WSN localization with the Senseless framework

Peter De Cauwer, Tim Van Overtveldt, Jeroen Doggen, Jerry Bracke, and Maarten Weyn.

Abstract—Localization of nodes in wireless sensor networks (WSNs) is important to context-aware and position-dependent applications; data are generally meaningless without a known location. Many algorithms exist for localizing nodes using RSS; however, a detailed quantitative comparison of these algorithms has not yet been published. In this paper, we present such a quantitative comparison of algorithms which use RSS as a ranging method and present a localization software framework called Senseless. We also conducted a survey about the influence of the orientation of a node. Our study finds that the received power is not equal in all directions and that no single best algorithm for localization exists to date. Each algorithm has a different purpose and diverse properties. Learning from these algorithms and techniques, the correct algorithm can be chosen for the correct environment and a more advanced localization system is feasible. Using this framework, we implemented a couple of centralized algorithms; Trilateration, Min-Max, Centroid Localization and Weighted Centroid Localization.

Index Terms—Wireless Sensor Network, WSN, RSSI, localization, TelosB, TinyOS, Senseless.

1 Introduction

A Formal definition of Wireless Sensor Networks is given in [1]. The purpose of a WSN is to monitor some physical phenomena such as the ambient temperature, air humidity, and the presence or absence of a certain chemical. Usually, this data is collected at a common point, the data sink, where the data can be further processed or analyzed by the user. WSNs enable a great deal of new applications including environmental and habitat monitoring, smart homes and battlefield control [2] [3].

Accurate and low-cost sensor localization is an essential service and is important to many of these applications. The measured data is generally meaningless without knowing where it originated from. That being said, there are other reasons to acquire the location of a sensor. Location can be a goal in itself, such as in warehousing and manufacturing applications. Another added benefit is the possibility of geographical based routing.

Given the popularity of GPS, one would think of it as a possible solution. However, this is considered a bad solution because of several reasons. Firstly, a great deal of WSN applications are built for indoor environments. Clearly, GPS does not suffice as it requires Line-of-Sight with at least four satellites. Secondly, GPS requires significant power to operate, which is a sparse resource in WSNs. Thirdly, GPS adds to the size and cost of a WSN node which should be kept as low as possible. This does not mean, however, that GPS is entirely out of the question. A small number of WSN nodes will need a priori knowledge about their location; this can be done by a GPS receiver or by manually mapping the node to a position on a map, or some other method. The number of these nodes should be kept as low as possible, in order to keep the deployment cost and time low.

Localization techniques which are specific for WSNs are based on pair-wise measurements between nodes to estimate the positions. A small fraction of the network should have a known position as described in the previous paragraph. These nodes are called anchor nodes. The other kind of nodes, without a known position, are called blind nodes. Thus, the goal of a localization system is to determine the position of the blind nodes by communicating with the anchor nodes. We can divide these techniques into two categories: range-based and connectivity-based. Range-based methods estimate the distance between nodes with ranging method such as ToA, AoA and RSS. These techniques typically provide superior accuracy but are more complex than connectivity-based algorithms. These do not estimate the distance between nodes but determine the position of a blind node by their proximity to anchor nodes. [4] describe agreement properties of localization techniques in more detail. There are other localization techniques which are used in other networks, the most common being RF fingerprinting. In this method, a map based on radio signals is created. In the first phase (offline-phase) we measure RSS on different points on our map and store them in a database. The second phase (real-time phase) measures the RSS of a blind node and compares it to reference points on the map constructed during the first phase, in order to determine the location of the blind node. The problem with this technique is that the RF pattern changes when the environment changes, which means that the radio map is no longer up to date and the accuracy will therefore drop dramatically. A second problem is that these maps are time-consuming to build. Other techniques exist, but they are far less common and unsuitable for WSNs. These techniques will not be used in this paper.

TABLE 1
List of abbreviations

WSN	Wireless Sensor Network
WSNs	Wireless Sensor Networks
RSS	Received Signal Strength
RSSI	Received Signal Strength Indication
TOA	Time Of Arrival
AOA	Angle Of Arrival
CL	Centroid Localization
WCL	Weighted Centroid Localization
ML	Maximum Likelihood
LS	Least Square
LOS	Line Of Sight
AN	Anchor Node
BN	Blind Node
RN	Root Node
GUIs	Graphical Unit Interfaces

Although many ranging techniques exist, in this paper we

- *P. De Cauwer, T. Van Overtveldt, J. Doggen, J. Bracke and Maarten Weyn are with the University College of Antwerp, Paardenmarkt 92, 2000 Antwerp, Belgium*
E-mail: see http://www.e-lab.be

