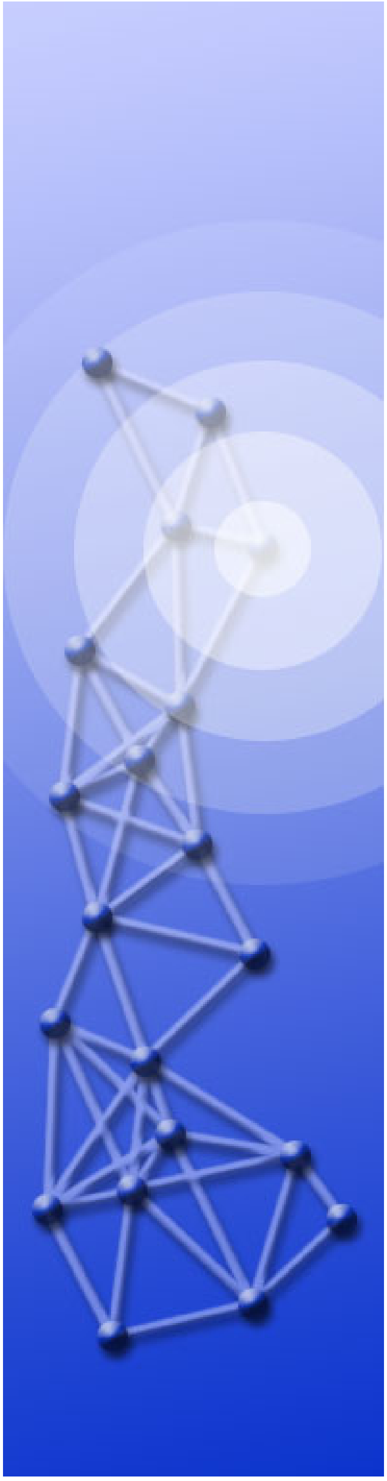


Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Positioning in mobile Ad-Hoc Networks

Semester Thesis
presented by

Stefan Tschumi
ETH Zürich, Switzerland

Supervisor: Jan Beutel, Prof. Dr. Lothar Thiele

July 2002

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Acknowledgement

Before you keep reading this Semester Thesis, I would like to introduce to you all persons who have encouraged and helped me to succeed this thesis.

First of all I would like to thank my friend and class-mate Ernesto Wandeler for all his help and advices during the implementation of our new developed Network Simulator. Without his ideas and knowledge, the Simulator wouldn't look and function the way it does now.

Then I also would like to thank my supervisor Jan Beutel. He let me find my own way through the thesis, but knew always some good advices or help me out, when I was stuck and didn't get any further.

I also like to mention and thank the Services Group of the institute TIK, who organized the Semester Thesis very well and who supported me with all the material that was needed. For carrying out the measurements, I also like to thank Hansruedi Benedickter, who was very cooperative and helped me out in the absorber room.

And last but certainly not least I would also like to thank my parents for supporting me during my studies at the ETH and my girlfriend Katja for all her encouragement at home.

Table of Contents

Abstract

This Semester Thesis gives in a first chapter a short introduction to mobile ad-hoc networks with its applications, to the Bluetooth technology and to positioning in ad-hoc networks.

In the second part, the Receiver Signal Strength Indicator (RSSI) is examined and analyzed, if this value can additionally be used for measuring a distance between two nodes. Unfortunately the result shows, that the requirements on the RSSI are insufficient for the use of calculating the distance between two network nodes. With the used hardware, the RSSI value can only be used to make a rough quantization of a distance between two nodes in near and far.

Through these measurements and earlier conducted results by J. Beutel, the effect of quantized distances to the precision of positioning services in mobile ad-hoc networks was examined by E. Wandeler in his Semester Thesis. To be able to simulate such behaviour, a GUI to his new developed MANET Simulator Framework was implemented. A detailed description of the Use of the Network Simulator with the GUI frontend is given in the last chapter.

Chapter 1

Introduction

In the last decades, the success of consumer products like Personal Computers, cell phones, cordless phones, personal digital assistants and many more has been based on continuous cost and size reduction. The transfer of information between all those devices however relied mainly on cables. Therefore the industry has to come up with a new technology to allow all devices to communicate in a more flexible and mobile way and to eliminate the cables to transfer information over a short distance.

A short overview to mobile ad-hoc networks, to the Bluetooth technology with its general applications and to positioning is given in this introduction.

1.1 Mobile ad-hoc networks

Mobile networks have become increasingly popular and are used in many different ways. Today, there are two variations of mobile wireless networks.

The first one operates with a fixed gateway. That means, that every node in the network knows exactly over which gateway it can communicate with the rest of the network. However direct communication between two nodes is not possible. Therefore, this technology is called *infrastructured network*. The wireless local area network (WLAN) is a typical application of an infrastructured network.

The second type of mobile networks is the *ad-hoc network*. Ad-hoc networks don't use any infrastructure like central servers to initialize or organize the communication. Ad-hoc networks have to initialize themselves, and can therefore arise whenever some nodes are able communicate together. Some nodes act like routers, that maintain routes to other nodes in the network. Hence, in ad-hoc networks, there is no fixed gateway, and each node can communicate with every node that is in its transmission range. Connection establishment and management has to be done dynamically by each individual node, so that a node can leave or enter the network whenever it is desired. Therefore the entire network topology is dynamic. If a node wants to communicate with another node that isn't in its transmission range, it has to find other hosts, that are willing to forward its message to the receiving node.

Because ad-hoc networks change very fast, adaptivity, flexibility and self-configuration are key features of mobile ad-hoc networks.

A technology, which can be used to build a mobile ad-hoc network is the Bluetooth standard, that is described in the next paragraph.

1.2 Bluetooth

Bluetooth is a new standard to replace the cables connecting portable and fixed electronic devices over a short-range radio link. Key features are robustness, low complexity, low power, and low cost [1]. Because of all those characteristics, Bluetooth is nowadays becoming widely accepted. The Bluetooth concept supports dynamic network topology and dynamic connection establishment. Therefore the Bluetooth standard is possible technology to build mobile ad-hoc networks with [3].

Bluetooth operates in the unlicensed Industrial-Scientific-Medical (ISM) band at 2.4 GHz. This spectrum is open to the public without having to pay for licences and is available over most of the world. This band is also used e.g. by cordless phones or microwave ovens. To reduce interference and fading, Bluetooth is using frequency hopping. The frequency changes with a nominal hop dwell time of 625 μs. The short dwell time only allows the transmission of a single packet.

The Bluetooth standard provides a point-to-point or a point-to-multipoint connection. We talk of a piconet, if there are two or more nodes connected to one master. The master defines the phase and the particular hopping sequence to which all slaves have to synchronize. However, in such a piconet, only seven slaves can be attached. A network can grow anyway, if a slave belongs to more than one piconet, or if a slave starts a new piconet itself, where it becomes the new master. In those cases, the node remains locked (passive mode) in the first piconet. In this mode, the node still saves the phase and the hopping sequence of the first piconet. Without getting in the active mode, it can not communicate with the master of the piconet. In one piconet, there are at most only seven active slaves, but up to 256 slaves in the passive mode. By definition, a unit can not be the master in two or more piconets, since the master parameters specify the piconet frequency hopping channel.

Typically many independent networks with a limited number of units overlap in the same area. This will be indicated as a scatter ad-hoc environment, or is just called a scatternet. The following figure shows the different possible connections.

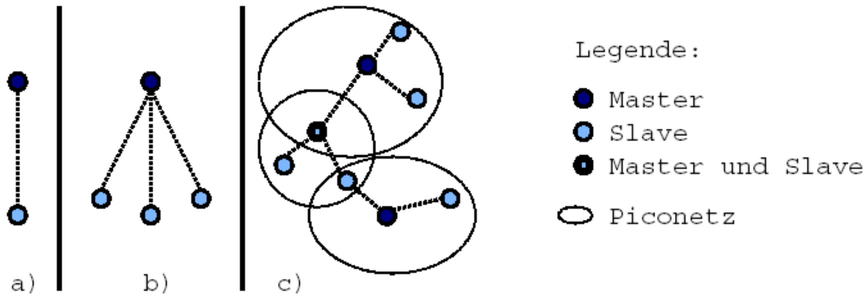


Fig. 1-1: a) Point to Point connection b) Piconet and c) Scatternet

Some technical data about the Bluetooth standard are listed in the following section.

Technical data:

- 2.4 GHz band, bandwidth 79 MHz
- Frequency hopping, 1600 changes per second
- 10 – 100 m transmission range
- Maximal bandwidth 1Mbit per second
- Radiation power 1mW

1.3 Applications for mobile ad-hoc networks

The mobile ad-hoc network technology can be used for many different applications. The following section gives a short insight, where and how such networks can be applied.

Mobile ad-hoc networks come to use whenever network infrastructure such as the internet or a Local Area Network (LAN) is not available or may be undesirable due to reasons such as cost or convenience.

As Jan Beutel writes in his diploma thesis [4], that there are three main groups in the broad application spectrum of mobile ad-hoc networks, that can be identified: A smart environment, group collaboration and sensor tags.

A smart home environment could contain many sensors for temperature and light, and would connect all electronic devices and appliances like a smart refrigerator or microwave oven in each room and in the whole house.

To the topic 'group collaboration' there are many different examples. Ad-hoc networks could be used for example during conference meetings. It is very attractive to have instant network formation to share files and information without the presence of fixed base stations and systems administrators. For emergency services or military purpose, these ad-hoc networks could also be used. For example to broadcast information to all team members or if one team member can communicate with a small handheld to any other member of the team without the need of any global coordination or fixed infrastructure.

The third topic of ad-hoc networks are sensor tags. Sensors could be placed in any part of a building. These sensors could then form an ad-hoc network, certainly also with other devices present in the building, and share information.

1.4 Positioning

An important service in ad-hoc networks is the estimation of the position of a certain device. Because a lot of information shared in the network can only be used, if the position of the node is known. For example emergency calls with mobile phones could be ameliorated. When the ambulance gets the exact position of the caller, it can react more quickly and more precise. This Semester Thesis deals with the positioning of Bluetooth nodes in ad-hoc networks.

A famous service to calculate a position is the Global Positioning System (GPS). This positioning service can well be used outdoors, however if one likes to calculate the position of a device inside a building, this service doesn't provide high accuracy. The idea of positioning in ad-hoc networks uses some nodes, which know their position from GPS or traditional measurement. Such nodes are called anchors. From the position of these anchors, other nodes in the network can then estimate their position by measuring the distance to all anchors in their transmission range. If a node was able to estimate its position, it is called a settled node, and can then also provide its position information to other nodes to get settled.

There are a variety of ways in which position can be derived from the measurement of signals. The most important measurements are propagation time, time difference of arrival (TDOA), angle of arrival (AOA), carrier phase, and Receiver Signal Strength Indicator (RSSI). To be able to calculate its position, a node has to get a signal from at least three nodes (in 2D) or four nodes (in 3D), that know or have already calculated their position before. If the density of nodes is high and more measurements can be made than required to uniquely define the position, for example a least squares approach can be used to combine all the measurements into a more accurate position estimate.

