds to achieve a given accuracy, it broadcasts the result. The big advantage is that it saves a lot of traffic, but an overload of the radio channels can occur. This has to be carefully studied, and the rules for priority clearly established. Another drawback is the fact that those techniques usually require a great amount of nodes.

APIT

The APIT idea is to divide the environment into triangles, given by beaconing nodes. An individual node's presence or absence in each of those triangles will allow to reduce the possible location area. This goes until all the possible sets are exhausted, or the desired accuracy reached.

The APIT algorithm is then ran at every node:

1. Receive locations from n anchors.
2. For each possible triangle, test if inside or not.
3. If yes, add it to the InsideSet.
4. Break if accuracy reached.
5. Estimate position as $CenterOfGravity(\cap T_i \in InsideSet)$.

For testing if the node is inside or not a triangle according to the Point-in-Triangle (PIT) test, it needs to move. To cope with situations where nodes

are static and unable to move, an *Approximate* PIT test is defined according to:

*If no neighbor of M is further from/closer to **all** three anchors A , B and C **simultaneously**, M assumes that it is inside triangle ABC . Otherwise, it is outside.*

This is of course subject to errors, especially if the node is close to one of the network's edges, or if the neighbors have an irregular placement. The authors of [9] have performed extensive simulations and claim that the error has never exceeded 15 % (on their particular scenarios).

To minimize such errors, there is an aggregation of the algorithm's results, not only an intersection. A grid represents the possible location for the mote. Initially filled with zeros, it is incremented for every triangle that had a positive APIT test, and decremented for others. The area with maximum overlap has then the highest numbers, and its center of gravity will be the estimated position.

An important aspect of this solution is that APIT uses indeed signal strength, but not as an approximate for a distance. It just assumes that signal strength decreases monotonically with the distance (usually valid). Thus it is used to *compare* distance, and APIT is still a range-free algorithm.

2.3.2 Multi-hop

Multi-hop methods are mainly range-free, but can also use estimation of the distances. Their purpose is to compute a connectivity graph, and then trying to make it fit the known positions as good as possible.

Multi Dimensional Scaling

In a large sensor network, Multi Dimensional Scaling (MDS) only uses connectivity information, *i.e.* which nodes are within communication range of which others. The process has three steps:

1. Rough estimation of the distance between each possible pair of nodes.
2. MDS to derive locations fitting the estimated distances.

3.Optimization by taking the known positions into account.

The system is modelled by a connectivity graph, the edges having the value 1 (if the distances are known, the values are used instead). This gives a symmetric matrix, which is run in a classical all-pairs shortest-path algorithm.

The resulting distance matrix is used in classical MDS, and gives a relative map locating each node.

Linear transformations (scaling, rotations, reflections, translations) are used to fit the anchors' estimated positions to the correct ones, and perhaps all other known positions, if any.

There are many types of MDS techniques: metric/nonmetric, classical/replicated, weighted, deterministic/probabilistic. Classical MDS, where the proximities are used as being distances, seems to be the best choice in this issue [12]. The Euclidian distance has then to be as close as possible to the proximities (least squares).

N-Hop Multilateration

Multihop multilateration's technique [16] is aiming to give to give nodes that are several hops away from beacons the possibility to collaborate in finding better position estimates. By allowing this type of collaboration, the ratio of beacons to nodes can be decreased.

The algorithm could be centralized or decentralized, see [16] for a detailed account of the distributed version, fitting best sensor networks (communication costs distributed, accepts node failures).

Chapter 3

Development of a new algorithm

3.1 Purpose

As the Telos motes used at the Automatic Control Group at S3 have no acoustic transceivers, focus is being put on geometrical methods for localization. Using the connectivity information, each node tries to determinate its position. Due to the network configuration, part of the nodes will not be able to achieve such a goal, but they will minimize the uncertainty in their estimated position.

3.2 The Bounding box

3.2.1 Description

A first implemented approach was the bounding box. Each node listens to the beacons in his neighborhood, and collects their position. As the beaconing range is known (noted br), the node applies a simple algorithm: if a beacon positioned at (x_B, y_B) is heard, then the node's coordinates fulfill the following relations:

$$x_N \in [x_B - br; x_B + br]; y_N \in [y_B - br; y_B + br] \quad (3.1)$$

The position of the node is guaranteed to lie in the intersection of the bounding boxes corresponding to all the beacons within the radio range. This set itself is a box, whose minimum and maximum values are computed by iterating the process upon all the beacons heard: at each stage, the new boundary values are compared to the previous ones, and the smallest set is kept. Finally, the center of gravity of this last intersection set is computed, and said to be the estimated position, as seen in Figure 3.1.

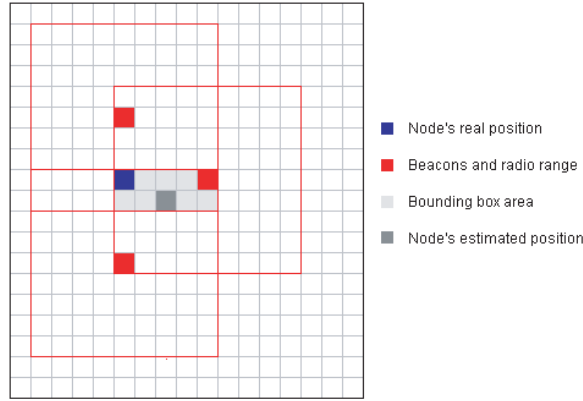


Figure 3.1: Principle of the bounding box

3.2.2 Modelling

Such an algorithm, as all the geometric approaches of the localization problem, requires a large number of nodes and beacons. As we only have a dozen of nodes, this could not be practically implemented on our motes. All the research is then made with a Matlab model of the system.

Given the size of the network, and the respective density of the nodes and the beacons, two grids are created (one for nodes, the other for beacons) and the motes randomly deployed. As the complete system is made by the two grids together (Figure 3.2), a quick check ensures that there are no locations with both a node and a beacon. If so, the node is removed from the grid.

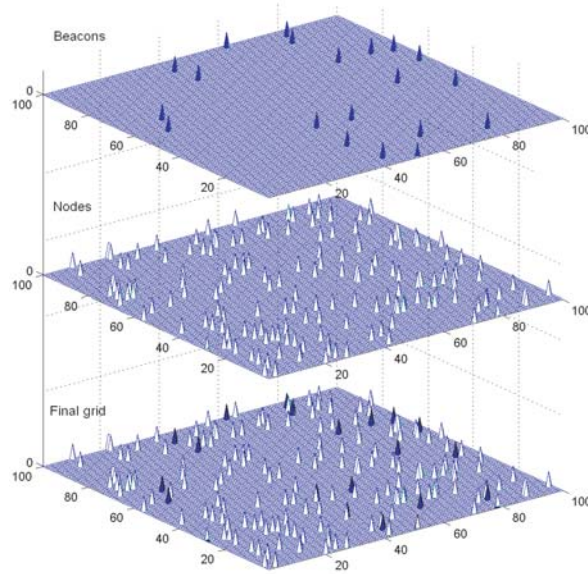


Figure 3.2: Creation of the system

The localization process will be ran at each node, and consists of 5 different steps:

1. A local grid is created around the node's position. This grid is as big as the beaconing radio range, that means that the node cannot hear the beacons located outside this set.
2. A check of the beacon's grid lists all the heard beacons and their position in a table **beaconlocation**.
3. For all those beacons listed, the bounding box relation is applied, thus reducing the location possibilities' set.
4. After going through all the beacons, if this set is equal to one, the node successfully localized itself. If not, the size of the set is kept as an uncertainty value to allow the study the efficiency of this method, and the position is estimated as being the center of gravity of that set.
5. The **nodeestimate** grid is updated with the result, keeping track of the error parameters.

Once those five steps achieved for all the nodes, the **nodeestimate** grid is complete and can be compared to the initial node grid. Several statistic

values are available, such as the number of the correctly localized nodes, the average and maximum uncertainty volume, etc.

3.2.3 Results

As the network is randomly generated, some particularities can appear, such as isolated nodes or beacons in a corner. A node on the edge of the grid will hear half less beacons than another one in the middle, and thus this could decrease the overall efficiency. To get rid of such particular cases, every simulation is ran several times (5 to 10), and a mean value is taken out of the results. Also, even if the network is generated by the value of the density, the number of nodes and beacons can fluctuate from one simulation to another. For comparing the results, we then express some of the statistics in percentages.

First simulations were designed for evaluating the impact of the network's settings (density, radio range, size). Due to the many involved parameters, it was not feasible to make them all vary at the same time. The difficult part of that study was then to evaluate them by pairs or triplets, and draw general conclusions of the global interactions.

For a given radio range, the proportion of correctly estimated nodes is increasing with the density of beacons. For a fixed network density, increasing the radio range allows the number of correctly estimated nodes to grow, but after a certain distance, this is not significant anymore.

Deciding on an appropriate radio range is a matter of scale. For a network represented by a 100 by 100 grid, a radio range of 16 is about the same as a radio range of 8 in a 50 by 50 network. The only difference concerns the accuracy. One unit is the best accuracy we can achieve, so if a radio range of 12 represents the 125 meters of allowed propagation (characteristic of the Tmote motes used), then the accuracy of the model would be 10 meters. Using a smaller grid will make computations easier at the expense of the accuracy.

Keeping the fact that 1 unit represents 10 meters, our grid then represents a 1 square kilometer network, and a node is said to be correctly localized if

the model manages to compute in which 10 by 10 meters square it is positioned.

A basic configuration is decided, to be able to evaluate further improvements in terms of accuracy. Given by density inputs of 0.30 for the nodes and 0.08 for the beacons, the network has approximately 2400 nodes and 770 beacons. The Bounding box approach localizes from 30 to 35 percents of the nodes. Considering a radio range of 12 units, 32% are perfectly localized (accuracy=1). The table shows that 52% are considered as "false". This means that the 16 other percent were nodes that were not able to reach an accuracy of 1, but are correctly localized. This is due to the computation of the center of gravity of the uncertainty zone. Those nodes were most likely to have a quite good accuracy (maybe 2, 3 or 4 units, but those values are not reachable in our model) and the center of gravity of this zone was matching the correct location.