Manuscript received June, 2009; revised June 17, 2009.

have narrowed them down to RSS-based ranging. RSS-based ranging is founded on the principle that RSS attenuates with distance due to free-space losses and other factors. RSS is generally considered as a bad method because of its high variability due to interference, multipath and shading. Errors can be divided into two categories, environmental and device errors. Environmental errors are due to the wireless channel; these include multipath, shadowing effects and interference from other radio sources. Device errors are generally calibration errors. The most important consideration here is to keep the transmitted power constant, despite inter-device differences and depleting batteries. Environmental errors can also be divided into two parts: rapid time varying errors and static environment dependent errors. The first is due to movement of people, additive noise and interference. This can mostly be modeled as Gaussian noise. As a result, this can be reduced considerably by averaging multiple RSSI measurements. The second type of error is due to the varying properties of the environment, such as multipath and shadowing. Since the layout of the environment and the placement of doors and furniture cannot be known without prior knowledge, this error should be modeled as random.

In order to create an accurate localization system based on RSS, the wireless channel properties and these other degrading effects must be modeled as accurately as possible. All these factors seem to give RSSI measurements a large random factor, thus making it very unpredictable. Even with very good modeling, it is inevitable that errors remain present because of the random factor; thus any good localization algorithm should also account for these factors. The upside of using RSS as a ranging method is that the radio can be used for communication and localization. This makes RSS very interesting because there is no need for additional hardware. Other ranging methods such as TOA, especially combined with ultrasound, usually yield better results, but require additional specialized hardware which adds to the size and cost of a node.

RSS-based localization can be divided into three categories:

- Range-based or fine-grained localization
- Connectivity-based or coarse-grained localization
- RF Fingerprinting, as described in the previous paragraph

We will restrict our algorithms to the first two types to satisfy the ad-hoc requirement of the network.

As WSNs have some unique properties, the algorithms in this paper have been developed or selected with the following goals in mind:

- RSS-based : using this technique no additional hardware is required, thus the cost of a node can be kept low. However, because nodes have an antenna embedded on the PCB, it would be better to have an external antenna as these have a more uniform radiation pattern.
- Distributed and self-organizing: The algorithm should be able to run locally on the nodes to avoid a central processing dependency. This is especially important for WSNs due to the fact that individual nodes and links between nodes are more prone to failure than in a traditional computing environment. Batteries may be depleted and radio links can be obscured.
- Robust: The algorithms should account for localization errors and node failures.
- Receiver-based: The task of localization is up to the blind node so that the network scales well.
- Responsiveness: The localization latency needs to be kept as low as possible. Mobility is fairly limited in WSNs as most nodes have a static position; however, certain nodes can be mobile, so this factor needs to be accounted for as well.
- Energy usage: Given the sparse amount available, processing and communication needs to limited. Unfortunately, this means that certain applications are not suitable for WSNs because of the high computational requirements and the lightweight microcontroller that drives the nodes. Communication between nodes needs to be limited as well because the radio requires much more power than the microcontroller.
- Adaptive: We want our algorithm to be adaptive to the number of ANs and the density of the network. If the density or the number of ANs rises, the accuracy should improve. The algorithm should thus benefit from the high density of the WSN. The algorithm should still perform well with a low network density and AN ratio.
- Multihop localization: Nodes not in range of an AN should still be able to localize themselves by the use of other BNs, this is referred to as cooperative or multihop localization [5], compared to single-hop localization, where only blind nodes in range of enough anchor nodes can determine their position.

The main contributions of this paper are:

- We present an overview of the existing algorithms and literature.
- We compare three algorithms with quantitative measurements.
- We researched the influence of the orientation of a node compared to a node with an external antenna.
- We present Senseless, a software framework to manage WSN localization.
- We interface this framework to Scala, a localization middleware project.

The rest of the paper will be organized as follows: chapter two summarizes related work, and provides an overview of the suitable algorithms and work that contribute to building a good algorithm; chapter three presents our software framework; chapter four presents the ranging and the various algorithms that we have implemented and tested framework; in chapter five the methods and results are presented and we conclude in chapter six.

2 Related Work

The amount of literature on this topic is quite substantial. It becomes more manageable if we limit ourselves to three categories: ranging algorithms, location estimation and frameworks. Limited surveys on this topic do exist; however, they fail to point out a superior algorithm and provide few quantitative comparisons, so further investigation is required.