For positioning, the key functionalities of a Bluetooth device are the following:

- Inquiry: With this functionality, a Bluetooth device can search the environment to detect all other devices in its range.
- Open a connection: Once a device is detected, the device can open a connection with another Bluetooth device and ask for some specific data.
- RSSI: (Receiver Signal Strength Indicator) On an open connection, the RSSI can be measured to get the signal strength of the other Bluetooth device. This Indicator was examined during the carried out measurements in this semester Thesis.
- Link Quality: The Link Quality can be measured for an open connection between two nodes.

Chapter 2

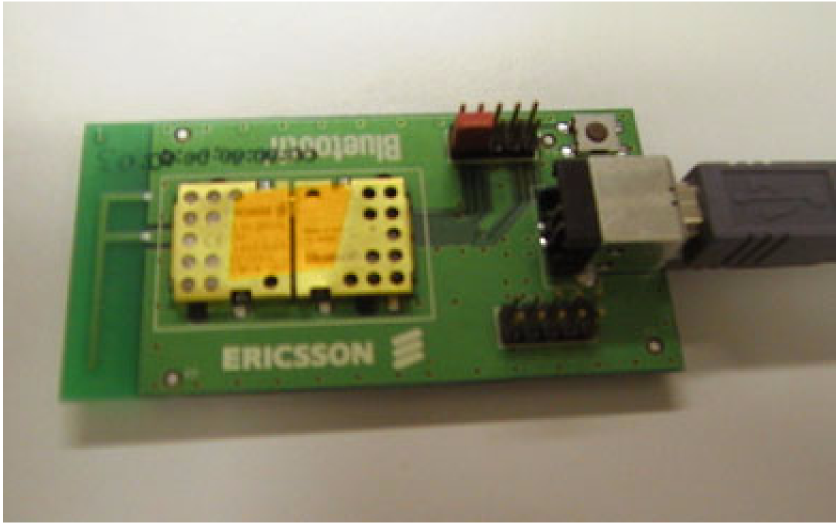
Measurement of the RSS Indicator

The newer versions of Bluetooth nodes are equipped with two indicators. One is the RSSI, the Receiver Signal Strength Indicator, and the other one is a indication of the Link Quality on a connection of two nodes. This Semester Thesis analyzed the RSS Indicator for measuring a distance between to nodes. This chapter gives a detailed description of the carried out measurements of the RSS Indicator. Two different test series have been performed: a test series carried out indoors in a special absorber room, and one series outdoors on the roof of the ETZ. In the section 'Experimental Setup', the used hard- and software, the location and the measurement setup are explained in detail. In the following part, all graphs of the results from the outdoor and indoor test series are given and in the discussion all results are interpreted and evaluated. In the outlook at the end of this second chapter, the future development is explained.

2.1 Experimental Setup

2.1.1 Hardware

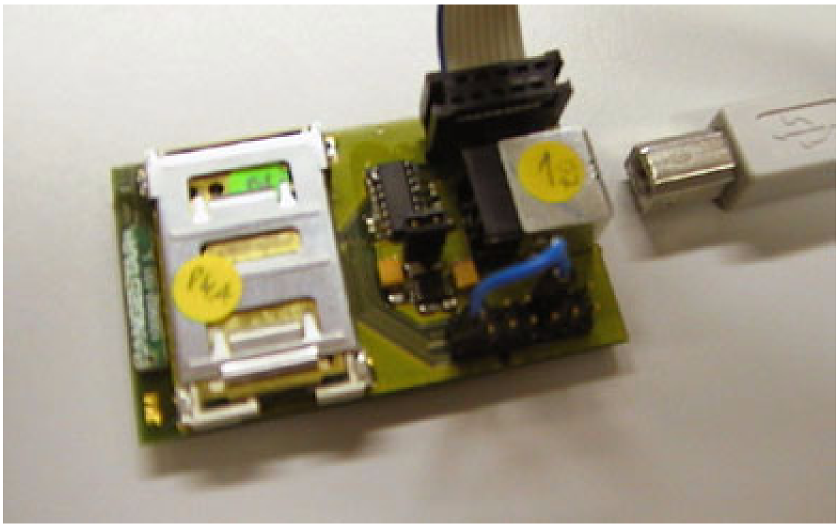
The Hardware that was used to carry out the measurements was the 'Bluetooth Application Tool Kit'. The following two versions of the Toolkit have been used during the measurements: Ericsson ROK 101007 and ROK 101008. Communication between the module and the host controller is provided via an USB interface for the first module and via UART and PCM interface for the second module. Detailed specification of the hardware can be found in the data sheet provided by Ericsson Microelectronics [6]. In pictures 2-1 and 2-2 the two used Toolkits are shown.



Picture: 2-1 : Bluetooth Application Toolkit ROK 101007 with USB interface

In the following list, the most important data about the Hardware is specified.

- Ericsson Bluetooth module ROK101007 and ROK 101008
- UART interface for data transmission (max data rate 460 kb/s)
- USB interface for data transmission (max data rate 12 Mb/s)
- PCM interface for voice transmission
- All Bluetooth layers from radio up to HCI are implemented in Hardware
- Point to Point operation
- Power supply 5V DC, 200 mA provided via USB-connector
- Average current consumption is 40 mA
- Antenna included
- Frequency range is 2.402 to 2.495 GHz
- Manual reset possible



Picture: 2-2: Bluetooth Application Toolkit ROK 101008 with UART interface

2.1.2 Software

BlueZ - Bluetooth Stack for Linux

In this Semester Thesis, the Bluetooth Stack BlueZ was used. BlueZ is an open source stack and can be downloaded from [7]. For the installation the "How To" written by J. Beutel [8] is very useful.

The BlueZ Packet includes some helpful applications, with which the Bluetooth nodes can be attached, configured and tested. If an UART node has to be attached, the following command is necessary:

```
hciattach /dev/ttyS0 ericsson 57600 flow
```

The USB node is attached automatically. Once the nodes are attached, the hci-config module is used to configure and operate the nodes. To start this module, the following command can be used:

```
hciconfig hciX up
```

The letter X represents the logical umber of the node, under which it is attached.

RSS Indicator

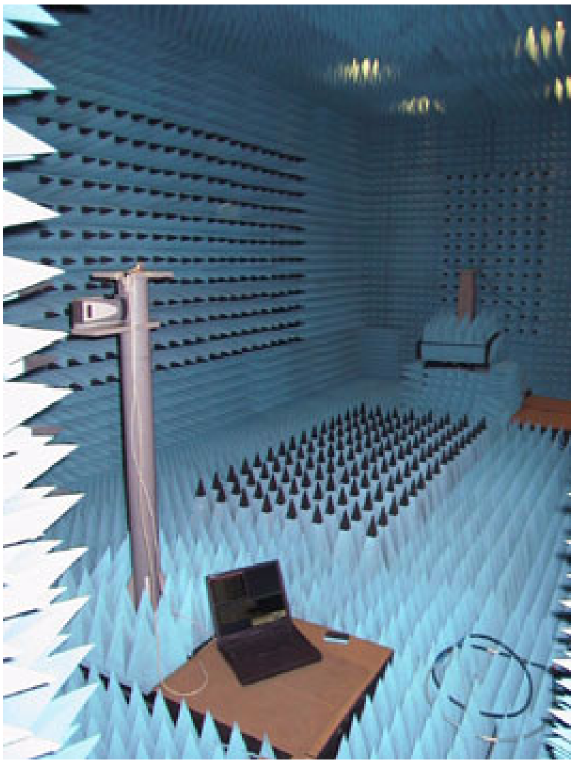
With the RSS Indicator the transmission output power level of each Bluetooth node can be controlled. When some nodes are close together, the output power of the two nodes has to be decreased, so that other nearby nodes are not disturbed. Therefore a node can measure its receiver signal strength and determine if the transmitter on the other side of the link should increase or decrease its output power level.

As written in the Bluetooth Specification [1], the Read_RSSI command reads the value for the difference between the measured Received Signal Strength Indicator (RSSI) and the limits of the Golden Receive Power Range, which has lower and upper threshold level boundaries. The lower threshold level corresponds to a received power between -56 dBm and 6 dB above the actual sensitivity of the receiver. The upper threshold level is 20 dB above the lower threshold level to an accuracy of +/- 6 dB. Any positive RSSI value returned by the Host Controller indicates how many dB the RSSI is below the lower limit, any negative value indicates how many dB the RSSI is above the upper limit. The value zero indicates that the RSSI is inside the Golden Receive Power Range.

The idea of using the returned RSSI to get an approximate value of the distance between two nodes, to calculate a nodes position, is examined in this Semester Thesis. Because the signal power decreases with distance, the RSSI is expected to increment to indicate a necessary increase of the output power of the other node.

2.1.3 Measurements setup

To realize the indoor measurements, one workstation and one laptop were used. The workstation was placed just outside the room with the Bluetooth node connected with a long USB extension cable. Like that, it was still possible to close the door of the absorber room to provide ideal environment. The second Bluetooth node was attached to a laptop, which could be moved around to certain places inside the room. The following picture shows the Bluetooth node attached to the laptop in the absorber room. The absorber room will be explained in the next part.



Picture: 2-3: Moveable laptop with Bluetooth node attached

To be more mobile and flexible, two laptop computers were used during the outdoor test series. Like that one Bluetooth node was attached to each laptop. Both laptops were put on a small trolley and could be easily moved to different places on the roof.