		OK	False	Others	Global acc	Real acc
Brt:	11	30.94	51.67	17.39	8.17	18.55
	12	32.18	51.43	16.39	9.43	21.54
	13	33.51	51.23	15.26	10.81	23.92
	14	34.52	50.69	14.79	12.42	27.26

Figure 3.3: Bounding box results

The non correctly localized nodes give an idea of the performance of the algorithm, in this particular situation. According to the detailed results, those nodes have a mean uncertainty zone of 13.8 units. The mean height is 3.9 units, the mean width 3.8. Looking at the more precise accuracy tables and figures, we can extract the data of Figure 3.4 and see that some of the nodes have an uncertainty zone greater than 100 (the biggest one here being 168), whereas most of them have a value smaller than 20.

This illustrates one big disadvantage of this method. As the final accuracy is the product of the reached accuracy in height and width, the result will be quite bad if one of them is bad. The table of detailed results shows that many nodes have a perfect accuracy in one dimension (1 or 2), but a poor one in the other (14 to 17) leads to a final bad result.

Another rating for the accuracy can be given by spreading that accuracy

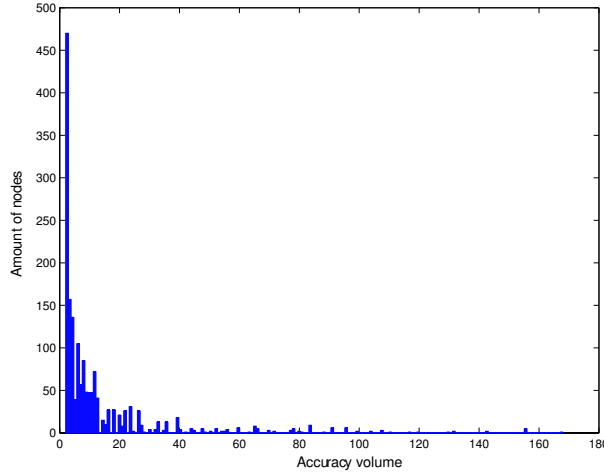


Figure 3.4: Reached accuracy using the bounding box

value over all the nodes in the network (even the correctly localized). This acts as for the "global effectiveness" of the system, and gives us a global accuracy of 9.7. Those values will be used for later comparisons.

One density ratio will not lead to the same result in terms of efficiency. This is due to two different aspects.

The first one is inherent to our Matlab model. As the beacon and node grids are designed separately, if a node and a beacon are located in the same slot, the node is removed. This superposition is higher for a bigger global density. While a 0.10 input each density will create 950 beacons and 860 nodes (degradation: 0.9), a 0.20 input will result in 1900 beacons and only 1400 nodes (a 0.73 degradation). This could be taken into account by some try-and-correct simulation of the network to design the good one: a 0.20 and 0.16 inputs will respectively create 1520 nodes and 1500 beacons.

The second aspect is linked to the localization process, especially margin effects. As the node listens to its neighbors, a node present on the side of the grid will have less neighbors to listen to, and thus will have a poor accuracy. Increasing the global density will increase the number of nodes close to the edges, but will also increase the presence of beacons. This results on more beacons available for the edge nodes, and then in a better accuracy. Out of

860 nodes, 41% are correctly localized with 950 beacons at a radio range of 12. This proportion grows to 55% for the same radio range but 1520 nodes and 1500 beacons.

3.3 Improvement

As the amount of perfectly localized nodes is not negligible, they could be used as *additional beacons*. This means that, when a node computes its position, it can also use the other normal nodes in his radio range. If a node detects that it is perfectly localizable, it will start acting as a beacon, broadcasting its position information, and thus be included in the computations. As the node emitting radio range is smaller than the beacon's, that node used as an extra beacon will provide a much smaller possible positions set, thus increasing significantly the accuracy.

For doing this, a beaconing mode has to be implemented on the nodes. If the node is able to compute its position, it then switches to this beaconing mode and keeps broadcasting its position. When another node receives this information, it also needs to know that the sender is a normal node and not an original beacon, to take into account the good radio range. As this radio range can differ, the messages sent should have the both information: position and radio range.

The main drawback of this method is linked to the resolution method. The grid is checked sequentially, so that means that, when the node located at (i,j) is computing its position, the only nodes' information that can be used are the ones from nodes already computed: those located at $(k < i, *) \cup (k \leq i, l < j)$, see Figure 3.5.

3.3.1 Results

The listening feature requires a new setting: the node radio range. Smaller than the beacon radio range, it is modelled by different values, from 3 to 6. Of course, the biggest the value, the greatest amount of corrected nodes. The most significant values are 4 and 6, as being the third and the half of our default beacon radio range. The following table (Figure 3.6) shows the

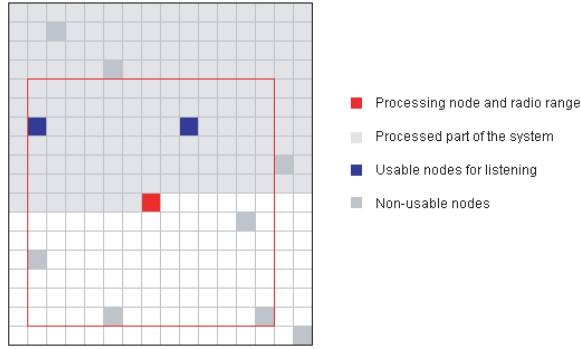


Figure 3.5: Restrained listening possibilities

results of the listening procedure.

		OK	Corrected	Total	Global acc	Real acc	False
Br.	11	30.59	10.68	41.27	8.17	18.55	42.44
	12	32.54	9.75	42.28	9.43	21.54	42.28
	13	33.75	8.45	42.20	10.81	23.92	43.77
	14	34.48	7.60	42.09	12.42	27.26	44.30
Nrr.	3	32.90	7.47	40.37	10.26	22.30	44.51
	4	32.47	8.83	41.30	10.28	22.73	43.22
	5	32.88	10.00	42.88	10.17	23.12	42.40
	6	33.10	10.18	43.29	10.12	23.12	42.17

Figure 3.6: Listening approach results

The "corrected" column stands for the additional amount of well estimated nodes, after listening to the neighbors (expressed as a percentage of the total amount of nodes). The first remark is that the corrected number increases with the node radio range. This was expected, and is following exactly the first results with the beacon radio range.

A second result was unexpected: this amount decreases when the beacon radio range increases. This seems to be due to the fact that low beacon radio ranges leaves more nodes with a small uncertainty domain. Applying the listening approach to those nodes will reduce those domain until many of

them reach a perfect accuracy.

Overall, if we sum those results, the largest amount of localized nodes is around 44% for a 12 and 6 setting (the table shows mean values for every Brr and Nrr setting). Extending those simulations to larger beacon radio ranges confirms this maximum, whereas for the node radio range, the corrected number increases slightly (45% for 12 and 12).

The percentage of still false estimated position is around 42, which means that again around 14% of the nodes were not able to perfectly localize themselves, but happened to estimate their position as being the correct one.

3.4 The Circles Intersection

3.4.1 Description

The radio signal having a limited propagation range, it will be heard from any node inside a circle centered on the emitting beacon. The Bounding Box approach was as to estimate those circles as being squares. In this second method, we are closer to the reality with using circles. But this is still an ideal model, as the real pattern suffers from distortions due to obstacles (free-space propagation) and the antenna's radiated pattern. However, the circle approach is closer to the reality than the bounding box. The node will here compute the intersection of all the broadcasting circles of the nodes he hears, and the rest of the process will be hardly the same.

A first idea was to mathematically compute the intersection of those circles, then adapt the result to our grid. This would have been the more precise way to compute the intersection of the circles, but is quite complicated. Furthermore, we do not need such a precision as we have a discrete grid.

The effect of that discretization can be seen in Figure 3.7, which depicts the geometrical discretization of the two shapes. Depending on the radio range, the difference between circles and squares can be quite different, and we notice that it is negligible for the sizes corresponding to the node radio range.

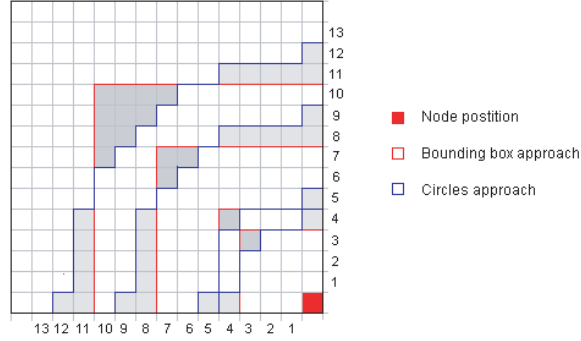


Figure 3.7: Comparison of the two shapes

Despite this, as the circles intersection is already implemented for the beacons, it is still interesting to apply it to the nodes as well.

3.4.2 Modelling

The modelling is similar to the one of the bounding box algorithm. For the circle intersection, we proceed by iterations, first intersecting two circles and then intersecting the result with another circle. Once the beacon's location listed, an additional local grid is created with the broadcasting range of the beacon and this will be the temporary grid for all the intersections.

Once all the beacons have been taken into consideration for a node, the center of gravity is computed. In some cases, as shown on the Figure 3.8, the final set can be non-compact, so the center of gravity can be outside the set. Choosing this as the estimated location is a problem: as it is outside the intersection set, it is certain that it will be a false location estimation. In such cases, the position inside the set being the closest to the center of gravity is the one chosen as being the estimated position for filling the `nodeestimate` grid. Similar statistic evaluation is then made on that grid.

3.4.3 Results

A first coarse execution gives that 36 to 44 percent of the nodes are correctly localized, that is 6 to 10 points more than the bounding box. Consider-

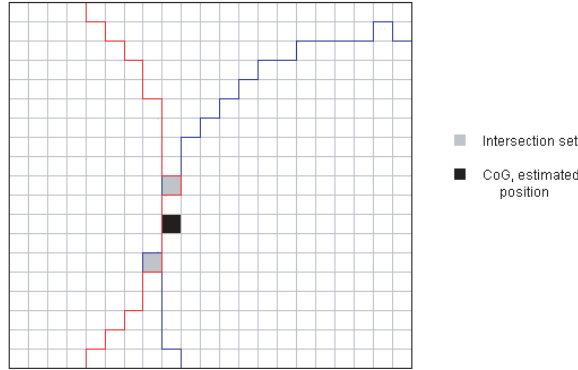


Figure 3.8: An example of non-compact intersection set

ing a radio range of 12 units, 39.5% are perfectly localized (instead of 32), and 46% are uncorrect. That leaves 14.5% of the nodes that are correctly positioned even if the process did not reach perfect accuracy (previously: 16).