Sum Dist is a distributed multihop algorithm that makes use of the hop count as a primitive distance metric [6]. The distance to the anchor nodes is determined by simply adding the ranges (RSS) encountered at each hop during the network flood of beacon messages by the anchor nodes. The downside of this algorithm is that the range errors rise exponentially when the beacon message travels over multiple hops. Thus, in large networks with little anchor nodes, it will lead to poor ranging. A good alternative is DV-hop. It makes use of the topological information; counting the hops instead. When the topology of the network is very irregular, the ranging will be very inaccurate because of the high variance in hop distance. The anchor nodes broadcast a beacon message to the blind node, which will forward the message to the blind nodes that are out or range of the anchor nodes after filling in the hop count as the path length. The paper ”RSS-based location estimation with unknow pathloss model” [7] dynamically estimates the distance-power gradient; parameter of the radio propagation pathloss model [8]. It adapts automatically to the environment, thus eliminating the need for extensive channel measurements. This leads to a more accurate conversion of RSS to distance. Sorted RSSI Quantization [9] is a connectivity based algorithm that uses hopcount and RSS as a ranging method. It multiplies a hop by the radio range or a chosen distance. It sorts the obtained RSSI and applies a quantizer that represents a level of range in the hop. This makes the algorithm insensitive to RSSI errors.

MoteTrack[10] is decentralized location tracking system based on RF. It is similar to RADAR [11], but does not rely upon back-end server or network infrastructure. The location of each blind node is computed using a RSS signature from the anchor nodes to a database of signatures. This database is stored at the anchor nodes themselves. Cricket [12] is decentralized and uses RF and ultrasound to determine the location of a blind node. Anchor nodes broadcast beacon messages, together with te RF message, it will transmit a ultrasonic pulse. Blind nodes listen to beacon messages and upon receipt, they will listen to the corresponding ultrasonic pulse. A distance to the transmitting anchor node can be estimated with that pulse. A noteworthy survey is [6] by K. Langendoen. This survey describes three algorithms:

- Ad-hoc positioning by Niculescuand Nath [13],
- N-hop multilateration by Savvides et al. [14],
- Robust positioning by Savarese et al. [15].

These algorithms are fully distributed algorithms; they require no central processing node and are designed towards multihop localization. The survey concludes that no single algorithm performs best under different circumstances. Robust positioning works best when no or very bad ranging information is available. Ad-Hoc positioning only works well when the ranging error is very low ((¡)20%). The N-hop multilateration is to be preferred in other situations.

This survey identifies a common three-phase structure in these localization algorithms. The first phase determines the distances between blind and anchor nodes. Note, however, that this does not mean that a specific ranging method, such as RSS, should be used,. The second phase derives a position using the RSS from the anchor nodes. These two phases are roughly equal to what was described in the introduction. Finally, there is a third phase called the refinement phase, where the positions are refined through iterative measurements.

Another comparison is given in [16] by Zanca et.al. This paper compares four algorithms:

- Min-Max [6][17]
- Multilateration [6][17]
- Maximum Likelihood [18] by Patwari N.
- ROCRSSI [19] by Liu C.

A brief introduction to the radio channel is provided. The absolute ranging errors of the algorithms are presented with the number of anchor nodes as a parameter. The authors conclude that ML provides superior accuracy compared to the other algorithms when the number of anchor nodes is high enough. Interestingly, despite its simplicity, Min-Max achieves reasonable performance. This is probably due to the fact that it localizes the node in the center of the estimated area. The authors also note that a good radio channel model is required to obtain a relatively high accuracy. The algorithms presented in this paper are one-hop algorithms; they can only localize nodes in reach of enough anchor nodes.

3 Framework

We have developed a software framework, Senseless, which provides a common data interface to the WSNs and GUIs. It also controls the data flows between the WSNs and GUIs, and stores this data in a database for later retrieval. The system is capable of working with different algorithms and new algorithms can be easeliy added. If there are three anchor nodes available, we can work with range-based algorithms, thus obtaining a better accuracy. If, on the other hand, only one anchor node is available, a connectivity-based algorithm can and must be used. We will use this system to test the different localization algorithms and analyze the RSS data.

Senseless has a Model-View-Controller (MVC) design, as in Figure 1; the system is divided into three different parts each with different tasks. The separation of these responsibilities enhances the modularity of the system. The details of this design are as follows:

- Model: This layer defines the representation of the information which the application works with. The data is stored in a MySQL database.
- View: Information can be accessed and controlled via this part. User interfaces are defined in this layer. The view does not process data.
- Controller: It processes and uses polling to react to events, mostly caused by the actions of the user and the data delivered by the WSN.