Once the two Bluetooth nodes were attached and configured with the computers, a connection between the nodes was built up. With the Host Controller Interface (HCI), an inquiry of a nodes environment can be started, to detect all nodes that are within its transmission range. The following command starts an inquiry of the whole transmission range of the attached node hci0 during 5 seconds. The result is found in the third line, where the address and clock offset of the detected node is given.

```
[root@pc-3632 src]# ./bluez_tools -i hci0 inq 5
Inquiring ...
00:00:00:00:00:55clock offset: 0x360cclass: 0x000100
```

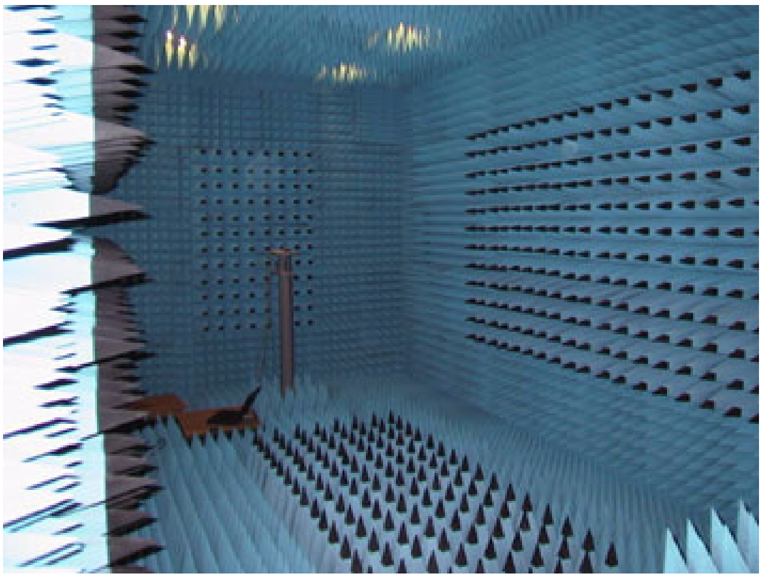
Through the following connection command, a connection from node hci0 to the node with the address 00:00:00:00:00:55 is built up. The command "hcitool con" shows all connections that are established at the moment. Like that, the configuration can be checked.

```
    [root@pc-3632 src]# ./bluez_tools -i hci0 cc :::::55
    [root@pc-3632 src]# ./bluez_tools -i hci0 con
    Connections:
< ACL 00:00:00:00:00:55 handle 1 state 1 lm MASTER
```

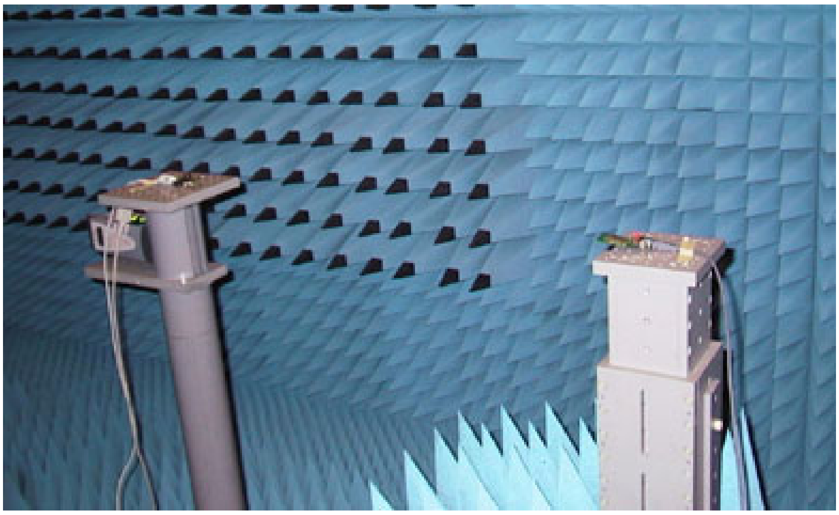
With the result written on the last line, it can be seen, that a connection between the Master (hci0) and the Slave (00:00:00:00:00:55) is established.

2.1.4 Indoor location

The indoor measurements were carried out in a special room of the Laboratory for Electromagnetic Fields and Microwaves. This room is supplied with a carbon loaded urethane foam around all walls, that absorb all incoming waves. Through this foam absorber, the return loss of all waves with a frequency of 1GHz and up is at least 40dB. The room has a length of around six meters; therefore in the length only limited measurements could be carried out. That was one reason a second test series was realized on the roof on the ETZ. Picture 2-4 shows the interior of this chamber, and Picture 2-5 illustrates the two Bluetooth nodes taped on top of two posts.



Picture: 2-4: Interior of the absorber room



Picture: 2-5: Two Bluetooth nodes on top of two posts inside the absorber room

2.1.5 Indoor Measurement

Once a connection between two nodes is established, the RSSI can be measured. To get this Indicator, the following command was executed on different distances. The node hci0 asks to get the RSSI on the connection to the node with address 00:00:00:00:00:55. On the second line, the RSSI value (13) is returned.

```
[root@pc-3632 src]# ./bluez_tools -i hci0 rssi 00:00:00:00:00:55
RSSI return value: 13
```

During the first test series always 20 measurements of the RSS Indicator at a given distance were carried out. The results were written into a file, which was analyzed afterwards. The distance between the two nodes was measured with a laser measurement, which is available in the used absorber room. Like that, the RSS Indicator was measured every 30 cm from 10 cm up to 520 cm.
Once the data was measured and saved in different files, the results were exported to MATLAB and plotted in a graph shown in Figure 2-1.

2.1.6 Outdoor location

To be able to measure the behaviour of the RSS Indicator when the nodes are farther apart, the second test series was carried out on the roof of the ETZ building. This place was chosen, because there is enough space to measure up to 40 m and there is no other building close to reflect the waves. Picture 2-6 shows the measurement setup on the roof of the ETZ.

Picture: 2-6: Laptop on a trolley with the Bluetooth node attached

2.1.7 Outdoor Measurement

During the outdoor measurement, the RSSI was taken 1000 times at each distance. Up to 2 m, every 30 cm a measurement was carried out, then up to 35 m every 50 cm a measurement of 1000 iterations was taken. All results were written into a file which were analyzed with MATLAB.

Picture: 2-7: The location of the second test series on the roof of the ETZ

2.2 Results

2.2.1 Indoors

The result of the carried out measurement in the indoor room is shown in Figure 2-1.

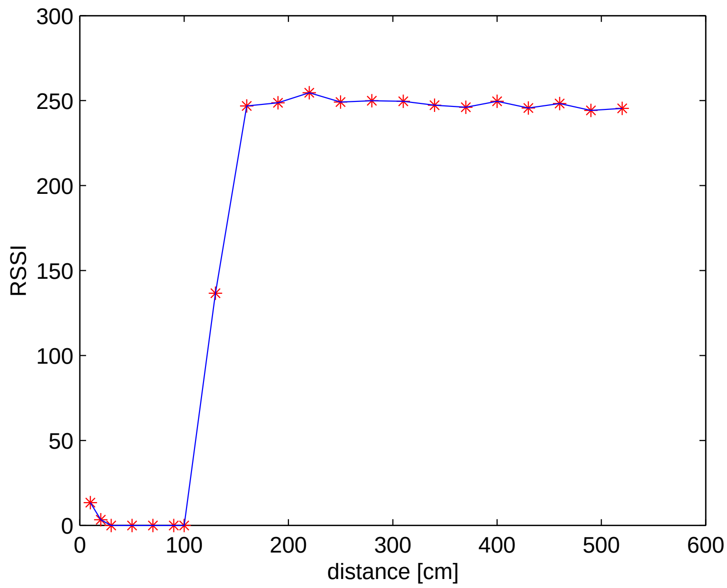


Fig. 2-1: Results of the measurement of the RSSI indoors

Whenever the two nodes are closer together than 100 cm, the value of the RSS Indicator was around zero. At a specific distance of 100 cm, the value raised abruptly to a new level of 250. When the distance was farther increased, the indicator stayed unchanged at this level of 250, with almost no variation. The longest distance that was possible to measure in this absorber room was 520 cm.

2.2.2 Outdoors

Also the result of the outdoor test series shows a similar result as the indoor measurements. The following two figures show the graphs of the outdoor measurement, where the second picture also indicates the variance of the measured values.

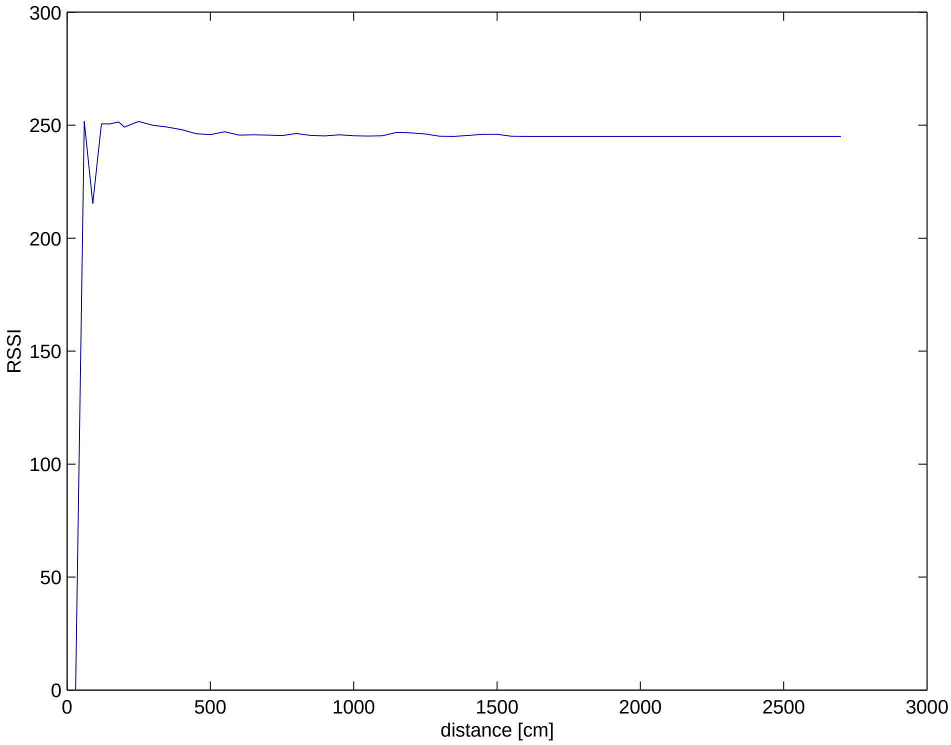


Fig. 2-2: Result of the outdoor test series without variance

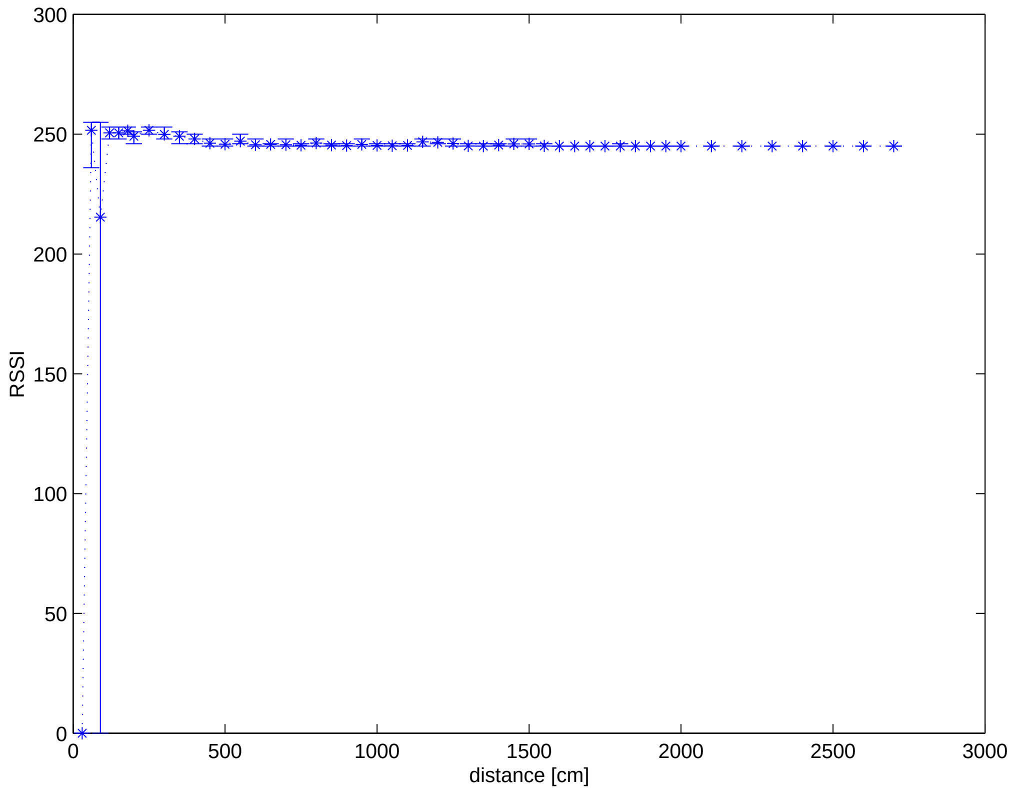


Fig. 2-3: Result of the outdoor measurement with shown variance

The measurements of the RSS Indicator returns a value of zero up to around one meter. Then the value increases rapidly to a new value of 250, where it stays until the connection between the two Bluetooth nodes terminates. The distance, where the nodes lost their connection varied from test series to test series. Sometimes the connection was lost at around 27 m, but sometimes the nodes stayed connected up to 35m.