Going further in the comparison shows that the mean accuracy volume for the non-localized is 10.8 units, which is, instead of 13.8 a significant progress. Only a few nodes have a high accuracy volume, most of them being under 20, Figure 3.9.

This circle intersection cannot suffer of low accuracy in one dimension, but, as the bounding box, isolated nodes will have a poor accuracy.

Spreading the accuracy results over the whole network gives an global effectiveness of 6.8%, better as expected than the previous 9.7%.

If we combine circles intersection and the listening to the neighbor nodes (radio range = 6), the improvement is of about 16.5% of the nodes being corrected, instead of 10.5%. A bigger improvement is on the estimated nodes that are fitting their real position, 23.3% instead of 6%. That leaves only 17% of the total amount of nodes that are not correctly localized, shown on Figure 3.10.

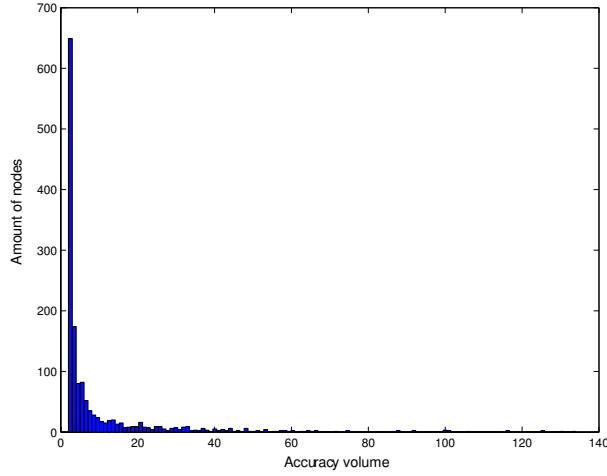


Figure 3.9: Reached accuracy using the circles intersection

3.5 The Use of Mobility

As previously seen in [8], having a moving part in the network can increase the accuracy of the overall localization schemes. Three schemes are possible: static nodes - moving beacons ; moving nodes - static beacons ; both moving.

For our purpose, the first one seems to be the most suitable one. In a big sensor network, an additional beacon can be fixed on a mobile robot, for example. This robot would be moving around, although following a predetermined path or randomly. Broadcasting regularly its position, it will give additional information to the nodes. Of course, those nodes have to be aware of the existence of the moving beacon, and be able to wait for that signal before ending their localization process.

As it is, the moving beacon (or maybe more than only one) acts like an additional beacon mesh that completes the initial one (Figure 3.11).

Using mobile beacon(s) provides many extra beacon locations, so it is also a way to reduce the number of beacons initially present. This issue has to be solved specifically in each different case of the use of a WSN. In certain conditions and scenarios, the use of a mobile device can be more difficult, or completely impossible.

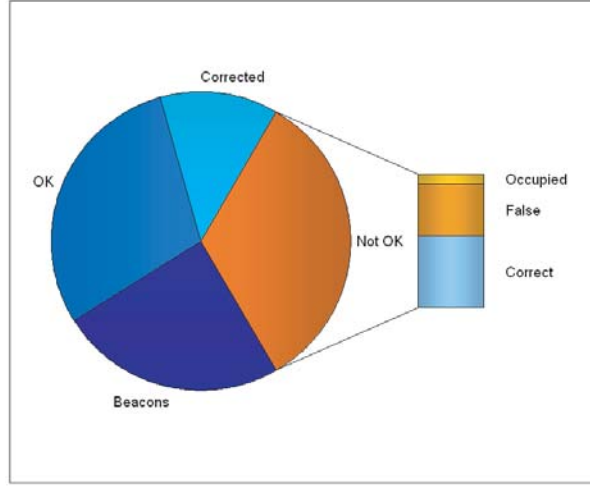


Figure 3.10: Global results for the circles intersections

3.5.1 Modelling

Following the presentation showed previously, the mobile beacon is modelled as being an additional grid of many beacon locations.

The scheme needs three different inputs to prepare the grid: original position (x_{mov}, y_{mov}) , moving beacon radio range, position increments (x_{inc}, y_{inc}) . If the simulation asks for using the moving beacon positions, then, while each node is checking for the heard beacons, it will check in both the **beacon** and **movingbeacon** grids. (Note: the two radio ranges are in general different.)

3.5.2 Results

The mobility alone is able to correctly localize around 7% of the motes, as it can be seen in Figure 3.12, but this is less than what we had before with the listening to the neighbor nodes. However, we notice that even if the amount of correctly localized motes is smaller, the accuracy volume of the others is significantly decreased.

The discussion is on how much the mobile beacon should move. It appears evident that if the smaller the movement step of the mobile beacon is, the higher the influence will be. However, one cannot aim for a mobile beacon having a step as small as the grid unit. This would be exactly the

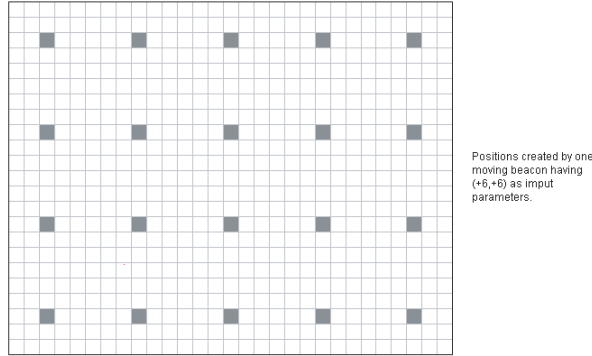


Figure 3.11: Numerous positions added by the moving beacon

		OK	Corrected	Total	Global acc	Real acc	False
Brc:	11	30.50	7.48	37.98	3.64	7.49	44.54
	12	32.23	7.91	40.14	3.51	7.28	43.77
	13	34.06	7.49	41.54	3.47	7.37	42.46
	14	34.71	7.05	41.76	3.52	7.28	43.56
Nrr:	3	33.06	7.42	40.48	3.54	7.36	43.57
	4	32.93	7.51	40.44	3.53	7.39	43.64
	5	32.73	7.48	40.21	3.53	7.32	43.79
	6	32.76	7.52	40.28	3.53	7.35	43.59

Figure 3.12: Influence of a beacon moving by 6 units

same as having a beacon at every single position, and thus all the normal beacons become useless.

Looking for the appropriate step size of the beacon can be a hard task, and depends strongly on the environment. In some cases, a regular movement can be applied, in some others it cannot. Through our simulations, we discovered that, in the case of a regular movement mesh, the effects are more important when the movement size is one half or one third of the beaconing range.

Thanks to the mobile beacons, more nodes are perfectly localized, and can then act as beacons themselves. For this reason, it is obvious that, when both listening and moving methods are used, the moving beacon should be the first applied.

In such a case, and with a beaconing moving by 6 units, we could reach significant results as shown on Figure 3.13.

		OK	Corrected	Total	Global acc	Real acc	False
Brr:	11	30.94	18.48	49.41	3.06	7.98	34.46
	12	33.07	18.27	51.34	2.96	7.73	34.02
	13	33.86	16.61	50.47	3.01	7.71	34.73
	14	34.29	15.64	49.93	3.09	7.49	36.67
Nrr:	3	33.01	16.13	49.13	3.06	7.66	35.62
	4	33.12	17.38	50.50	2.98	7.64	34.38
	5	33.01	18.26	51.27	2.97	7.75	34.04
	6	33.03	17.22	50.25	3.13	7.86	35.57

Figure 3.13: Using a moving beacon (+6) and listening to the neighbors

Around 66 % of the motes are now perfectly localized (50 % reached the desired accuracy, and the 16 other percents revealed to be correct). And the accuracy of the last third of nodes significantly decreased.

3.6 Conclusions

From the initial simple bounding box algorithm, we saw that this range-free method could be quite efficient on wireless sensor networks, providing the fact that the beacon-to-node ratio is quite important. The use of circles gives more accuracy and is closer to the reality, but this requires more resources, more computation and harder implementation.

Both ideas could be improved by using mobile beacon and the neighboring nodes. As we saw that both techniques significantly improve the localization process, the algorithm to be applied at each node is then finally :

```

for each node
  Go through all the beacons heard
  if accuracy  $\neq$  ok
    use the moving beacons
  if accuracy still  $\neq$  ok
    use the neighbor nodes

```

```
if accuracy == ok
    localizable = True
else
    localizable = False
accuracy.value = size.set
end
```

In which, the relevant accuracy to be reached has to be decided according to the circumstances and the environment. Here we were always looking for an accuracy of one grid unit, but one must keep in mind that this is related to a real distance. The localization of a sensor mote inside a network monitoring the temperature and humidity of a forest should not be as accurate as the position of a mobile robot inside a factory.

Chapter 4

The IEEE 802.15.4 Standard

Before going further to our experiments in Chapter 6, it is necessary to have a closer look to the IEEE 802.15.4 standard [1], which defines the specific requirements for the Wireless Medium Access Control (MAC) and Physical Layer (PHY) for Low-Rate Wireless Personal Area Networks (LR-WPAN).

The main points of designing a LR-WPAN is for the need of low-rate, low-power and low-cost wireless applications. Easy to install, it provides reliable data transfer for short-range operation.

In such a network, two types of devices can be involved: a full-function device (FFD) or a reduced-function device (RFD). The FFDs can be operating as coordinators or normal devices. The RFDs are meant for extremely simple applications that do not need sending large amounts of data. Moreover, it can only be connected to one FFD at a time.

For constituting a WPAN, at least two devices should be running, one of them being a FFD operating as the PAN coordinator.

4.1 Network topology

Wireless Personal Area Networks in IEEE 802.15.4 can be implemented with a star or a peer-to-peer topology, as shown in Figure 4.1, with both full-function and reduced-function devices.

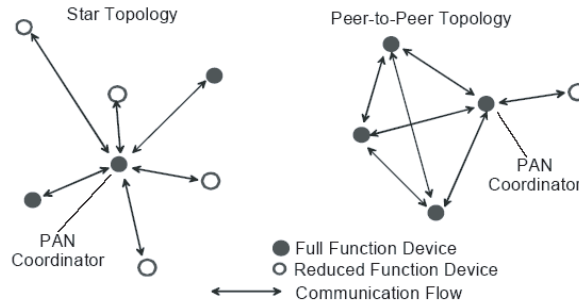


Figure 4.1: Network topologies

The star topology has a single central controller, called PAN coordinator, that holds the communication with all other devices. It might also have a specific application, but its main role is to control the network, and route the communications between two devices, if needed. The PAN coordinator can also be main powered, whereas in most of the cases the other devices will be running on batteries.