The advantage of this design pattern is that we can easily add views and models without changing the whole system.

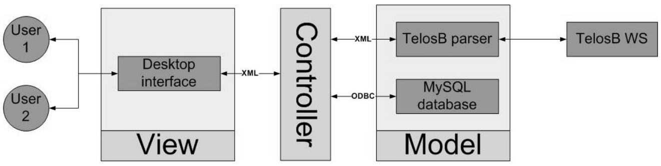


Fig. 1. The Model-View-Controller design

3.1 Functionality

The system consists of four main parts:

- WSN
- Database
- Graphical Unit Interface (GUI)
- Controller

3.1.1 WSN

The WSN consists of Telos rev.B nodes, which have the following specifications:

- TI MSP430 microcontroller with 10kB RAM
- IEEE 802.15.4 compliant CC2420 radio: it supports eight discrete power levels and 16 channels
- Integrated temperature, light, humidity and voltage sensor
- TinyOS 2.X compatible
- Programmable via USB interface

- Integrated antenna

Each node fulfills one of the three different roles:

- Root node (RN): this node receives data from the rest of the network , and acts as a bridge between the WSN and the rest of the framework. The root sends these messages to the controller via an XML parser. It also receives commands from the controller and disseminates these to the right node.
- Anchor node (AN): this node has a known location, and broadcasts a message with its ID to the blind nodes and anchor nodes for calibration. The node also transmits its sensor data to the root with the sensor message.
- Blind node (BN): this node has an unknown location and receives broadcast messages from the anchor nodes. The node uses these messages to determine the RSS, and transmits the RSS together with the ID of the anchor node to the root with the location message. The blind node also transmits its sensor data to the root node with the sensor message.

There are three different data messages which are collected from the wireless network:

- Sensor messages, which contain the data collected by the sensors of the nodes.
- Location messages, which contain data relevant to locate the blind node.
- Status messages, which contain data that represent the status of the node transmitting the message.

3.1.2 Database

We implemented a MySQL 5.0 database to store the data generated by the system, but any ODBC-compliant database can be used.

3.1.3 GUI

The user interfaces provides us with the ability to easily control and monitor the WSN. Rapid deployment was the key motivator to build this component. Using this, the user can set several node parameters, with a focus on localization: sampling period, coordinates (in meters), inactive/active, blind/anchor and the leds. This can be done in a few seconds. In contrast to manually hardcoding every single node of the network, which can be very time-consuming and sometimes impossible due to the fact that TinyOS-programmed nodes and other types of nodes usually provide very little to no user interaction.

3.1.4 Controller

The controller is the core of our system. The controller is programmed in C using the .NET 3.5 framework. It is divided into several class library's and a single Windows Forms project.

It has four main functions:

Firstly, the controller acts as a gatekeeper to the database, ensuring that all data is stored in the correct table and is of the correct type. This is especially important for WSNs as a plethora of hardware platforms exist. These platforms have different data types and can have a different endianness.

Secondly, the controller is also a central gathering point for all the data. By using the controller in our framework, every other component can use a single data interface and should only be aware of the location of the controller.

Thirdly, an interface to Scala is provided as well. Scala is a middleware for location systems. Scala and the interface will be described in the designated section. Finally, the controller implements the centralized versions of our localization algorithms. The user can instruct the controller to use an algorithm via a simple Windows Forms GUI.

WSN vs Controller

The communication between the WSN and the controller is done with an XML parser, which translates the messages of both sides into XML and back into an internal format. The root node of the WSN receives all the messages (sensor, location and status) from the nodes and forwards these to the controller, or if the controller needs to pass a command on to the root, it will be forwarded to the root. With the help of the dissemination protocol, the command is transmitted over the WSN.

Controller vs GUI

The communication between the controller and the GUI also happens with an XML parser, which translates the messages that need to be exchanged. Firstly, the GUI displays the data from the WSN, thus a request needs to be send to the controller. The controller receives the request, and gets the data out of the database and sends it to the GUI. Secondly, the GUI is used to control the WSN. If, for example, you want to change the sample rate of the WSN, then a command will be sent to the controller which forwards it to the root node of the WSN.

Controller vs Database

The communication here makes use of ODBC (Open Database Connectivity), which is an universal database interface. By using this interface, the controller does not need to worry about the database that is used. Stored procedures are used instead of full SQL syntax.