From the second graph, it can also be read out, that the variance of the results is very small. At each distance, 1000 measurements were taken, but the variance of the values is insignificantly small.

2.3 Discussion

Because the graphs of the two test series have a very similar shape, the measurements of the two test series seem to be plausible. The effect, that only 20 measurements at each distance have been taken during the first test series seem not to have affected the result in a negative way.

However the result of the measured RSS Indicator is not very useful for estimating a distance between two nodes. With the measured behaviour of the RSS Indicator it is only possible to say, wether the two nodes are connected (within transmission range) or not connected. This result can however also be received with the inquiry command. Therefore the RSSI works like a distance quantizor with only one step. (compare Distance Quantizor on page 38)

The requirements on the RSSI are insufficient for the use of calculating the distance between two network nodes. The purpose of the RSSI as specified by Bluetooth is simply to indicate, if the output power level of a Bluetooth node has to be increased, decreased or can stay the same. This is achieved by indicating, if the RSSI is inside, above or below the Golden Device Power Range. Therefore the output power level of the two nodes is only sufficient up to one meter. The steady result of 250 after one meter indicates, that the output power level of the other node should be increased. Because these two used nodes aren't able to adjust their output level, the RSSI indicates to increase the output power level, until the connection determines.

With the existing RSSI, the possible distances between two nodes can be split into 2 - 4 ranges. Therefore a rough quantization in nodes that are near and nodes that are far apart can be made.

2.4 Outlook

As it is written in the discussion above, the RSS Indicator is not very useful for estimating a distance between two nodes. Therefore other indicators have to be taken in consideration. As mentioned in the introduction of this chapter, also the Link Quality on a connection of two nodes could be used to get an estimation of the distance between them. If a connection between two Bluetooth nodes is established, the Link Quality represents the quality of the link between the two nodes and returns a value between 0 and 255. Maybe a correlation between distance and Link Quality can be found.

During future measurements the RSS Indicator should still be experimented in more detail. For example the behaviour of different and larger numbers of Bluetooth Toolkits should be examined and compared. Also newer versions of the Bluetooth Application Kits, that are able to adjust the output power level, should be examined in detail.

It would also be very interesting to measure the output power level of each Bluetooth Toolkit to examine, when a node changes its power level. Maybe the sent signal strength could also be transmitted with the signal. Like that the sent and received signal strength could be compared to calculate a more precise distance between the two nodes.

The RSSI could also be extended, that it can be represented with more than 8 bits. If the RSSI is calculated with 16 bit, larger values could be shown and therefore larger differences could be used to calculate a distance. It should also be possible to get the actual value of the signal strength in high resolution. However this operation is not provided at the moment by the API.

Chapter 3

NetSim Users-Guide

This chapter describes the use of the new MANET Simulation Framework (NetSim) with the Graphical User Interface (GUI). The Network Simulator was implemented for this project in collaboration with E. Wandeler. Detailed description of the design is found in his semester thesis "Analysis of Quantization-Effects in Distributed Positioning-Algorithms for Mobile Ad-Hoc Networks" [5].

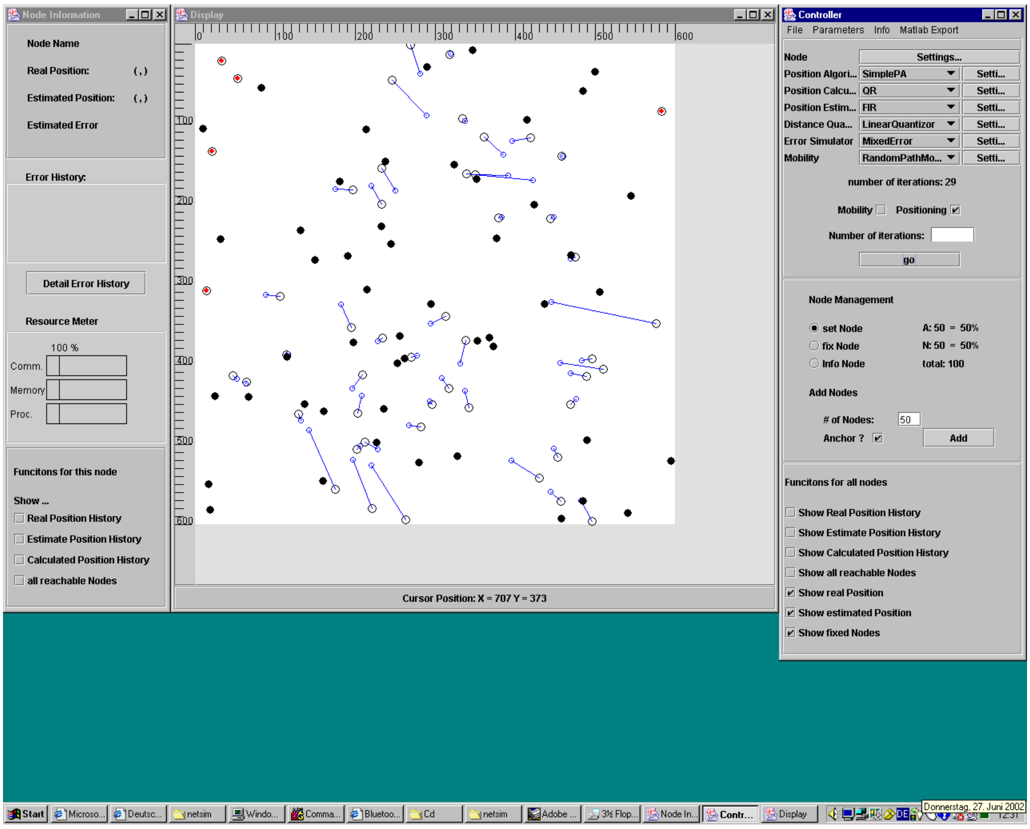
With the new simulator, any kind of different simulations of mobile ad-hoc networks can be performed. Our new developed simulator is designed in a way, that it can easily be extended in the future and that can be parameterized for different use. Also the GUI front-end is implemented that it can easily be adapted to future changes. The whole parametrization of the simulator is implemented using XML. Therefore any adaptations can be made in the external XML file and the NetSim, including the GUI, recognizes and reacts to all changes dynamically.

NetSim is completely written in Java. For the development the latest Java version, J2SDK1.4.0 was used. To implement the GUI, the user interface library 'Swing' was used. Swing has many advantages compared to the older Abstract Window Toolkit (AWT). Besides that Swing has a much richer and more convenient set of user interface elements, it also looks the same way on Windows, Linux or Macintosh.

The next chapters explain the use of the GUI in detail. First a general overview over all different frames is given and in the following chapters each frame is explained in more details. At the end, the future developments are listed.

3.1 Overview

As you can see in Screenshot 3-1, the GUI of the NetSim uses different frames. The three main frames are called Controller, Display and Node Information.



Screenshot 3-1: Overview of the NetSim

With the Controller on the right side, all general manipulations for the whole NetSim can be performed. There you find different menus e.g. to print, save but also to export some information to MATLAB. In the controller you also specify the parametrization, start the chosen algorithm to calculate the positions, add nodes or anchors and you can choose, what kind of information you would like to have drawn in the display.

The Display in the middle is the main frame that shows all actions of every node. The following modules can be drawn, which will be explained in the following chapters:

- real, calculated and estimated position
- real, calculated and estimated position history
- raster to indicate the transmission range

The user can decide, whether he wants to see all reachable nodes, the error history, the estimated position history and/or the calculated position history of every node.

Once a node is selected, all the information about this selected node is shown in the third frame on the left, in the Node Information. Besides the name of the node, you also get its real and estimated position with the calculated error. In a small window the error history is shown and the resource meter indicates the used resources compared to the rest of the nodes. In the Node Information you can also check, what kind of information of the selected node you would like to have drawn in the display.

3.2 Controller

The controller frame on the right side is divided into three parts: a parametrization field, a node management field and a selection field. However first, the use of the different menus of the controller is explained.

3.2.1 Menu

Screenshot 3-2: File Menu

In the controller frame, you can choose between the File, Parameter, Info and Export menu. In the File menu the choices are: New, Load, Save, Save As, Print, Page Setup, Reset and Exit. Those menu entries are explained in the following section.

- New: With the New action, the current display is deleted and an empty display, with no nodes set, appears to start a new simulation.
- Load: If Load is selected, a common window appears, where a saved simulation can be chosen to be reloaded into the display.
- Save, Save As: To be able to load a simulation, it has to be saved first. If Save is chosen, a common window appears to specify the name and the place to save the current simulation. If you modify an already saved simulation, the save option overrides the old version. However you can choose Save As to give a new name for the modified version without loosing the old saving.
- Print, Page Setup: With this selection you can print everything that is shown in the display. The size of the current display window is adjusted in that way, that the complete simulation fits on one page. Smaller simulations are scaled to fill the chosen page size.
- Reset: With the Reset option, all existing nodes stay in the display, however all positioning information is deleted. With this option, the same simulation situation can be restarted several times to analyze different behaviour.

- Exit: And with the last option of the file menu you can close all windows and exit the NetSim.

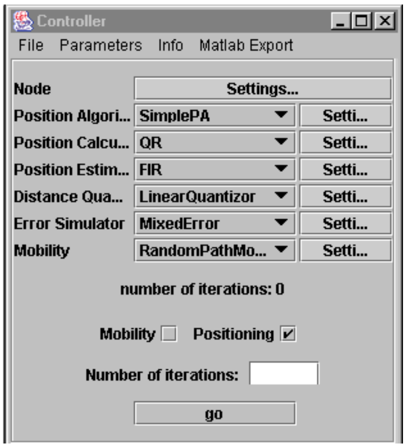
With the Parameters menu in the controller frame, the parameters for all different topics can be set by choosing ‘settings...’. All topics and parameters are explained in the next section 3.2.2. If Reset Parameters is chosen, all parameters will be reset to their default value.