In the peer-to-peer topology, every device can communicate with any other one, providing the fact that they are in radio range of each other. This allows much more complex network formations to be implemented, and they could be ad-hoc, self organizing and self-healing. Routing a message is then a more complex task, but more reliable.

For example, a star topology would be used for automation processes, computer peripherals, toys and games, whereas a peer-to-peer topology is more likely to be used when talking about wireless sensor networks or industrial control and monitoring.

4.2 Physical layer

On the Physical layer side, the IEEE 802.15.4 defines specific RF frequencies, modulation formats, data rates and coding techniques. A broad description is not relevant here, but could be easily found on the standard specification

[1]. More relevant for us is the separation of the frequencies and the channels.

The 2.4 GHz radio band is designed worldwide for unlicensed operations. It consists on 16 RF channels, ranging from 2405 MHz up to 2480MHz. The channels are being spread every 5 MHz.

As specified in the Standard's specification, those channels are not coinciding with the Wi-Fi channels. Therefore, IEEE 802.15.4 systems can coexist with Wi-Fi systems.

Two other geographical RF bands are defined in the Standard. The first one, 868.3 MHz, is on the European unlicensed band (from 868 to 870 MHz), whereas 10 other channels are for the American Industrial, Scientific, and Medical band (ISM, 902 to 928 MHz).

4.3 Specificities for sensor networks

One of the most important issues for wireless sensor networks is their element's lifetime. In most of the cases, those networks are meant for monitoring situations where access is difficult, hazardous or too expensive. Maintenance operations must then be avoided, and maximizing the elements lifetime goes in that way.

Concerning the hardware, both electronic devices and sensors have a long lifetime, provided the fact they are used in the correct conditions (temperature and humidity range, etc.). During their design, they suffer many tests to guarantee their optimal behavior, such as salty atmosphere tests for their rust resistance. The other important component is then the batteries.

The batteries will be affected too by the external conditions, but their lifetime can be extended by proper power management.

The first possibility for saving batteries is by dividing the mote's life into active and inactive behavior. While active, the mote executes his tasks then goes into sleep mode. This sleep mode is included in the IEEE 802.12.4 Standard, and can be with fixed length, or until a new event occurs. Another aspect is to choose the appropriate sampling rate. If a mote is supposed to

monitor the temperature in some residential building, it makes no sense to make measurements every ten seconds. In the case of a nuclear power plant, the sampling rate should be much higher. Such a critical application, and many others, might require a continuous monitoring of the environment. In those cases, it is more relevant to place more sensors than needed. Beyond the safety side of this measure (if some sensors break down), this redundancy will allow the user to individually manage the power consumption, with an important flexibility.

Chapter 5

Moteiv Tmote rev.B

This chapter consist on a small description of the Telos motes used at the Automatic Control Group in the S3 Departement at KTH. More specific and detailed information is available on the motes datasheet [17] and on the *Moteiv* webpage.

5.1 Introduction

Telos is an ultra low power wireless module for use in sensor networks. It has the Chipcon CC2420 transceiver [20], that is a radio chip especially designed for the 2.4 Ghz band of the IEEE 802.15.4 standard.

Each node has a set of onboard integrated sensors, associated with a large on-chip RAM (10kB), and an on-board antenna providing up to 125 meters radio range outdoors. Each one is completely independent and can communicate with the other nodes collecting information, receiving and sending data in the network.

Designed with the dual goal of fault tolerance and development ease, it is robust and easy to handle. The motes are programmed using TinyOS [22], a modular open-source operating system that was developed by the U.C.Berkeley EECS Department. Specially designed for sensor networks, it is fitting the requirements of nodes with very limited resources. TinyOS has a component based architecture, that enables quite rapid and easy innovation

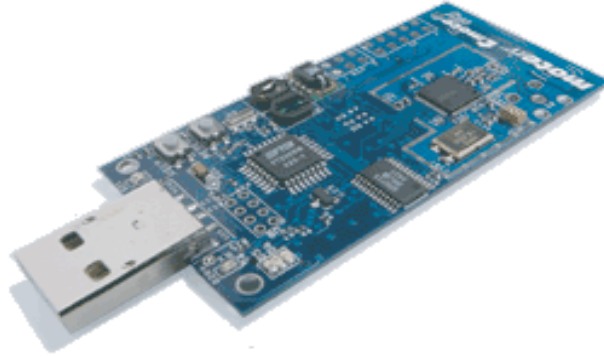


Figure 5.1: The Telos Rev.B mote

and implementation of the applications. The available library is composed by network protocols, distributed services, data acquisition tools and sensors drivers. All those tools are directly available for use, but could also be developed further by the user.

This modularity ensures that only the parts of the operating systems that are used by the application are loaded into the mote, thus minimizing the amount of needed memory.

Constantly under development, TinyOS has been ported to many different platforms, and is now one of the most used for sensor motes. The second full version was just published as a pre-release on August 30th, 2005.

5.2 Creating applications

The applications are written in NesC [2], a programming language close to C especially designed for networked embedded systems. NesC's programming model incorporates event-driven executions and component-oriented application design.

Every application is programmed out of *components*, that are linked during the compilation. Every component is made of several *tasks*, and the component's behavior is defined by *interfaces*. Those can be either provided

by the component, or used by it and can basically be seen like this: the used interfaces are all what the component needs to be able to run his tasks; in the other side, the provided interfaces are the functionalities that the component will give to the its user. Interfaces are bidirectional, and consist on *commands*, which are the functions to be implemented by the provider, and *events*, which are the one that are to be implemented by the interface's user.

The compilation is ran under Cygwin, a Linux-like environment for Windows platform (included in the TinyOS distribution). During the compilation of the application, every component of the TinyOS that is needed will be uploaded. The generated binary, thus including both application operating system, will then be loaded on the mote. Once the mote is loaded, it is completely independent, and will run its application.

5.3 Communication between motes

Depending on the application running on the motes, they will be communicating with each others, or with one or more access points to the network (data collector). If the user wants to interact with the WSN, he has to connect to one of those access points, which becomes the Base Station. Many applications are designed such as the Base Station is always plugged into the USB port of a personal computer, for monitoring the network.

Once the data packets are received, they are routed to a logical port on the computer thanks to the *SerialForwarder* program. They are now available for any use, and can be sent to a graphical interface or a Cygwin shell depending on the application. This java tool has also been developed by the same group than TinyOS and is also open. It is included in the default TinyOS distribution.

Another useful java tool is the *Listen* command, that displays the packets in raw data. Although they are not easy to read at first sight, the user quickly gets used to that and identifies which packets are the most relevant for him.

5.4 Emitting power

According to the IEEE standard, a 802.15.4 transmitter shall be capable of transmitting at least -3 dBm, but devices should transmit lower power when possible to reduce interferences to other devices and systems.

The output power of the Telos rev.B motes is programmable and is coded by a number between 3 and 31, the first one being the lowest power (-25 dBm), and last representing full power emission (0 dBm).

PA_LEVEL	Output Power [dBm]
31	0
27	-1
23	-3
19	-5
15	-7
11	-10
7	-15
3	-25

Figure 5.2: The programmable output RF power values (dBm)

At full power, the radio range is about 50 meters indoors and 125 outdoors. More details, as the radiation patterns are available in the motes' documentation [17].

Chapter 6

RSSI experiment

6.1 Overview

The use of received signal strength intensity (RSSI) is one of the most common methods that has been studied for localization purposes.

As the signal propagates, its energy decreases with distance. If the receiver knows the transmission power, it can read this value when the signal is received and thus have an estimate of the distance between sender and receiver.

The model and the technique are simple, but have different parameters that can affect its use for the 802.15.4. The model requires a reference distance d_0 and the received power at that distance $P(d_0)$:

$$P(d) = P(d_0) - 10\alpha \log \frac{d}{d_0} \quad (6.1)$$

The factor α indicates how the signal decreases with distance. In open air, α is 2, but it is much more complicated to determine a realistic value for indoor radio propagation. An approximation for 2.4GHz is proposed in [18], for a value around 3.5.

6.1.1 Limitations for RSSI approach

Radio propagation is affected by fading and shadowing. For an indoor application, this will have an even more important effect, as the radio signal

can be affected by the surrounding environment, and the reflections will create a multipath solution. The walls and the furniture will of course work as obstacles for the radio signal, but those are permanent obstacles. In a working environment such as the case of the lab at the S3 department, working people and their computers will act as extra obstacles for the radio signals. They are mobile, and are not always present, so the radio propagation profile will have a different shape if computed by daytime or nighttime. And the received power at the reference distance for the model would vary.

6.1.2 Application to Telos

The CC2420 radio chip used by the Telos motes has a programmable output power (default power being 1mW, noted as 0dBm). Depending on the application used, this value could change or stay fixed, so the receiver then knows it *a priori*, and there is no need to broadcast it in the data packets.

The received signal strength will also depend on the battery settings. As described in the Telos motes' datasheet [17], the mote plugged to the base station receives power from it, and so it operates at 3V. But the independent motes need a battery voltage ranging from 2.1V to 3.6V. Those experiments are done with mid-life batteries, having a 2.5V potential. An additional study with new batteries will be presented in Section 6.6.

6.2 Measurement campaign

The idea of our localization technique is the following. Knowing in which environment one would like to compute the localization of some elements, we first build a signal strength map. This is achieved by sampling and interpolation under known circumstances. Any device in that environment can then measure the signal strength and match it with the map to estimate its position.

When applied to a new space, the "learning" of the new environment could be done with radio commanded robot that will be moving along and measuring the available signal strength on different spots.

This chapter describes the details of our measurements, studies different

influences the environment can have, and looks for the limitations to this method for localization.