3.2 SCALA

Scala is a TETRA [20] project which aims to shorten the gap between localization technologies and possible end-user applications. The goal of this project is to ease the development of location aware applications by providing these applications with a common interface to the location technologies.

From a technical point of view, Scala adapts the interfaces of the existing localization technologies into a common interface. Doing so, the user can receive location information transparent of the underlying technology. Another feature of Scala is the fusion of location information. By combining the information received from different localization technologies, a more accurate and robust position can be determined.

The Senseless framework provides an interface to SCALA. The framework will plug into SCALA as an engine. Doing so, our algorithms become accessible to a variety of applications. The interface with Scala is performed in the controller via WCF [**?**]. This interface supports the exchange of data about the nodes registered with our framework, the maps on which these nodes are positioned; including images of the maps, the position of these nodes and finally the data coming from their sensors.

The latter two support two types of communication are available: polling and event-based. By polling the engine, information is requested only when needed, as opposed to event-based communication where, Scala subscribes to information coming from the framework. The interface is loosely based on the ANSI Rtls API [21]. The API documents a polling-based system. Possible data fields are specified and a method of filtering data is documented as well.

4 Localization

Localizing the position of a node happens in 2 phases: the first phase is the ranging: the estimation of the distance from an anchor node to the blind node. This is done with a propagation model: translates the RSSI to distance. This model can be made more accurate if you apply a calibration process. The second phase is calculating the node with an algorithm.

4.1 Propagation model

There exist a number of propagation models:

- the Rayleigh fading model[22]: does not account for a Line Of Sight (LOS) component
- Rician distribution model[23]: the determination of the model parameters is difficult
- Floor attenuation factor propagation model[8]: manual determination of model parameters
- log-normal-shadowing model[8]: the most simpel model

We use the log-normal-shadowing model for the ranging. It is the most widely used signal propagation model and the determination of the parameters is simple and can be obtained dynamically with Least Squares (LS).

$$RSS(d) = PT - PL(d0) - 10 \times n \times \log(\frac{d}{d0}) + Xo$$

Where, *PT* is the tramitted power, *PL(d0)* is the path loss for the reference distance *d0*, *n* is the path loss exponent (the rate at which the path loss increases with distance) and *Xo* is a gaussian random variable with zero mean and standard deviation *o* dB. The most common reference distance is one meter. In the calibration phase of the WSN, we determine the path loss exponent and the path loss for the reference distance, so that the propagation model is adapted to the environment and thus more accurate.

4.2 Algorithms

Centroid localization is a simple approach for coarse grained localization. All blind nodes calculate their position as the centroid of the anchor nodes within their communication range. This algorithm has a low accuracy because it does not use signal strength to denote the range. A solutions to make CL more accurate is the Weighted Centroid Localization (WCL)[24]. A weight is coupled to each anchor by its RSSI:

$$Weight = \frac{1}{RSS^g}$$

The degree *g* has to ensure that the remote anchor nodes still have impact on the position determination. In case of a very high *g*, the calculated position moves to the closest position of the anchor node and the positioning error increases.

Min-Max is a popular and a very easy algorithm to implement. Anchor nodes, that are in range of the blind nodes, will create a box around them. This box has the anchor node as the center and has a height and width of twice the estimated distance to the blind node. In an ideal situation, will this algorithms work, but the estimated distance between the node is often underrated. So, one or more boxes will not collide and thus a location can not be determined. In this case, the estimated distance between the nodes is expanded with 10% until all boxes collide.

Multilateration is used in a variety of localization systems including GPS. Multilateration calculates the intersection of three or more circles. If these circles intersect in exactly one point, the coordinates of this point can be determined by linear equations. This method has one major flaw; the circles almost never intersect in a single point. They do not intersect at all or overlap a part of each other. In more dramatic cases one circle can entirely overlap another circle. This is due to the erroneous ranging. The range is over or underestimated.

[25] presents two solutions to solve this problem. The first solution is based on non-linear LS . Through an iterative process the range of the circles is changed until a single point of intersection is found. This method can yield fairly accurate results. The downside however is that it is computationally intensive. The second solution is much simpler and requires that the circles overlap. Given x circles, the algorithm determines the x-nearest points. The centroid of these points is taken as the result. The second solution is simpler to implement but has the requirement that the three circles should overlap a part of each other. We present a simple solution to this problem. There are three cases in which the range of the circle should be modified:

- The circles are too far from each other and do not intersect
- One circle completely overlaps another circle
- A circle is completely inside another circle

In each of these cases the range should be adapted accordingly until these conditions are no longer met. The algorithm iterates through all the circles so that all circles are slightly changed instead of drastically changing one circle. The algorithm finally converges to the situation where all circles overlap each other. We chose to implement both, but an easier from of the first solution namely the normal least square[26], this algorithm is much simpler to implement and has a better performance.