The Info menu can be used to save the information about one node, all nodes or the entire playground to log/log.txt. Make sure, that you choose INFO for all entries in the config/logging.properties.

The last menu is very useful to analyze the results and to plot some graphs. With the export menu all information about one node, all nodes or the complete playground will be exported to MATLAB files. These files can then be loaded and processed with MATLAB by using the following command:

```
load filename.mat -ASCII
```

3.2.2 Parametrization Field



Screenshot 3-3: Parametrization Field

In this first field of the controller frame many parameters about the node, the Position Algorithm, the Position Calculator, the Position Estimator, the Distance Quantizor, the Error Simulator and also about the Node Mobility can be set.
To change from one algorithm to another or to change e.g. to another Position Estimator, the selection field just close to the topic name can be set. To set all details about each topic, click on the Settings button to get a new window with all possible parameters to set.

In the following part a summary of every topic with all possible parameters is given. Detailed description of the parametrization can be read in the semester thesis of E.Wandeler [5], who actually implemented the application.

Node

With the first button, the 'Node settings...', a window to specify the parametrization of all nodes in the playground can be opened. Any changes on those parameters effect all existing nodes in the playground and all future nodes that are set in the same playground. The nodes know two different parameters that can be set.

- range: This parameter defines the transmission range of all nodes.
- historyLength: With this value, the number of entries in the different position histories of all nodes can be set. If the value -1 is set, then the histories can hold an infinite number of positions.

Position Algorithm

With the Position Algorithm module, different algorithm to calculate the distance between nodes can be implemented and analyzed.
At the moment, the ***"Simple Position Algorithm"*** is implemented. This algorithm simply gets the real or estimated position of every node that is within its transmission range. With those positions, it then calculates the distances to every node. These distances are given to the Position Calculator to actually calculate the current position of the node.
For the "Simple Position Algorithm" two parameters can be set:

- minMeasurements: this is the minimum number of distances to nodes that are necessary to calculate its position. In 2-dimensional space at least three nodes have to be within the transmission range, that a node is able to calculate its position.
- maxMeasurements: This parameter defines the maximal number of positions/ distances to be given to the Position Calculator to calculate the nodes position.

Position Calculator

The Position Calculator is the module that calculates a node's position, if many distances to other node positions are given. It calculates the node's position by using triangulation. Whenever a node reaches more than just three (in 2-dimensional space) or four (in 3D) other nodes, the calculator has to solve an over-determined system by using any form of minimising errors.
At this time, the ***"QR Position Calculator"*** is implemented, which doesn't have any parameters to set.

Position Estimator

The Position Estimator acts like a filter, that estimates the nodes position by utilizing the their position histories. The Position Estimator e.g. can filter out all jittering caused by measurement errors or can weight present data more than older ones.
At the moment, there are three different Position Estimators available: FIR, Mean and Median. These Position Estimators only use the calculated position history. In a later version of the simulator however, also the estimated position history could be taken into the calculation.

a) FIR

This module uses a FIR filter to calculate an estimation of the node's position. To be able to do that, the firCoefficient parameter is given to the module. With this parameter you can define, how much old position calculations should be weighted to estimate the current position.

- firCoefficient: This parameter is a comma-separated array of double values, which are multiplied with the corresponding entry of the position history of each node. With the following example of a firCoefficient, the newest entry of the position history is weighted 100%, then up to the tenth entry, every value is taken into the calculation with 10% less weight.

```
firCoefficient = [1, 0.9, 0.8, 0.7, ..., 0.2, 0.1]
```

b) Mean

The second choice of Position Estimators is the Mean module. This module calculates the statistical mean of the calculated position history. There are no parameters to be set. The following example shows the estimated position, when there are n entries in the calculated position history.

$$X = \frac{X_1 + X_2 + \ldots + X_n}{n}, \quad Y = \frac{Y_1 + Y_2 + \ldots + Y_n}{n}$$

c) Median

The third possibility is the Median module. This module also takes the calculated position history, sorts all entries and returns the entry, that is in the middle of the sorted history. If the length of the history is even, the mean value of the two entries in the middle is returned. In this module there are also no parameters to be set.

$$X = X_{middle},\ Y = Y_{middle}$$

Distance Quantizor

Once a distance from one node to another is measured, the Distance Quantizor module can be applied to the result. With this module, a distance can be quantized, that means e.g. that only a certain number of different values are expected. With this module, real behaviour of nodes can be reproduced.
At the moment, the ***"Linear Quantizor"*** is implemented and is explained in the following section.
The Linear Quantizor splits the transmission range of a node into a given number of intervals. With the given parameter ratio, the exact position, within the interval is specified.

The Linear Quantizor knows four parameters through which the real situation can be specified.

- range: This is the transmission range of the node.
- steps: This parameter defines the number of quantization steps the distance should be divided in. With this module the transmission range is divided into this number of equal parts. If steps is set to -1, no quantization is applied.
- limit: If the boolean parameter limit is set to true, the quantizor returns a maximal value that is the node's transmission range.
- ratio: This parameter defines the value within a quantization step, that is returned, if the given distance lies in this interval.

Error Simulator

The Error Simulator is used to simulate errors occurred in real measurements. The module adds a specified error to a distance between two nodes. Today, there are three different error simulators available.

a) Uniform Error

With this error simulator module, randomly distributed errors using a uniform distribution are added to a calculated distance.
This module has three parameters that can be set.

- errorRate: If proportional is set to true, this double value defines the ratio of added error and calculated distance.
- range: If proportional is set to false, with this value the range of errors can be set.
- proportional: This boolean parameter defines, whether the added error is proportional to the calculated distance or the value given with the parameter range.

b) Gaussian Error

With this error simulator module, randomly distributed errors using a gaussian distribution are added to a calculated distance.
This Gaussian error simulator knows four different parameters.

- deviation: this is the standard deviation of the gaussian distribution which is used to generate the random errors.
- range: If proportional is set to false, the range of errors can be set with this value.
- limit: This value defines a limit for the generated errors.
- proportional: This boolean parameter defines, whether the added error is proportional to the calculated distance or the value given with the parameter range.

c) Mixed Error

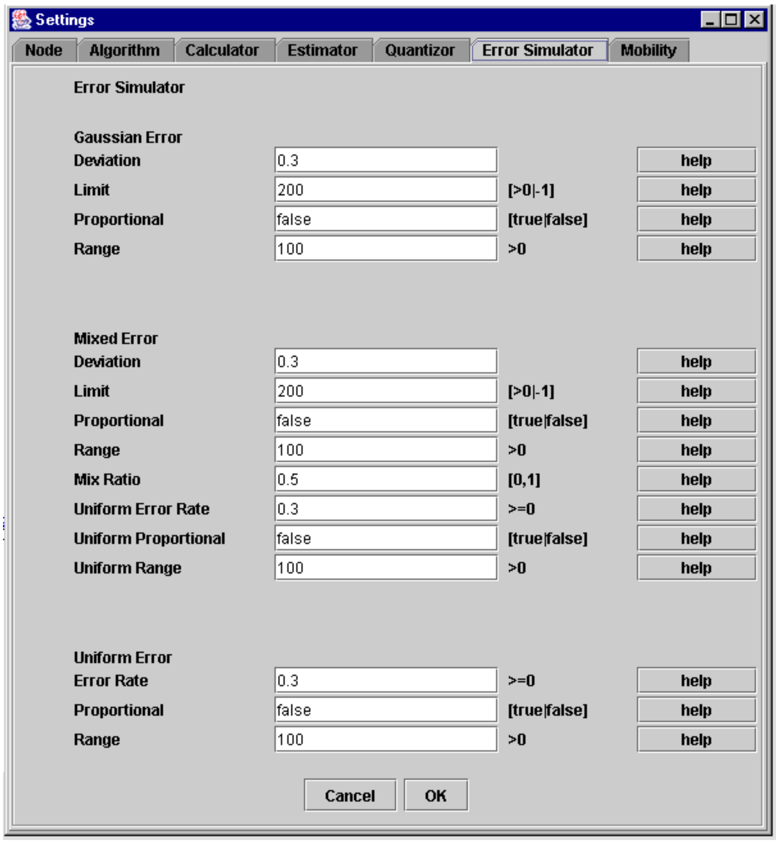
This Error Simulator is a mixture of the Uniform Error and the Gaussian Error Simulator. The mixed error is generated from those two components.

The Mixed Error module has eight parameters to set. For the meanings of the uniformErrorRate, uniformRange and uniformProportional see the part Uniform Error and for the parameter gaussianDeviation, gaussianRange, gaussianLimit and gaussianProportional check the section Gaussian Error.

- mixRatio: With this parameter, the ratio of the mixture of uniform errors and gaussian distributed errors can be set. The calculation of the mixedError is shown in the following line.

```
mixedError = (uniformError*mixRatio)+(gaussianError*(1-mixRatio))
```

The following screenshot 3-4 shows the window to set all parameters of the Error Simulator.



Screenshot 3-4: The window to set all parameters for the Error Simulator

Node Mobility

The Node Mobility module is used to generate new positions of a node to simulate various different node movements. With this Mobility Service a node is given a path of positions where it was placed.

There are three different mobility possibilities implemented at the moment.

a) No Mobility

This module is chosen, if no node mobility should be executed.

b) Random Mobility

With the Random Mobility module a node changes its position with a defined probability to a new randomly calculated position within a given range around the current position.

For this module two parameters can be set.