6.3 Measured values: RSSI and LQI

Both values are coded on one byte. The received signal strength is mapped to dBm according to the chart presented by Moteiv (see Figure 6.1). As it is described in the Telos motes specification [17], this corresponds to an offset of roughly -45:

$$RF_{power}(dBm) = -45 + RSSI_{read}(decimal) \quad (6.2)$$

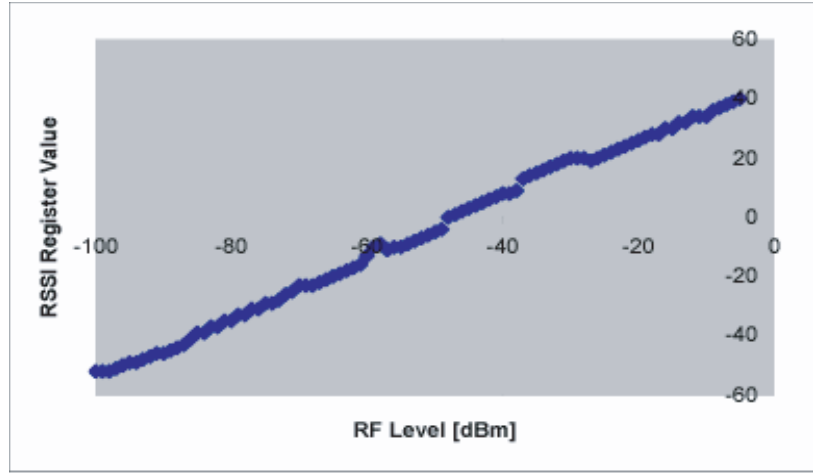


Figure 6.1: Received Signal Strength Indicator mapped to RF power (dBm)

The Link Quality Indicator value is produced for each packet by calculating the error rates on the first eight chips, the CC2420 gives a value between 50 and 110, the latter being the best one.

Signal strength and link quality values are not necessarily linked. But if the LQI is low, it is more likely that the RSSI will be low as well. Nevertheless, they also depend on the emitting power. A research group [19], had the following results:

	Low RF	High RF
LQI	105	108
RSSI	-75 dBm	-25 dBm

Even though they do not precise how far from each other the sender and the receiver were located, this illustrates perfectly that both low and high power emissions guarantee a good link quality. The low RF emissions could be more sensitive to external disturbances.

6.4 Experimental settings

The RSSI values are available on the receiver side, but using a specific application is more convenient. ChipconRSSI is an application dedicated to signal strength measurements and experiments was developed by Kamin Whitehouse and made available on the CVS tree of the TinyOS project. This was initially made for Mica2 motes, so not usable on the Telos rev.B motes used at the S3 Department. That application have been ported later on to Telos motes, under the name TelosRSSI, but getting it running is not an easy task.

The purpose is to define some motes as being remote senders. They are placed on the different locations from which one would like to measure the signal strength, and a last mote is connected to a computer. Running the Matlab application should ask node number one to chirp, and computes the corresponding RSSI, then asking for node number two, and so on. Setting up the task requires a collaboration of Matlab, the Cygwin shell and the different Java classes. The diversity of those tools implies that some incompatibilities may appear. For example the compilation of the Java classes needed for connecting Matlab to the SerialForwarder tool is not possible when using the latest Matlab built (R14).

The TinyOS environment being open-source, it allows many contributions and modifications from users. The drawback of this is that it is quite poorly documented. When having a problem with someone else's application, going into it for debugging it could be really tricky, especially if it has been developed further a couple of times. The code is commented from time to time, but many of those comments even are in contradiction with themselves.

However, the TinyOS-Users help mailing list is a very important resource for solving some of the problems [23]. Searching in the archives, one can find other people that had similar problems, and look for the indications or solutions given by other members, if any. There is always the possibility of sending an e-mail to this list, but it has to be very precise to attract the attention, as the community receives hundreds requests a day.

After a long time spent on setting up all the devices and environment, that was still unsuccessful with the TelosRSSI application. The decision is then made to modify an existing application to extract the desired values.

6.4.1 The application

Each radio packet consists on two parts: the AM header (10 bytes), and the Payload, with a size depending on the application used. Once one checks the AM.h header file, we see that the RSSI and LQI values are not included in the packets, but are computed even if not displayed.

We then choose to use the CntToLedsAndRfm application for our distant motes, which consists on running a 4Hz counter and displays its value on the mote's leds, sending it at the same time on air. On the receiver side, a small modification has to be done on the TOSBase application in order to be able to read both values on the received packets.

Looking to the header file description it is straightforward to identify all the different fields:

```
04 01 08 FC FF FF FF FF 04 7D FC 00 01 00
04 01 08 FD FF FF FF FF 04 7D FD 00 01 00
04 01 08 FE FF FF FF FF 04 7D FE 00 01 00
04 01 08 FF FF FF FF FF 04 7D FF 00 01 00
04 01 08 00 FF FF FF FF 04 7D 00 01 01 00
04 01 08 01 FF FF FF FF 04 7D 01 01 01 00
```

- The first byte is the size of the payload, here being 0x04 bytes.

- The next two ones, 0x01 and 0x08 are frame control fields specific to Telos.
- The fourth one is the Data Sequence Number, also used for controlling the frames coming from each mote. As one can see, it is always incrementing, thus here following the counter on the payload.
- The next four ones are for the destination PAN identifier and destination address.
- Then 0x04 is for the AM message type, and 0x7D for the Group ID.
- The last four are for the payload itself, where one can see the counter increase.

It has been pointed out by members of the TinyOS community that making such modifications could make the payload disappear. Instead of routing the RSSI and LQI values to the two last bytes of the payload, we then decided to put on the AM header bytes instead, as the two last fields are not relevant for us (Type and Group Identification). Doing this, and losing the payload, we reduce the capacity of our counter. It was coded on two bytes on the payload, but now only the fourth byte of the header, the packet sequence number, can be used.

The event `RadioReceive.receive(Msg)` of the `TOSBase` code is completed with:

```
MsgRSSI = *Msg;
MsgRSSI.type = Msg->strength;
MsgRSSI.group = Msg->lqi;
pBuf = &MsgRSSI;
```

This new `TOSBase` application is compiled and loaded on the base station whereas the `CntToLedsAndRfm` mote will then be placed on different locations to report the respective RSSI and LQI.

Another application often used by the Research Group at the S3 Department, is the *Aegis* application [24]. A closer look to its messages shows that the higher byte of the frame control field (*FCFHI*, second one in our AM

header) is always set to 0x21, while it is always 0x01 for the counting mote. This will allow us to easily differentiate our messages from the Aegis ones, and thus use Aegis messages as "STOP" messages between two locations' measurements.

6.4.2 Post-processing

On the laptop's side, running the SerialForwarder and the Listen tools allows us to see the received packets, and forward them to a text file. Those results have to be converted from hexadecimal to decimal values, in order to be later imported on Matlab for exploitation. That conversion can easily be made through an Excel macro, and we can also remove all the non-useful bytes from the messages.

Using the DSN field has another advantage: as all the packets are numbered, we can easily check if some of them have been lost.

From the initial 14 bytes, only 4 remain with the needed information:

11	68	250	108
11	69	247	107
11	70	248	109
11	71	250	108
99	01	245	103
99	02	243	104
99	03	250	108
11	01	235	102

- The first number describes if it is a measurement packet (11) or a STOP packet (99).
- The DSN field is the number of the received message.
- The third value is the measured signal strength. This register value is not directly the RF level, the relation between the two having been described in Section 6.3.

- The last one is for the LQI value, coded from 50 to 110, the bigger the better.

This file is then saved and can be analyzed using Matlab. First of all, the Matlab code will look for the STOP packets to divide that file into several matrices included in a cell array structure.

6.4.3 The map

For creating the map, simplifying conditions had to be set. The received signal strength can vary with the orientation of the antenna, and height position of the emitting node, as well as for the receiving mote on the laptop. The mote is also very sensitive to its direct environment (presence of a body, hand and fingers holding the mote, etc...). In order to have reliable results, the mote was always oriented and handled in the same way.

Assuming that the receiving node will always be the laptop, the mote will then be roughly around one meter above ground (the height of a desk). In the other side, the sending node can be placed everywhere. Figure 6.2 shows how this is influenced. The blue plots are for the RSSI values when the mote is hand held, one meter above ground, and the red ones when the mote is standing directly on the ground. From the first graph to the third one, the distance to the base station increases, thus the RSSI value are decreasing. But we see that the difference between the two elevation stays around 10 dBm, the mote on the ground giving of course a weaker signal.

The first plot shows big disturbances, due to two persons passing by. Those two results clearly comfort the fact that in order to be able to use this measurement for our method, the repeatability of the experiments conditions is highly important.

The lab with the desks is shown in Figure 6.3, also including the measuring spots and the laptop's location.

6.4.4 First results

The first packets received can show some oscillations in the RSSI. This could be due to the setting up of the mote, after a reset. We then compute two

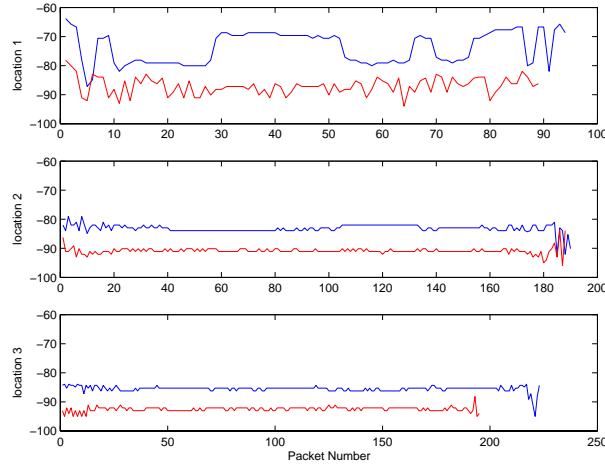


Figure 6.2: RSSI (dBm) varying with distance and height

RSSI values: one as a mean of the 48 first received packets, and a second one without counting the first 16 packets.

In most of the cases, the received value is stabilized. In some measurements, even if it is not the case, the variations are not very important. It could happen from time to time that a packet has a low strength, but this is generally linked to a very bad LQI for that moment. An example of this situation is shown on Figure 6.4 where the fourth packet has the lowest RSSI and LQI values.

From those observations, it is then appropriate to extend the measuring period. The counter has a capacity of 255 which means 64 seconds, as we are sending at 4Hz. Having around one minute for each measurement helps getting rid of the temporary events (one person passing by, for example). However, it is not a problem if we measure a bit longer than one minute. Of course the counter will loop, but it could be taken into account by knowing approximately how long was the sampling time. The Matlab analysis should check this, in order not to consider some received packets as if being lost.