5 Method

5.1 Test environment

Our experiments were performed in an indoor an outdoor environment. The indoor tests were done in a living room which houses a considerable amount of electronics including Wifi equipment operating in the same frequency range. People were walking in the room as well to further interfere with the signal. The outdoor test were done on a local basketball court. Nodes were placed at about five to ten meters from the fencing and few people were walking by.

5.2 Antenna orientation

The relative antenna orientation between receiver-transmitter pairs is a major factor in signal strength variability, even in the absence of multipath and shading effects[27]. This is due to the fact that antenna's are not perfectly omnidirectional. Ideally, the radiation pattern of an antenna should be uniform and it should look like a circle (2-D space) or a sphere (3-D space). However, in practice, the orientation of the antenna can influence the signal strength by several decibels. Thus, different antenna orientations can produce different sets of RSS values for the same distances between receiver and transmitter, increasing the position error.

Therefore, it is imperative that the antenna's orientation should be accounted for as well. One possible solution would to use a compass to determine the nodes orientation. Given the antennas radiation pattern the transmitted power can easily be obtained.

A straightforward solution would be to use a more omnidirectional antenna to minimize these effects. For example, Telos rev.B nodes use an onboard antenna. The power received from this antenna can differ by as much as 20dB depending on the orientation. Fortunately, an external antenna is supported. This can be mounted on the circuit board via an optional standard SMA connector.

We tested the difference between a node employed with and without an external antenna.

5.2.1 *Set up*

Two Telos rev. B nodes were set in the outdoor environment. The test exists out of two parts: In the first part, the two nodes are equipped with an external antenna with a gain of 6dBi and placed at a distance of one meter and at a distance of five meters from each other at a height of one meter. This way the ground will attenuate the signal. One node is set as an anchor node and will broadcast beacon messages at a rate of 200 ms. The anchor node was rotated and samples were taken at every 20 degrees. The other node, the blind node sends the RSS readings to the database. Approximately 25 samples were collected at every orientation.

In the second part, only one node is equipped with the external antenna and placed at a distance of one and five meter. This node is set as the blind node and receives ideally the same power in every direction. The other node with an integrated antenna is the anchor node that will broadcast the beacon messages.

5.3 In & outdoor positioning

The following experiment was devised to test the accuracy of our developed localization algorithms.

5.3.1 *Set up*

We placed a total of 10 nodes in both the indoor and outdoor environment. Nine of them are configured as anchor nodes and the last one if configured as a blind node. Every node is placed at a height of one meter. The anchor nodes are placed randomly at fixed locations (in meters):

- Node one: 1.19 ; 6.98
- Node two: 2.00 ; 8.48
- Node three: 3.00 ; 1.50
- Node four: 3.19 ; 6.23
- Node five: 1.19 ; 5.14
- Node six: 4.64 ; 3.88
- Node seven: 4.67 ; 0.00
- Node eight: 2.50 ; 0.00
- Node nine: 0.00 ; 0.00

The blind node will be located at the following locations (in meters):

1) 5.07 ; 8. 48
2) 5.07; 4.63
3) 2.00 ; 1.50
4) 0.00 ; 3.29
5) 3.19 , 6.23
6) 1.19 , 8.48

6 Results

6.1 Antenna orientation

Figure 2 and 3 display the mean and standard deviation of the RSS when rotating the node. Table 2 displays the average standard deviation for each orientation and the standard deviation of the entire dataset. Note that the standard deviation is sometimes too small to be seen on the graph. These results show that RSS readings are more stable when the node is equipped with a more omnidirectional antenna. However the readings with onboard antenna are fairly stable as well. The results also clearly show that the external antenna is more invariant to orientation than the onboard antenna. The results are consistent with the common belief that node orientation influences RSS readings.