- maxDistance: This parameter specifies the maximum range in which the new position of the node can be calculated to.
- movePossibility: A new position is calculated with the probability of movePossibility. With a high movePossibility a node changes its position with almost every click on the go button.

c) Random Path Mobility

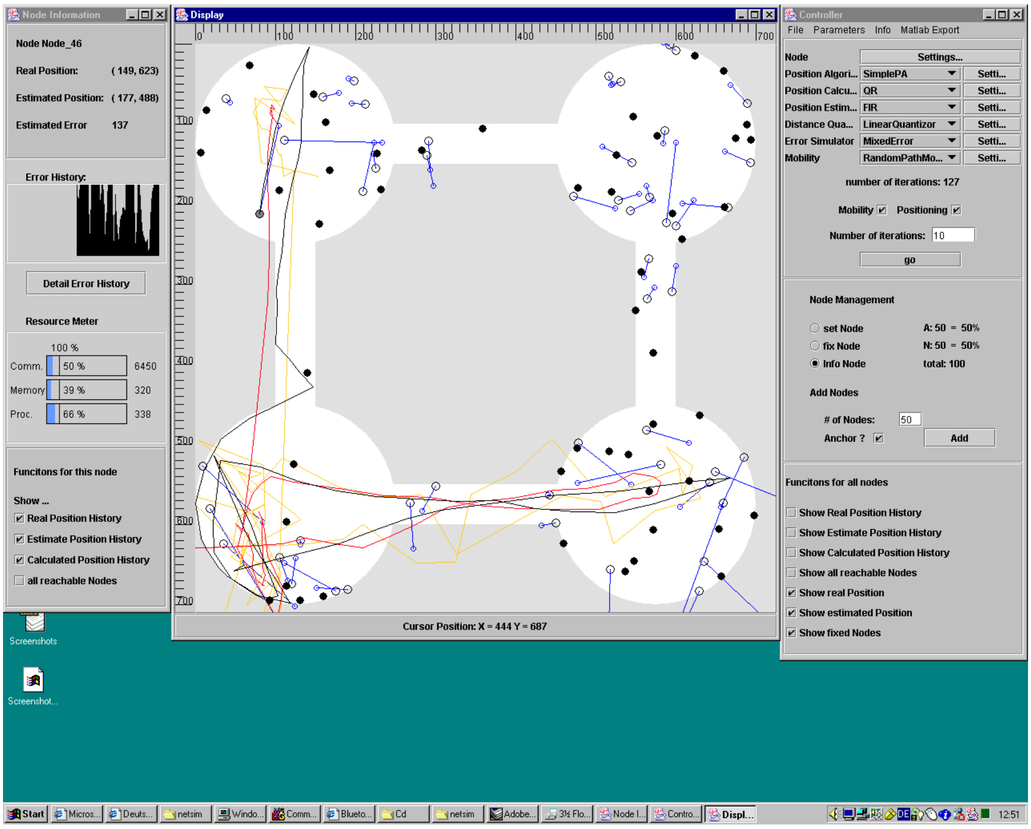
This Mobility module also generates random positions, however the generated positions lie closer on paths on the Playground. Once a node is moving in a certain direction, the next generated position stay on the chosen moving path.

- maxDistance: This value defines the maximum distance a node can move in one step.
- speedDeviation: This parameter defines the deviation of speed changement from one step to the next one.
- angleDeviation: Similar to the speedDeviation, with the angleDeviation the deviation of direction change of movement can be set.
- movePossibility: With this given movePossibility, the node changes its state from the standstill state to the moving state.

- stopPossibility: This is the possibility with which a node changes from the moving to the standstill state.

The following Screenshot 3-5 shows an example of the NetSim, where the node mobility module was applied. The four big circles in the display simulate four cities and the connections between them represent four highways. During the simulation, the nodes followed a random path and the positioning algorithm estimated the nodes position in every iteration.

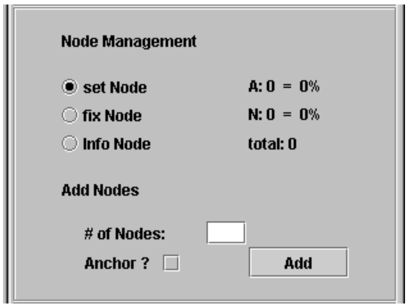
The selected node moved from the city on the lower right side to the city on the upper left side of the display, and moved therefore through three cities. Now the calculated and estimated positions can be analyzed by following the light brown and red lines. It can be seen, that the estimated (red) positions were always closer to the real position than the calculated (light brown) positions. The applied FIR filter caused a better result and therefore the history of calculated position should be taken into the estimation of the position.



Screenshot 3-5: Example of the Node Mobility with the City Example

Below all settings possibilities, you find the 'go' button to actually start the simulation specified through the parametrization. In the field 'number of iterations' the simulator can be told, how many times it should repeat the calculation. In the field above the number of iterations already performed is shown. With the check boxes 'Mobility' and 'Positioning' you can either choose, if the Mobility Service and/or the Positioning Service [5] should be executed. You have to choose at least one of the two check boxes, otherwise NetSim asks you to choose one.

3.2.3 Node Management Field



Screenshot 3-6: Node Management Field

In the second field of the controller frame you can specify all actions you want to perform with the nodes in the display. If you want to manipulate only one node at the time, you can choose between 'set Node', 'fix Node' or 'Info Node'. Whenever you click in the display in the '*set Node*' mode, you can create a node at the current position of the mouse. If the check box 'Anchor' below is checked, the set node becomes automatically an anchor, otherwise the node is a normal one, that doesn't know its position yet. By double clicking on a certain node it can also be removed again. In this mode you can also change a nodes real position by simply dragging the node to a new position.

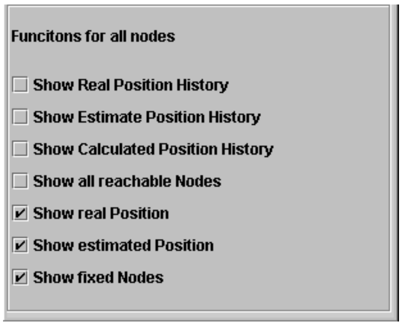
The second mode, the '*fix Node*' mode is used to anchor a node that has estimated its position, or to do the reverse, to change an anchor to become a normal node again.

And whenever you like to get more information about a certain node, change the mode to '*Info Node*' and click on a chosen node. You will see, that you get a lot of information about this chosen node in the frame Node Information. Details about that frame are given in section 3.4. Also in this mode, you can drag the mouse from one node to another in the display, and the distance between them will be indicated in the field at the bottom of the display.

These three modes are useful, if you want to work with only one node. However if you like to add many nodes randomly in the display, you better enter the wished number of nodes in the text field 'number of nodes', specify if those nodes should be anchors or not, and press the 'Add' button. In this way you can generate randomly distributed nodes at once.

The numbers on the right side tell you, how many anchors, normal nodes and how many nodes in total you have added in the display. You also get the percentages of the two different kind of nodes. Like that different simulations with always the same ratio of anchors and normal nodes can be started.

3.2.4 Selection Field



Screenshot 3-7: Selection Field

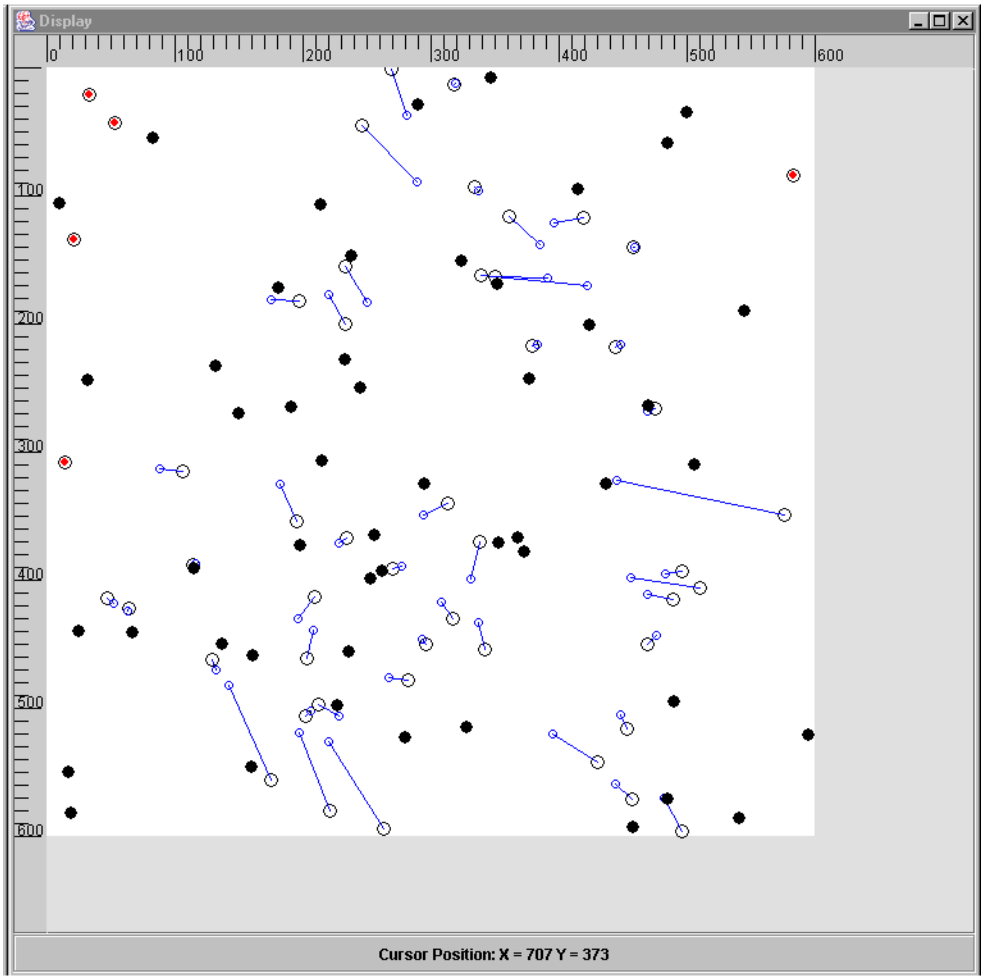
The last field of the Controller is the Selection Field in the lowest part. In this field, you can choose what kind of data of all nodes you would like to have drawn in the display. When “Show Real Position History” is checked, all real positions each node has been placed at is drawn with a black line. With “Show Estimate Position History” and “Show Calculated Position History” the corresponding red and light brown lines are drawn to indicate all estimated and calculated positions. The calculated position is generated by triangulation of the distances to other nodes. The estimated position however takes the calculated positions a node was placed at before, and estimates a new position of the node with this information. The fourth check box can be chosen, if the grid with all reachable nodes should be shown. With this grid, you can analyze, which nodes can communicate together and stay therefore within transmission range.

Because you normally want to see all normal nodes with their real and estimated position and also all anchors in the display, the last three checkboxes are already checked at the start of NetSim. You can certainly switch those checkboxes off to get a better look on certain details.

3.3 Display

The frame in the middle of the screen is called the Display. In this frame all nodes with all their actions can be shown.

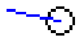
The size of the field, where nodes can be set, is indicated with a horizontal and a vertical ruler at the top and on the left side of the display. At the bottom of the field the current position of the mouse is indicated, or if the mouse points over an existing node, its name is shown.



Screenshot 3-9: The Display frame

To understand all different circles and lines on the display, here is a short summery of those meanings

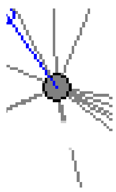
The circle that is filled with dark black indicates an anchor. That means, that this node knows its exact position.



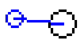
The circle that isn't filled with any color indicates the real position of a node, that doesn't know its exact position in the real world.

The small blue circle is the estimated position a node has calculated. If this small blue circle is close to the real position, the algorithm has well estimated its position.

A small red circle indicates, that this node hasn't calculated its position yet, because it doesn't get enough information to do that. Maybe its range is not far enough to reach at least three other nodes.

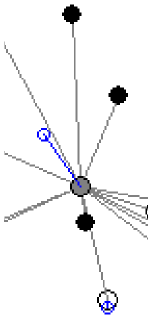


The circle that is filled with dark gray indicates the selected node. All information in the Node Information frame is about this node.