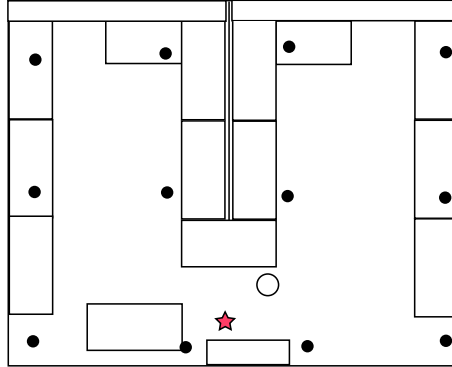


Figure 6.3: The lab environment

6.5 Results

6.5.1 Normal environment

This first experiment is made in the normal environment of the lab, that is with people working. However, for each measurement, nobody is walking through the direct line of sight between the sending mote and the base station. This is to reduce such disturbances that would have a very bad impact on the signal strength values. Other movements and activities will also create disturbances, due to multipath or reflections, but we consider those ones as being part of a normal environment.

The first measurements show that the RSSI decreases very fast in the direct neighborhood of the sending mote:

If we place the two motes so that their antennas are superposed, we get an RSSI of -3 dBm, which almost corresponds to the highest emitting power (see Figure 5.2). When our two motes are distant of only a few centimeters, the received power drops around -25.5 dBm, and at half a meter distance it is just -40 dBm.

This important attenuation around the emitting node would require a very precise mesh on that area. And also makes the variations on the other parts of the lab appear relatively much less important. It is then decided not

Packet	Rssi	Lqi	Packet	Rssi	Lqi	Packet	Rssi	Lqi
1	-79	107	17	-66	106	33	-66	107
2	-80	107	18	-66	104	34	-65	106
3	-78	108	19	-67	104	35	-66	105
4	-90	83	20	-68	107	36	-66	104
5	-71	106	21	-68	101	37	-66	105
6	-70	106	22	-67	103	38	-66	107
7	-66	107	23	-66	104	39	-66	104
8	-66	103	24	-66	104	40	-66	106
9	-68	107	25	-65	104	41	-67	104
10	-68	104	26	-66	106	42	-68	105
11	-69	106	27	-66	107	43	-68	105
12	-70	105	28	-66	103	44	-68	106
13	-71	107	29	-67	105	45	-68	105
14	-71	106	30	-66	106	46	-68	107
15	-70	103	31	-66	106	47	-67	103
16	-68	104	32	-66	106			

With all values:		Without the first 16:	
Mean RSSI	-68.5	Mean RSSI	-66.5
Mean LQI	104.8	Mean LQI	105.0

Figure 6.4: A bad LQI illustration

to measure the strength in the direct neighborhood of the base station.

From our measurements, a global mesh over the lab is defined, and the results linearly interpolated for every location. A two dimensional presentation of the results is given on Figure 6.5, where the RSSI is color coded.

On this we see one irregularity: the fact that the lowest RSSI value is for a central spot (-67 dBm), instead of being on the upper corners (i.e. the more distant locations). Checking the values more in details, the link quality for all spots appears to be really good, as all the values are in the 105 range. Concerning the signal strength, we can see that the corresponding standard deviation is quite low (from 1.1 to 2.8) showing the fact that all packets in each measuring spot are quite coherent. For the incriminated spot, we have a value of 2.7, which is on the limit.

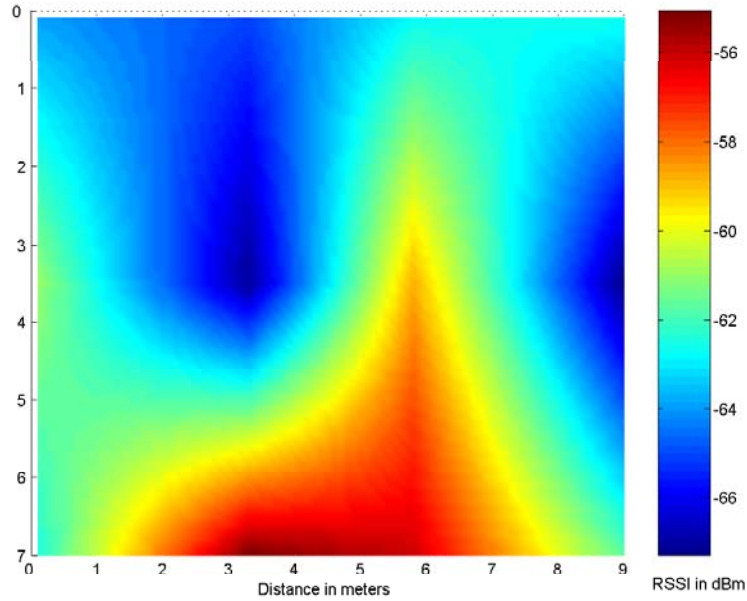


Figure 6.5: Measured received signal level as a function of position

6.5.2 Quiet environment

To see how important are the disturbances we considered before as being part of the normal environment, we perform again our experiments in a completely quiet environment. Nobody is present in the lab during all those, and this yields to a smoother repartition of the received signal strength, as shown on Figure 6.6.

One can see that the highest measuring point is a bit on the left, and this is due to the fact that the base station is not exactly in the middle between the two measuring spots. As we said, in the immediate range of the base station, the RSSI is varying very quickly, so a small difference in the distance shows up a bigger one on the measured values.

Fluctuations are even lower than for the normal environment, as all standard deviations for both RSSI and LQI are lower than previously. It is hard to state certainly if that is due to a better signal, or just to less disturbances because of the people not present on the lab, but this is much closer to what

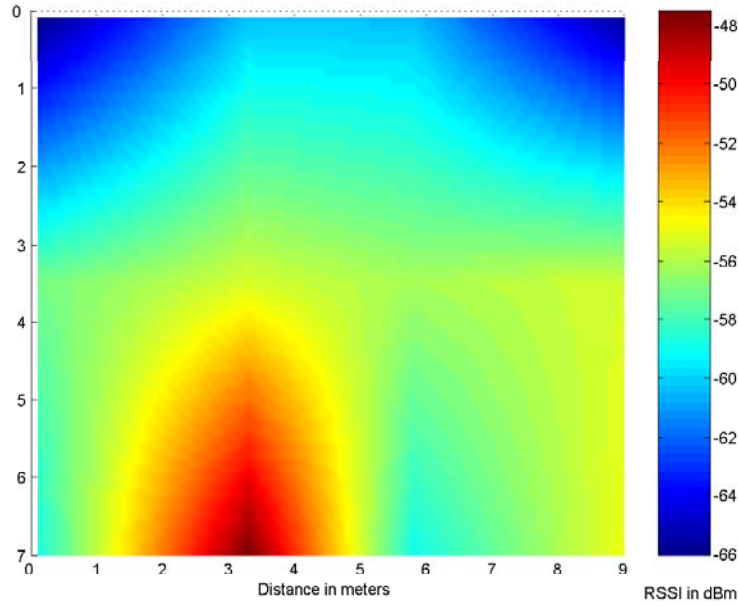


Figure 6.6: Received signal level in a quiet environment

should be an ideal mapping of the signal strength over the lab.

6.6 Influence of batteries

The radio communication is the most expensive activity, in terms of power consumption. When a mote is on standby, it only consumes $5\ \mu A$, and this increases to $2mA$ when the MCU is on but the radio off. When the radio is turned on, it increases further to $20mA$, should it be on reception or transmission mode [17].

As the motes are supposed to be working with batteries ranging from 2.1 to 3.6V, this current consumption should be granted, regardless the available voltage. But a longer radio activity will degrade the batteries faster and the result would be a chaotic behavior when too close to the lower voltage boundary.

We then performed twice the same measurements, on five different spots, once with our regular batteries and then a second time using new batteries, that have a potential of 3.2V. The results are presented in dBm on Figure 6.7, the red plotting being for those new batteries. The distance to the base station increases from each graph to the next one.

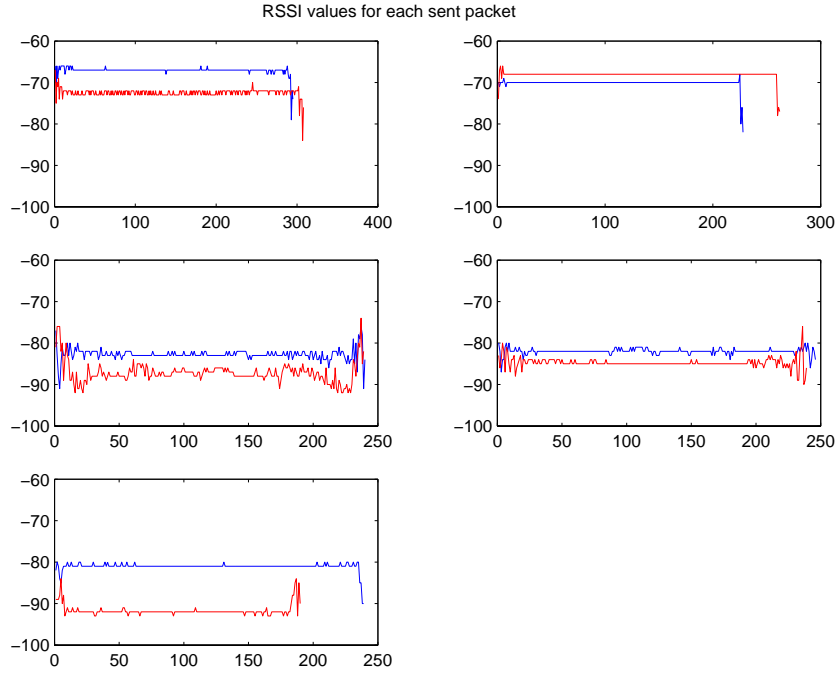


Figure 6.7: RSSI values (dBm) depending on the batteries used

For the second location, we don't have the expected result. This could be due to the specific environment around that spot: many metallic devices and shelves were surrounding the motes, such as a very small difference in the placement would be transformed in a bigger difference for the signal strength.