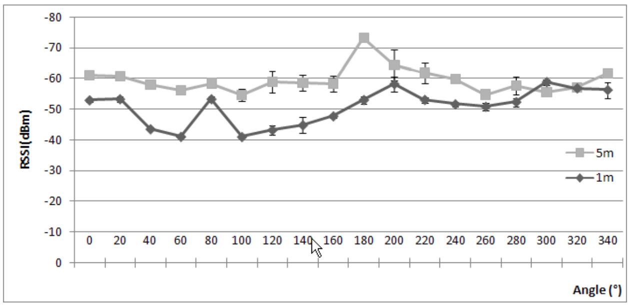


Fig. 2. Average RSSI at a distance of 1m and 5m with one node equiped with an external antenna

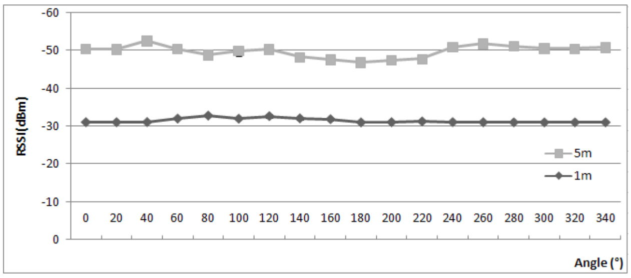


Fig. 3. Average RSSI at a distance of 1m and 5m with the two nodes equiped with an external antenna

6.2 In & outdoor positioning

Figure 4 and 5 show the relative positioning error for respectively the indoor and outdoor positioning. The relative positioning error is measured as the absolute positioning error divided by the average distance to every connected anchor node.

The relative positioning error of the algorithms in the outdoor environment is generally below 140%. The two trilateration algorithms perform really poor in the indoor environment. While Min-Max has a maximun relative positioning error of 68% and keeps dropping when more anchors are added.

The relative positioning error of the algorithms in the outdoor environment is generally below 80%. The trilateration with least squares has good computational performance but a very low accuracy in most cases. A better solution would be to use a more advanced form of least square like the weighted least square and the non-linear least square. Min-Max has the best accuracy, even under 50%. The results show that the accuracy improves when more anchor nodes are added to the system, but we do notice that the accuracy becomes stagnated when we reach the number of 7 anchor nodes. The trilateration algorithms lags behind in accuracy compared to the other algorithms. The Min-Max algorithm is computationally simple and devilers the best accuracy. Therefore it is our current algorithm of choice.

7 Conclusion

This paper has investigated the core aspects of WSN localization systems using RSS and made an overview of the existing

TABLE 2
Table with average RSSI and standard deviation

Results			
	Average	Stdv1	Stdv2
Onboard antenna at one meter	-50,65	1,144	5,71
Onboard antenna at five meter	-59,50	1,63	4,31
External antenna at one meter	-31,46	0,21	0,64
External antenna at five meter	-49,77	0,90	1,61

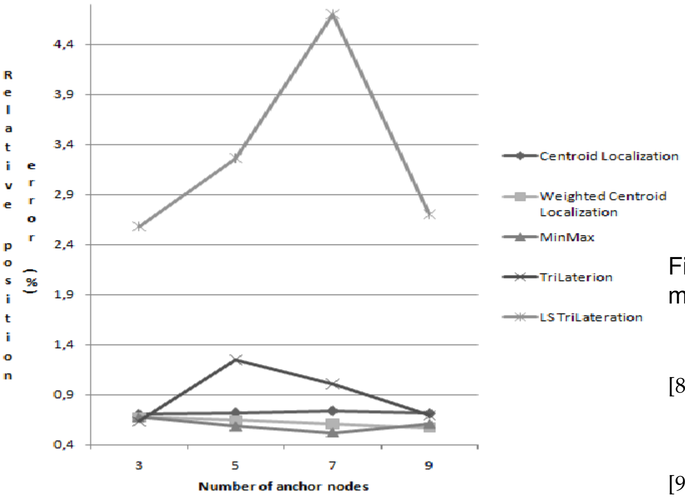


Fig. 4. Average relative positioning error in the indoor environment

localization algorithms. Different types of localization systems were introduced and important properties were discussed as well. An overview of our localization framework and the interface to scalawas given. Five centralized localization algorithms were discussed. The results showed that the relative positioning error was lower then 100% in most cases and that accuracy improves when more anchor nodes are added. We can conclude that the Min-Max algorithm performs the best in both outdoor and indoor environments.