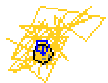


The blue line between the real position and the estimated position indicates the size of the error and tells the user, which two positions belong to the same node.

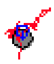
Screenshot 3-8: Explanation of all circles in the display



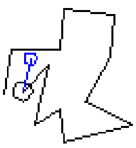
These gray lines connect all nodes, that are able to communicate together. This is only possible, when the two nodes are within their transmission ranges. This raster is shown, if 'show all reachable nodes' in the controller is checked.



The light brown line shows all positions a node has calculated during the iterations of the algorithm by triangulation of the measured distances to other nodes.



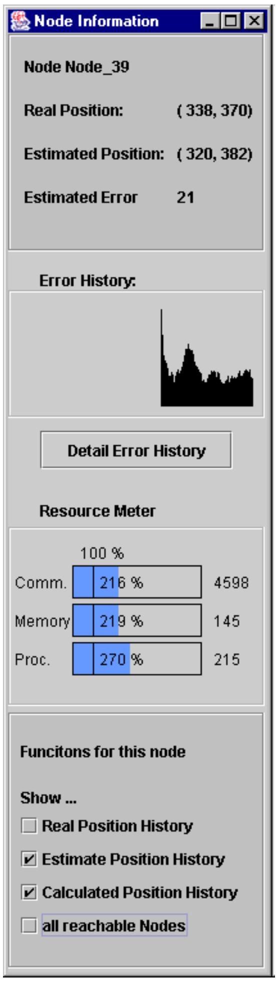
This red line indicates the positions a node has estimated by using the earlier calculated positions from the calculated position history.



If mobility of the nodes is allowed or if a node is dragged to another position, this black line indicates the different real positions a node has been placed at.

Screenshot 3-10: Explanation of all lines in the display

3.4 Node Information



Screenshot 3-11: Node Information Frame

The third frame that is shown on the left side on the screen is the Node Information. As written above this frame is filled with information of a node, that is selected in the display.

In the first part of the frame the node's name, its real and estimated position and its estimated error are shown. If an anchor is selected, '(Anchor)' is added to its node name.

The second part shows a small window, where the error history of the selected node is shown. This window only gives an impression on the behaviour of the error with the increasing number of iterations. If the detailed view of the error history is asked, click on the 'Detailed Error History' button to open a bigger window. This window is adjustable in its size, so that the whole graph can be seen. It also provides scroll bars, to get to the information that is asked.

The Resource Meter is shown in the third part of the Node Information frame. The implemented Resource meter examines the amount of communication, memory and processing needed by a node. The indicated value is calculated relatively to the rest of the nodes. That means, if the needed resource of the selected node is higher than 100%, the selected node performed more Communication, Memory Usage or Processing than the average of all other nodes. Whenever a node communicates with another node through an inquiry, connection establishment or through measuring the distance between two nodes, the Communication value on the right side of the Resource Meter is incremented by one. The Memory value represents the length of all histories of the selected node. At the moment the QR algorithm determines the value concerning the Processor resource. This value represents the processor usage from a node by inverting a matrix or by performing multiplications and additions.

The last section of this frame is similar to the last one of the controller explained in 3.2.4. With these different check boxes, you can indicate, what kind of information about the selected node you would like to have shown in the display. Check 'Real position history' to get a black line to all positions, the selected node had been placed before. With 'Estimated Position History' you get the estimated history of the node and with 'Calculated Position History' you get its calculated history. With the fourth check box, 'all reachable Nodes', a line from the selected node to all nodes, that are within its transmission range, is drawn.

3.5 Future development

At the end of this chapter, a short summary of some future development of the GUI is explained.

As it can be seen in Screenshot 3-5, the field for the playground can have different shapes. In the field there could also be some obstacles, where the waves are reflected or absorbed. It would be nice to be able to paint the field and the obstacles directly in the GUI. Therefore a square and a circle tool should be provided, with which all the different shapes can be drawn in the display frame.

A second idea for the future development is the group parametrization of nodes. At the moment, the parametrization has an effect on all nodes in the playground. However in real life, there are not always the same nodes with all the same behaviour in a certain area. Therefore it would be nice to have groups of nodes, which can be parametrized in a way, that doesn't change the settings of the rest of the nodes.

With the use of the Resource Meter, all nodes could get painted in a manner of how much they are using the Processor, Memory or Communication resource compared to the average of all other nodes.

To finish up the ideas of future development, there are two thoughts, that are nice to have in a GUI. The first one is a zoom, that could be very helpful for the analyze of detailed node movements. And the second idea is the extension of the print function. Maybe sometimes one would also like to print the Detailed History frame or the Node Information. At the moment, only the display can be printed.

Figures and Pictures

NetSim Screenshots

Glossary

API	Application Programming Interface
AWT	Abstract Window Toolkit
FIR	Finite Impulse Response
GPS	Global Positioning Service
GUI	Graphical User Interface
HCI	Host Controller Interface
ISM	Industrial Scientific Medical (band)
J2SDK	Java 2 Software Development Kit
MANET	Mobile Ad-Hoc Network
NetSim	Network Simulator
PCM	Pulse Code Modulation
PDA	Personal digital assistant
RSSI	Receiver Signal Strength Indicator
TERRAIN	Triangulation via Extended Range and Redundant Association of Intermediate Nodes
UART	Universal Asynchronous Receiver and Transmitter
USB	Universal Serial Bus
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

Bibliography

[1] Specification of the Bluetooth System - Core. Version 1.1 (February 2001)

[2] *The official bluetooth specification*, Bluetooth Special Interest Group. http://www.bluetooth.com/dev/specification.asp

[3] *Bluetooth Unleashed*,
Semester Thesis of L. Wernli and R. Semadeni, July 2001

[4] *Geolocation in a PicoRadio Environment*,
Diploma Thesis of J. Beutel, December 1999

[5] *Analysis of Quantization-Effects in Distributed Positioning-Algorithms for Mobile Ad-Hoc Networks,*
Semester Thesis of E. Wandeler, July 2002

[6] Ericsson Microelectronics. ROK 101 007 and ROK 101 008 Bluetooth Module Datasheet

[7] SourceForge.net. BlueZ - official Bluetooth Stack for Linux: http://bluez.sourceforge.net

[8] SourceForge.net. BlueZ - How To for the official Bluetooth Stack for Linux: http://bluez.sourceforge.net/howto

Appendix A

NetSim CD

The provided CD includes the following files and applications:

- Semester Thesis of E. Wandeler
- Semester Thesis of S. Tschumi
- Presentation
- NetSim, the new developed Simulation Framework
    - light distribution to run the simulator
    - complete distribution including source code and api documentation

- J2SDK1.4.0 for Windows and Linux
- ANT

Appendix B

Task definition of this Thesis

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Technische Informatik und Kommunikationsnetze (TIK)

Gruppe Technische Informatik

Sommersemester 2002

SEMESTERARBEIT

für
Stefan Tschumi

Betreuer: Jan Beutel, ETZ G63

Ausgabe: 2. April 2002
Abgabe: 5. Juli 2002

Positionierung in Bluetooth Netzwerken II

Einleitung

Um die Positionen von mobilen Geräten festzustellen kann man z.B. das satellitengestützte GPS [15] oder die Zellinformation von Mobilfunk wie etwa GSM verwenden [5, 6, 19]. Innerhalb von Gebäuden lässt sich GPS heute aber nur mit sehr grossem Aufwand betreiben und die geforderte Auflösung kann nicht erreicht werden. Um Positionen im Innen- und Aussenbereich im Zentimeterbereich aufzulösen kann man überlagerte Triangulationen mit möglichst hoher Vernetzung benutzen [2, 23] (siehe Abbildung).

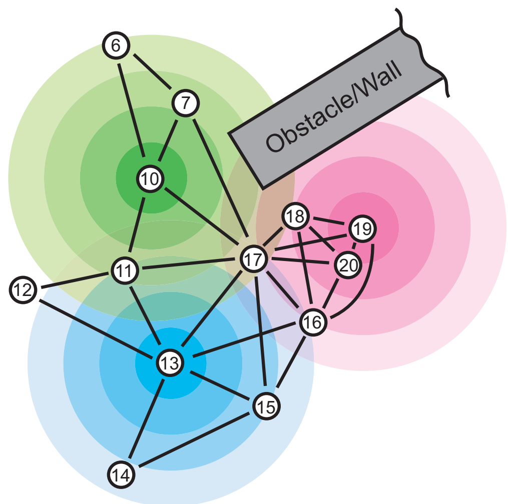


Figure 1: Ein ad-hoc Netzwerk erstreckt sich über weite Bereiche mit Konzentrationen und Ausdünnungen der Netzwerkknoten. Teile eines Netzwerkes können bei Mobilität auch abgeschirmt oder abgetrennt werden.

Mit Bluetooth [9] lassen sich die unterschiedlichsten Geräte drahtlos verbinden. Dabei können aber immer nur bis zu acht aktive Geräte in einem sogenannten Piconetz miteinander kommunizieren. In einem Raum können mehrere Piconetze in einem Scatternetz koexistieren. Über einen Backbone (z.B. Ethernet) können Bluetooth Geräte in unterschiedlichen Räumen dann miteinander kommunizieren.

Messungen von Topologie, Konfiguration und Entfernung in einem Bluetooth Multihop Netzwerk Prototypen, der am Institut entwickelt wurde [24, 4], sollen in dieser Arbeit durchgeführt werden und in einem Simulator [25] ausgewertet werden. Dazu müssen die vorhandenen Softwaretreiber angepasst werden, sowie in den Simulator integriert werden. Zum Vergleich können frühere Messungen mit Wavelan herangezogen werden [2]. Dadurch wird es möglich sein, verschiedene Algorithmen zur Positionsbestimmung in Simulation zu erproben und dann mit Versuchen zu belegen.