For the last location, we can see that the difference between the two situation is quite important. The RSSI obtained with the new batteries is much weaker than with old ones. Checking the LQI parameter for that experiment on Figure 6.8, we notice that it is unusually low (around 80 instead of 105). A low LQI value explain the low RSSI values, but how to explain

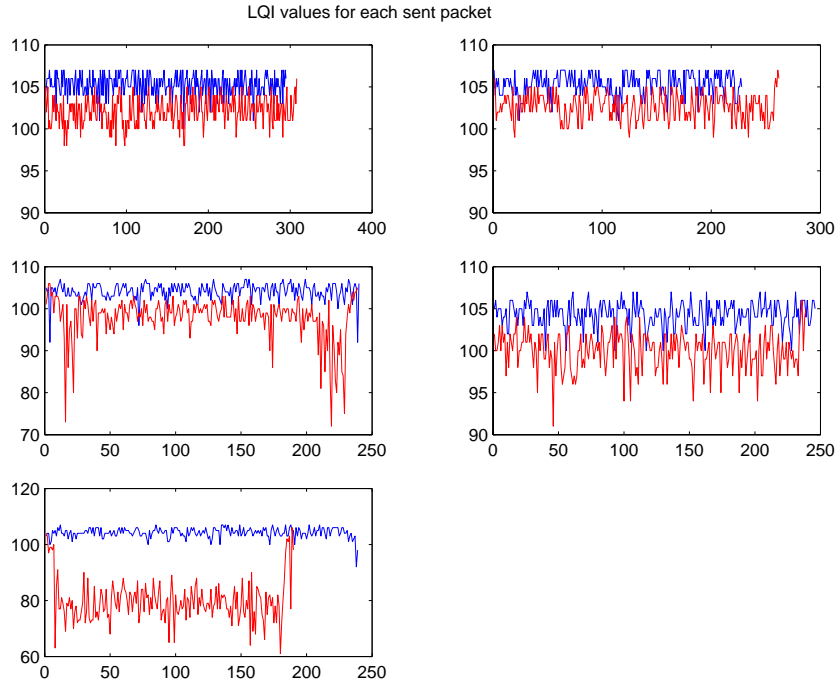


Figure 6.8: LQI values depending on the batteries used

the poor quality for that experiment ? Such a stable LQI degradation never was observed on other measurements made on that spot, nor on any other experiment.

6.7 Influence of emitting power

The received signal strength depends obviously on the emitting power. Moreover, to be able to use this information for localization purposes, the receiver needs to know at what precise power the sender is emitting.

Another aspect that is influenced by the emitting power is the radio range. According to *Moteiv*, the onboard antenna has a 50m radio range indoors, and up to 125m outdoor. On our measurements with the *CntToLedsAndRfm* application, we experienced an important loss around 20-25 meters, despite

sending packets at full power.

As the output power of the Telos rev.B motes is ranging from -25 dBm to full power emission: 0 dBm (see Figure 5.2), we recommend using full power emission. Indeed, this would allow better visibility for the attenuations on the received signal strength, especially in an indoor environment suffering of many obstacles.

6.8 Influence of radio interference

As other motes could be broadcasting on the same radio channel, the interferences created could damage the signal strength received by the base station. The objective of this section is to analyze this aspect of the communication between the motes.

6.8.1 Using the Aegis application

The Telos applications' scheme is to organize motes by groups. This allows several small networks to be running on the same space, without that packets from one application are used by motes from other applications. In our experiment process, we use a mote programmed with the TOSBase application. Even if a GroupID is specified, the TOSBase hears every packet, but then filters out the ones with an ID not matching its own ID.

Thus the result should not be depending on the fact that the extra motes are from another group or not.

This first experiment is made using the Aegis application for the disturbing motes. Six motes have been programmed with the application, and thus are sending two messages per second. The RSSI measurement is made for one location, but for four different settings.

First of all, the blue plots on Figure 6.9 show the reference value for the signal strength on the chosen location, i.e. with no other motes running.

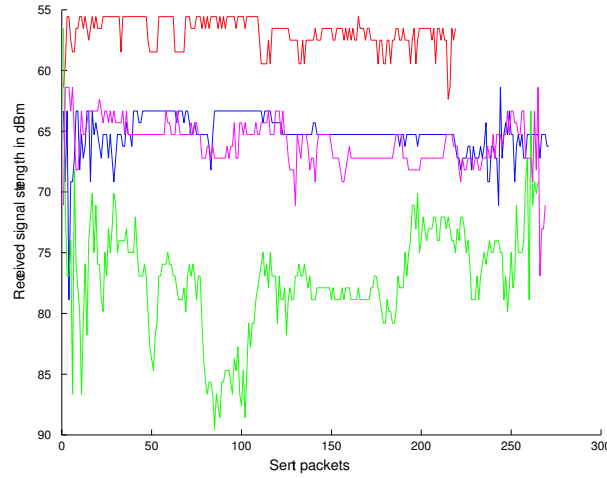


Figure 6.9: Influence of the Aegis motes

Then the Aegis motes are involved with three different settings. First placed quite close to the base station, around one meter range (red plots), then spread all over the room (green), then finally close to the emitting motes (magenta).

Surprisingly the received strength is higher when the motes are close to the base, even higher to the reference measurement: a mean value of -57.2 dBm against -65.1 dBm. When the disturbing motes are close to the emitting one, the influence is almost negligible (-66.0 dBm).

The biggest degradation comes when the motes are spread around the room. The measurement gives a mean value of -77.2 dBm, but this is also due to an extra disturbance around samples 70 to 110. During that time, one person was passing by, walking just along the direct line of sight between the emitting mote and the base station. This could also be seen by looking the LQI values for that time (Figure 6.10), where the big drop is evident. Those plotting also give the interesting result that the link quality is not affected by the other motes, as the value is always around 105, as we used to have before.

Another low link quality is observable at the beginning of the measurement, and the consequence of this can be seen on the RSSI plot. But this

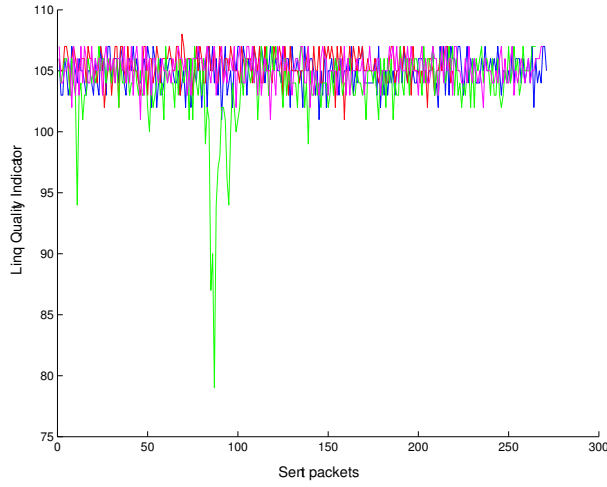


Figure 6.10: The link quality is not affected

does not correspond to any special event during the measurements, so it has been decided to keep them.

So the real mean RSSI value when the motes are spread in the room is more likely around -76.1 dBm (calculated by excluding the 40 samples), and this corresponds to the biggest degradation due to interferences.

This is a really important matter for our localization purpose. The matching of some measured values to an RSSI map could then really be affected by other people working with motes.

6.8.2 Due to other sources

As the Telos motes we are using are operating in the 2.4GHz band, there are many other devices that could be interfering with them. Many Bluetooth equipments are used in the building, such as wireless mice or mobile phone accessories. Moreover, the Department's wireless network is also using this band. Although there is no direct interference with the WiFi channels, the channel quality can be affected by the overload and thus distort some of the packets.

Due to this open band and the radio range of such devices, it is impossible to state surely enough how many are used and are in the radio range of our motes... But, being in an university, we can assume that at night the radio activity is much lower than during the day. Even if some users are present, they will not be a large demand of resources, and thus no many frequency changes that would affect more our signals.

Making measurements late in the evening, we obtain the plotting presented on Figure 6.11.

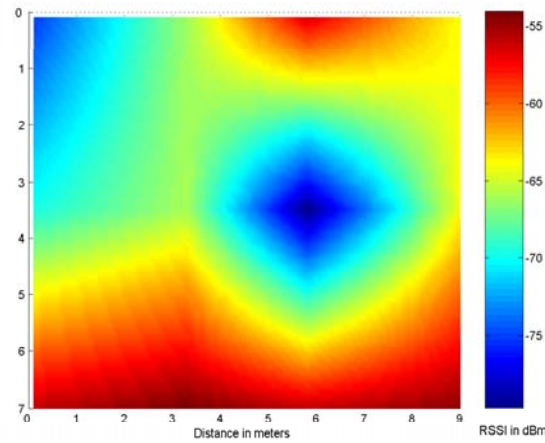


Figure 6.11: Received signal for a low radio activity

The first thought when looking at this picture is the central blue spot, denoting a very low signal strength. This should not have been the case, as the distance between the mote and the base station is relatively small. Looking further to the results, it appears that the RSSI was varying a lot and the link quality is mainly bad (Figure 6.12). The fact is that a pillar is present between the sending mote and the receiving station. Almost in the direct line of sight, this could explain such behaviors for this measuring spot - even though they are not systematical.

Making new measurements for that spot, and checking that the values seem to be more realistic, we came to a main value of around -68 dBm,

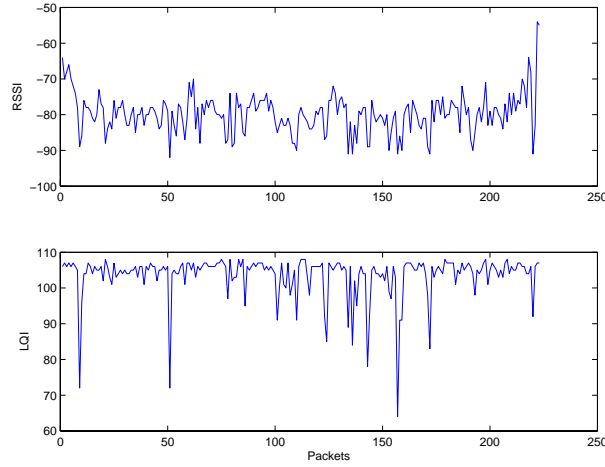


Figure 6.12: RSSI and LQI fluctuations

which leads instead to the map presented on Figure 6.13.

Those results are more appropriate than the ones of the normal environment showed of Figure 6.5. But this new map is not as smooth as the map on the quiet environment during daytime (Figure 6.6), that is close to an ideal RSSI map for the lab.

This illustrates that the whole procedure is not really static. Even if the complete measurements are done during a short period, the same experiences made at a different point of time will not produce exactly the same results. This non-reproducibility of the measurements is a important obstacle for using a calibrated RSSI map for localization processes.