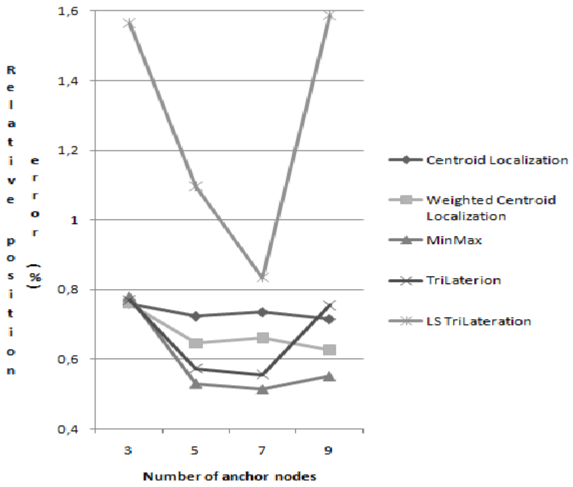


Fig. 5. Average relative positioning error in the outdoor environment

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay *et al.*, "A macroscope in the redwoods," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM New York, NY, USA, 2005, pp. 51–63.

[3] C. Chong, S. Kumar, and B. Hamilton, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[4] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[5] N. Patwari, J. Ash, S. Kyperountas, A. Hero Iii, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[6] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, pp. 499–518, 2003.

[7] X. Li, "RSS-based location estimation with unknown pathloss model," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, p. 3626, 2006.

[8] S. Seidel and T. Rappaport, "914 MHz path loss prediction models for indoor wirelesscommunications in multifloored buildings," *IEEE transactions on Antennas and Propagation*, vol. 40, no. 2, pp. 207–217, 1992.

[9] X. Li, H. Shi, and Y. Shang, "A sorted RSSI quantization based algorithm for sensor network localization," in *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*, vol. 1, 2005.

[10] K. Lorincz and M. Welsh, "MoteTrack: a robust, decentralized approach to RF-based location tracking," *Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 489–503, 2007.

[11] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *IEEE INFOCOM*, vol. 2. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 2000, pp. 775–784.

[12] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM New York, NY, USA, 2000, pp. 32–43.

[13] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267–280, 2003.

[14] A. Savvides, H. Park, and M. Srivastava, "The n-hop multilateration primitive for node localization problems," *Mobile Networks and Applications*, vol. 8, no. 4, pp. 443–451, 2003.

[15] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *USENIX technical annual conference*, vol. 2. Monterey, CA, 2002.

[16] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks."

[17] X. Nguyen and T. Rattentbury, "Localization algorithms for sensor networks using RF signal strength CS 252 Class Project," Citeseer, Tech. Rep., 2003.

[18] N. Patwari, R. O'Dea, and Y. Wang, "Relative location in wireless networks," in *IEEE VTS 53rd Vehicular Technology Conference, 2001. VTC 2001 Spring*, vol. 2, 2001.

[19] C. Liu, K. Wu, and T. He, "Sensor localization with ring overlapping based on comparison of received signal strength indicator," in *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004, pp. 516–518.

[20] http://www.iwt.be/steun/steunpro/tetra/index.html.

[21] http://www.qed.org/Incits/2003/Docs/INCITS025

[22] H. Hashemi, "The indoor radio propagation channel," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, 1993.

[23] S. Rice, "Mathematical analysis of random noise," *Bell Syst. Tech. J*, vol. 23, no. 3, pp. 282–332, 1944.

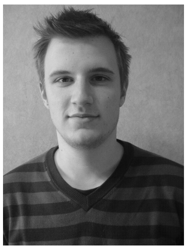
[24] S. Schuhmann, K. Herrmann, K. Rothermel, J. Blumenthal, and D. Timmermann, "Improved Weighted Centroid Localization in

Smart Ubiquitous Environments," *Lecture Notes in Computer Science*, vol. 5061, pp. 20–34, 2008.

[25] A. Kaminsky, "Trilateration," 2007.

[26] A. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 24–40, 2005.

[27] D. Lymberopoulos, Q. Lindsey, and A. Savvides, "An empirical characterization of radio signal strength variability in 3-d ieee 802.15. 4 networks using monopole antennas," *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3868, p. 326, 2006.



Student 1 Peter De Cauwer is currently working on his Master Thesis after graduating as Bachelor in Applied Engineering: electronics-ict in 2008 at the University College of Antwerp.

Student 2 Peter De Cauwer is currently working on his Master Thesis after graduating as Bachelor in Applied Engineering: electronics-ict in 2008 at the University College of Antwerp.

If you wish to cite this paper, please use the following code: Peter De Cauwer, Tim Van Overtveldt, Jeroen Doggen, Jerry Bracke and Maarten Weyn, WSN localization with Senseless framework, Master's Thesis,Department of Apploed Engineering, University College of Antwerp, Belgium, June 2009