Aufgabenstellung

1. Erstellen Sie einen Projektplan und legen Sie Meilensteine fest [27]. Erarbeiten Sie in Absprache mit dem Betreuer ein Pflichtenheft.
2. Führen Sie eine Literaturrecherche zu Themen wie Mobile und Ubiquitous Computing, Wireless Networking, Datenlinkprotokolle, und Multihop Netzwerke durch. Ausgangspunkte bilden z.B. die Arbeiten von Rabaey, Estrin, Imielinski und Meng [20, 11, 10, 21, 22, 12, 13]. Suchen Sie auch nach neueren Publikationen.
3. Machen Sie sich mit den am Institut und bei Prof. Mattern (Smart It's) bereits durchgeführten Arbeiten [18, 24, 1] vertraut. Es sollten möglichst viele Synergien aus schon durchgeführten Arbeiten genutzt werden.
4. Überlegen Sie sich wie eine Abstandsschätzung aufgebaut werden kann. Welche Indikatoren und welche Algorithmen sind geeignet? Wie lassen sich die gewonnenen Daten darstellen und verteilen um eine Positionierung zu ermöglichen? Wie kann ein soles Positionierungssystem auf das Netzwerk verteilt werden? Wie können diese in einem Simulator verarbeitet werden? Erweitern und verbessern Sie den bestehenden Simulator [25].
5. Machen Sie Versuche wie gut die Indikatoren (RSSI, Verbindungsstabilität, Datenrate, Bitfehlerrate, etc..) von Bluetooth sich auf eine Abstandsschätzung anwenden lassen.
6. Arbeiten Sie sich in die Softwareentwicklungsumgebung des Bluetooth Stacks unter Linux ein [14, 3, 8]. Es gibt mehrere Open Source Projekte, führend und im aktuellen Linux Kernel 2.4.x integriert is BlueZ. Dieser Protokollstack soll in dieser Arbeit zur Verwendung kommen. Überprüfen Sie welche Bestandteile des Bluetooth Protokolls sich für Positionierungsalgorithmen eignet und in wie weit diese auf heutiger Hardware implementiert sind. Realisieren Sie einen Network und Topology Discovery Service aufbauend auf den bestehenden Multihop Applikationen (siehe Abbildung) sowie eine Anbindung dieser an den Simulator. Verwenden Sie dazu die Spezifikation des in den Arbeiten [4, 24] erarbeiteten Protokolls XHOP.
7. Alternative können Sie auch noch eine Anbindung an den Wavelan Treiber oder ein GPS System realisieren.
8. Implementieren Sie ein GUI unter Linux das die aus dem Netzwerk gewonnenen Daten analysiert und gesammelt zur Verfügung stellt. Eine Basis dazu stellt die Arbeit von Karrer/Moser [18] sowie Strahl [25] dar (siehe Abbildung 5) die sie von Windows portieren können.
9. Erarbeiten Sie ein Testkonzept um verschiedene Netzwerkvarianten und Verhalten zu analysieren. Verifizieren Sie Ihr Konzept und die verwendeten Algorithmen in Simulation und Versuch.
10. Dokumentieren Sie Ihre Arbeit sorgfältig mit einem Vortrag, einer kleinen Demonstration, sowie mit einem Schlussbericht.

Durchführung der Semesterarbeit

Allgemeines

- Der Verlauf des Projektes "Semesterarbeit: Positionierung in Bluetooth Netzwerken II" soll laufend anhand des Projektplanes und der Meilensteine evaluiert werden. Unvorhergesehene Probleme beim eingeschlagenen Lösungsweg können Änderungen am Projektplan erforderlich machen. Diese sollen dokumentiert werden.
- Sie verfügen über PC's mit Linux/Windows für Softwareentwicklung und Test. Falls damit Probleme auftauchen wenden Sie sich an Ihren Betreuer.

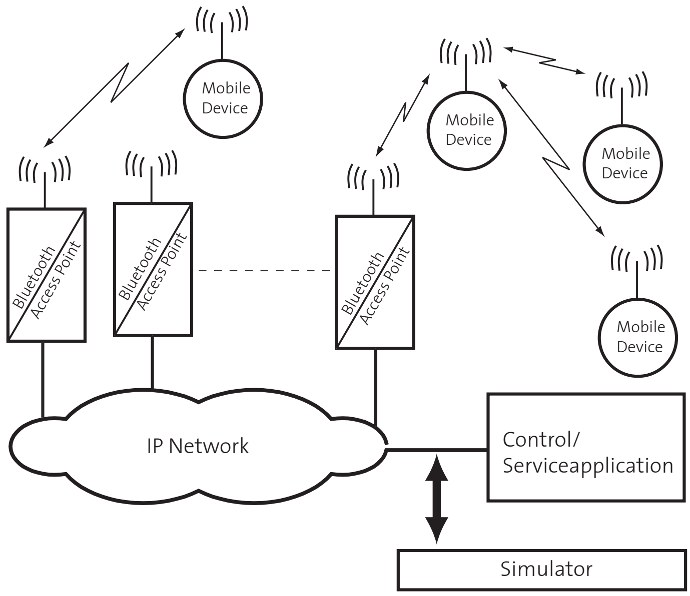


Figure 2: Verteilte Bluetooth Access Points und Piconetze.

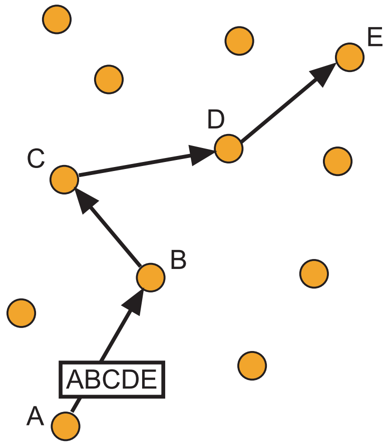


Figure 3: Multihop Routing: Beim Dynamic Source Routing wird die gesammte Routeninformation am Anfang ermittelt und mit den Daten zusammen verschickt.

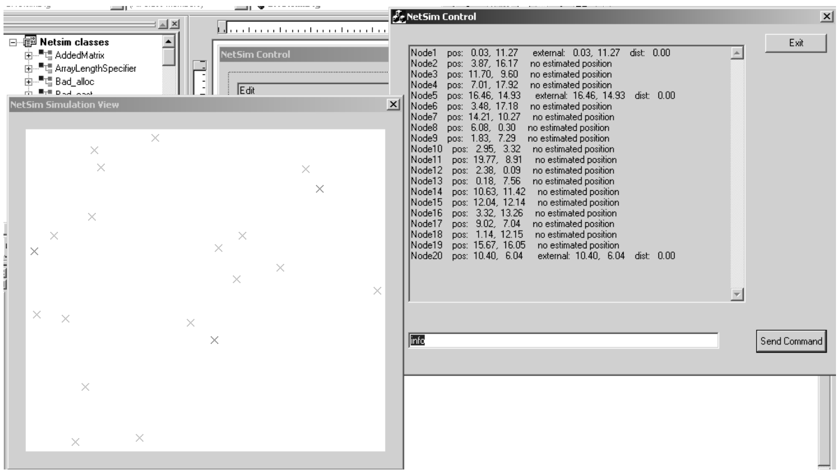


Figure 4: Netsim Positioning Simulator GUI auf Windows.

- Stellen Sie Ihr Projekt zu Beginn der Semesterarbeit in einem Kurzvortrag vor und präsentieren Sie die erarbeiteten Resultate am Schluss im Rahmen des Institutskolloquiums Ende Semester.
- Besprechen Sie Ihr Vorgehen regelmässig mit Ihrem Betreuer.

Abgabe

- Geben Sie vier unterschriebene Exemplare des Berichtes spätestens am *5. Juli 2002* dem betreuenden Assistenten oder seinen Stellvertreter ab. Diese Aufgabenstellung soll vorne im Bericht eingefügt werden.
- Räumen Sie Ihre Rechnerkonten soweit auf, dass nur noch die relevanten Source- und Objectfiles, Konfigurationsfiles, benötigten Directorystrukturen usw. bestehen bleiben. Der Programmcode sowie die Filestruktur soll ausreichen dokumentiert sein. Eine spätere Anschlussarbeit soll auf dem hinterlassenen Stand aufbauen können.

References

[1] The Smart Its Project. http://www.smart-its.org.

[2] Jan Beutel. Geolocation in a PicoRadio Environment. Master's thesis, ETH Zürich, Electronics Lab, 1999.

[3] Jan Beutel and Maksim Krasnyanskiy. *Linux BlueZ Howto*, 2001.

[4] Egon Burgener and Peter Fercher. Grenzenlose Piconetze mit Bluetooth. Master's thesis, ETH Zürich, TIK, February 2002.

[5] James J. Caffery and Gordon L. Stüber. Overview of Radiolocation in CDMA Cellular Systems. *IEEE Communications Magazine*, 36(4):38–45, April 1998.

[6] C. Drane, M. Macnaughtan, and C. Scott. Positioning GSM telephones. *IEEE Communications Magazine*, 36(4):46–54, 1998.

[7] Damon Fenacci, Marcel Grob, Peter Zberg, and Simon Künzli. Konzept Heim Netzwerk. Master's thesis, ETH Zürich, TIK, 2000.

[8] Bluetooth Special Interest Group. *Specification of the Bluetooth System - Core, v.1.1*, February 2001.

[9] Jaap C. Haartsen. The Bluetooth Radio System. *IEEE Personal Communications*, February 2000.

[10] Thomasz Imielinski and Henry Korth. *Mobile Computing*. Kluwer Academic Publishers, 1996.

[11] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed Diffusion: A scalable and robust communication paradigm for sensor networks. Technical Report 00-732, University of Southern California, March 2000.

[12] A. Iwata, Ching-Chuan Chiang, Guangyu Pei, M. Gerla, and Tsu-Wei Chen. Scalable routing strategies for ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, 17(8):1369–1379, August 1999.

[13] Rahul Jain, Anuj Puri, and Raja Sengupta. Geographical routing using partial information for wireless ad hoc networks. Technical Report M99/69, UCB/ERL Memo, December 1999.

[14] Maksim Krasnyanskiy. BlueZ, Bluetooth protocol stack for Linux. http://bluez.sourceforge.net.

[15] T. Logsdon. *The Navstar Global Positioning System*. Van Nostrand Reinhold, 1992.

[16] H. Mathys. Bluetooth Stack for the Palmpilot. Master's thesis, ETH Zürich, TIK, 2001.

[17] H. Mathys, A. Pellmont, and A. Petralia. Home Networking - Implementation (Übertitel: DMA - Drop a Message Anywhere). Master's thesis, ETH Zürich, TIK, June 2000.

[18] Thomas Moser and Lukas Karrer. The EventCollector Concept. Master's thesis, ETH Zürich, Distributed Systems Group, 2001.

[19] Kaveh Pahlavan, Prashant Krishnamurthy, and Jaques Beneat. Wideband radio propagation modeling for indoor geolocation applications. *IEEE Communications Magazine*, 35(4):60–65, April 1998.

[20] Danny Patel. Energy in Ad-Hoc Networking for the PicoRadio. Master's thesis, University of California at Berkeley, 2000.

[21] Jan M. Rabaey, Josie Ammer, Julio L. da Silva Jr., Danny Patel, and Shad Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, 33(7):42–48, July 2000.

[22] V. Rodoplu and T.H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, Aug. 1999.

[23] C. Savarese, J.M. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2037–2040, 2001.

[24] Riccardo Semadeni and Lars Wernli. Bluetooth Unleashed, Wireless Netzwerke ohne Grenzen. Master's thesis, ETH Zürich, TIK, July 2001.

[25] Reto Strahl. Build Your Own World. Master's thesis, ETH Zürich, TIK, July 2001.

[26] Sascha Wulff and Marco Wirz. Bluetooth Networking - Protocol and Application Development. Master's thesis, ETH Zürich, TIK, February 2001.

[27] Eckart Zitzler. Studien- und Diplomarbeiten, Merkblatt für Studenten und Betreuer. ETH Zürich, TIK, March 1998.

Date	Section	Changes
Jan. 28, 2002		Initial Version
Apr. 3, 2002		Final

Table 1: Revision History