6.9 Influence of the motes themselves

During our experiments, it appeared that the received signal strength is varying depending on the mote used as the counting mote. A series of measurements with four different motes, placed on the same location and using the same batteries, showed a difference up to 10 dBm (Figure 6.14).

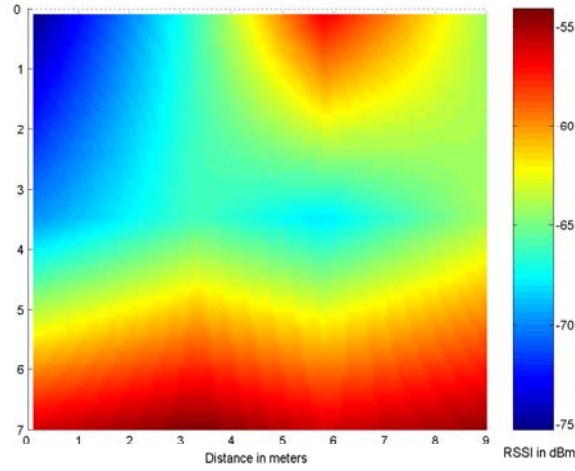


Figure 6.13: Received signal for a low radio activity

The hardware of the motes is supposed to be the same, and thus the radio broadcasting the same way. But even if the motes are relatively new, they have not always been used the same way, some of them might have been damaged in some way. If the RSSI varies, the LQI values are staying at normal and correct values. Of course, we always kept the same mote for our measurements, in order to be able to compare the results.

As the Department just purchased new motes, it is a good opportunity to check this with brand new motes. Those motes have no sensors, but as those sensors are not called in the application we are using, this should have no influence on the battery consumption nor on the signal strength.

The measurement made on the same spot than previously, this time with four new motes are presented in Figure 6.15.

The received values are a bit different, but the first impression is that they are not spread as much as in the previous case (with old motes). Here they are ranging from -59 to -55 dBm, which is in accordance with two of the values we had.

This would lead to the conclusion that the motes themselves have an influence on the signal strength. Their internal antennas might have been

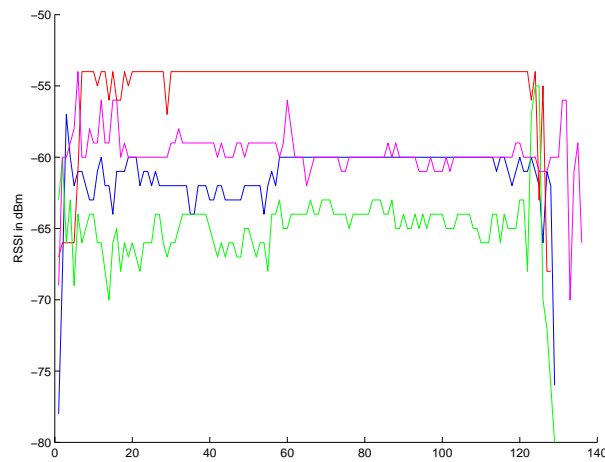


Figure 6.14: RSSI difference for four motes

damaged somehow, but this does not really fit the experience with the new motes. Furthermore, the link quality values are always very good, for both new and old motes. One could think that those should be lower if there were a hardware damage...

We forwarded this question to the TinyOS-users help list, but unfortunately no answers so far about anybody that could have experienced something like this.

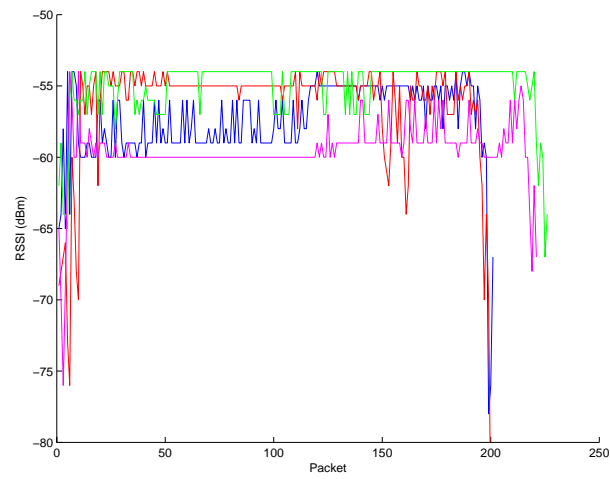


Figure 6.15: RSSI using four new motes

Chapter 7

Conclusions

As wireless sensor networks became a key technology and are used in more and more industrial and environmental problems, defining an effective localization algorithm became an important task. The literature survey work showed that an optimum algorithm could not be defined yet, and thus the choice of the suitable one has to be founded on the specificities of the situations, taking into account the size of the network, as well as the deployment methods and the expected results.

Continuing further with one of the ideas proposed, the Bounding box we studied the influence of different parameters in terms of performances. This lead to two major developments that have been proposed in the thesis, and which improvements significantly increase the positioning accuracy. They do not require much more computational costs, and perfectly match the distributed algorithm's requirements.

A third aspect of the work concerned the feasibility of approaches based on received signal strength indication. The measurement campaign held in the lab gave us precious information regarding the nodes themselves, and the use of signal strength for localization. Although the localization map gave promising information for the positioning possibilities, it pointed out that the fluctuations in the RSSI values may limit the reliability of many localization schemes.

Indeed, the experiments showed that the values are really sensitive to the environment and the nature of the disturbances, which are even more important in an indoor configuration.

Due to the important amount of scenarios where localization processes comes into play, it has been and will continue being an important research field. The different types of algorithms might be improved, and new ones developed, but another issue will be to determine which algorithm could give the more interesting results, for a given configuration.

List of Figures

3.1	Principle of the bounding box	19
3.2	Creation of the system	20
3.3	Bounding box results	22
3.4	Reached accuracy using the bounding box	23
3.5	Restrained listening possibilities	25
3.6	Listening approach results	25
3.7	Comparison of the two shapes	27
3.8	An example of non-compact intersection set	28
3.9	Reached accuracy using the circles intersection	29
3.10	Global results for the circles intersections	30
3.11	Numerous positions added by the moving beacon	31
3.12	Influence of a beacon moving by 6 units	31
3.13	Using a moving beacon (+6) and listening to the neighbors . .	32
4.1	Network topologies	35
5.1	The Telos Rev.B mote	39
5.2	The programmable output RF power values (dBm)	41
6.1	Received Signal Strength Indicator mapped to RF power (dBm)	44
6.2	RSSI (dBm) varying with distance and height	50
6.3	The lab environment	51
6.4	A bad LQI illustration	52
6.5	Measured received signal level as a function of position	53
6.6	Received signal level in a quiet environment	54
6.7	RSSI values (dBm) depending on the batteries used	55
6.8	LQI values depending on the batteries used	56
6.9	Influence of the Aegis motes	58
6.10	The link quality is not affected	59

6.11	Received signal for a low radio activity	60
6.12	RSSI and LQI fluctuations	61
6.13	Received signal for a low radio activity	62
6.14	RSSI difference for four motes	63
6.15	RSSI using four new motes	64

Bibliography

- [1] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4-2003, New York: IEEE Press. 2003.
- [2] D.Gay, P.Levis, D.Culler and E.Brewer, *nesC 1.1 Language Reference Manual*, May 2003.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, *A survey on sensor networks*, IEEE Communications Magazine, 2002.
- [4] C. Chong and S. Kumar, *Sensor Networks: Evolution, Opportunities, and Challenges*, Proceedings of the IEEE, August 2003.
- [5] K.Whitehouse and D.Culler *Micro-calibration in Sensor/Actuator Networks*, UC Berkeley, 2003.
- [6] S. N. Simic and S. Sastry, *A distributed algorithm for localization in random wireless networks*, Discrete Applied Mathematics, UC Berkeley, November 2002.
- [7] D. Moore, J. Leonard, D. Rus, and S. Teller, *Robust distributed network localization with noisy range measurements*, in Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys '04), Baltimore, 2004.
- [8] L. Hu and D. Evans, *Localization for Mobile Sensor Networks*, Proceedings of Tenth Annual International Conference on Mobile Computing and Networking, 2004.
- [9] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale sensor networks," in Proc. ACM/IEEE 9th Annu. Int. Conf. Mobile Computing and Networking (MobiCom'03), 2003.

- [10] D. Nicosescu and B. Nath, *Ad-Hoc Positioning System*, Proceedings of IEEE GlobeCom, November 2001.
- [11] D.Culler, D.Estrin and M.Strivastava, *Overview of Sensor Networks*, IEEE Computer Society, 2004.
- [12] Y.Shang, W.Ruml, Y.Zhang, and M.P.J.Fromherz, "Localization from mere connectivity," in Proceedings of the 4th acm international symposium on mobile ad hoc networking & computing (MobiHoc'03), June 2003.
- [13] K.Langendoen and N.Reijers *Distributed localization in wireless sensor networks: a quantitative comparison*, Tech. Rep. PDS-2002.
- [14] D.Niculescu and B.Nath *Error Characteristics of Ad Hoc Positionning Systems (APS)*, MobiHoc'04, May 2004.
- [15] K. Whitehouse *The design of Calamari: an ad-hoc localization system for sensor networks*, Master's Thesis, UC Berkeley, 2002.
- [16] A. Savvides, H. Park and M. Srivastava, *The bits and flops of the N-hop multilateration primitive for node localization problems*, In First ACM International Workshop on Wireless Sensor Networks and Application, 2002.
- [17] Moteiv Corporation, Telos rev-b datasheet, <http://www.moteiv.com/products/docs/telos-revb-datasheet.pdf>, 2004.
- [18] "An Introductio to Indoor Radio Propagation", Spread Spectrum Scene, <http://sss-mag.com/indoor.html>, June 2001.
- [19] Moteiv Corporation, Tinyos-users helplist, <https://mail.millennium.berkeley.edu/pipermail/tinyos-users/2004-August/000631.html>, 2004.
- [20] Chipcon AS, Chipcon CC2420 datasheet, http://www.chipcon.com/files/CC2420_Data_Sheet_1_2.pdf, 2004.
- [21] ScatterWeb, <http://www.scatterweb.net/>, 2005.

- [22] TinyOS community forum, <http://www.tinyos.net/>, 2005.
- [23] TinyOS community help-list, <http://www.tinyos.net/support.html#lists>, 2005.
- [24] I.C.Molero, *Telos and Aegis quick start guide*, April 2005.