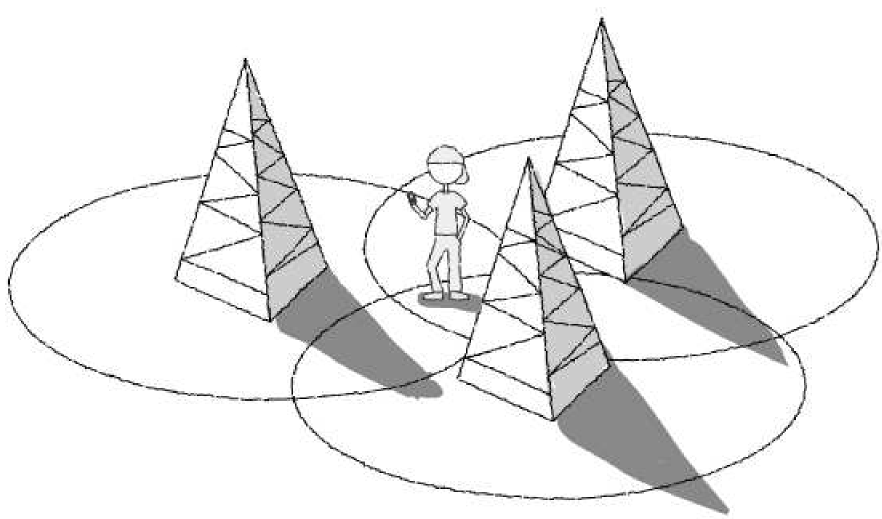
# Localization Information Retrieval Exploiting Cooperation Among Mobile Devices



Aalborg University
Institute of Electronic Systems
Department of Communication Technology

Project Group 1116, 2008

# Aalborg University

## Institute of Electronic Systems

Fredrik Bajers Vej 7 ▪ DK-9220 Aalborg Øst    Phone +45 96 35 87 00

**Title:** Localization Information Retrieval Exploiting Cooperation Among Mobile Devices

**Theme:** Cooperative Localization

**Project period:** October 15th 2008 to July 29th 2008

**Abstract**: Location estimation within cellular networks has been of increasing relevance in the recent years. Several positioning techniques have been developed, but most of them rely on line-of-sight path between the base station antenna and the mobile device. Thus, the performance degrades in densely built urban areas.
The *Global Positioning System* (GPS) is an attractive solution. However, it has several drawbacks. It needs line-of-sight path with the satellites. The time-to-first-fix for a conventional GPS receiver can take several minutes. Additionally, adding GPS functionality to handset can be costly, and drain battery power at an unacceptable rate.
The main idea of this project is to exploit cooperation among mobile devices to retrieve their geographical position without using GPS. The approach is fingerprint-based and it exploits Cell ID information. In this context Google has recently launched "MyLocation". We enhanced this service by using cooperation.
During this research work, a prototypical system has been developed. Its limitations have been discussed and possible enhancements have been suggested. The main result is that the proposed schema with three cooperative mobile devices can reduce the estimated area of more than five times compared to a non cooperative scenario. Moreover, for three cooperative mobile devices, the distance between the GPS position and the center of the estimated area is about 70 m and, as regard the range of the estimated area, this is about 145 m.

**Project group:**
1116

**Group members:**
Chiara Sammarco

**Supervisor:**
Frank H.P. Fitzek

**Co-Supervisor:**
Gian Paolo Perrucci

**Publications:** 1
**Pages:** 80
**Finished:** 29th July 2008

# Contents

# Introduction

In 1999, the US *Federal Communications Commission* (FCC), as results of the 1996 Report that established the guidelines for the basic and enhanced 911 (E911) capabilities, imposed to the wireless network providers to provide location capabilities with precise accuracy requirements during an emergency call [25].

This was the starting point for the development of radio localization techniques within cellular network. Later, the *Commission of the European Communities* started a similar initiative [6]. However, no precise accuracy requirements have been established. In Europe much emphasis has been always given to the commercial aspect of this business. Today, positioning systems and solutions are under strong development and the variety of value-added services is on the rise.

The idea behind the mobile positioning methods is to figure out the geographical location of a mobile device by using radio waves from one or more reference points. The reference points can be satellites orbiting in the space, base stations of cellular network or access points of a wireless local area network. Several positioning schemes have been presented in literature. The most used parameters are Angle of Arrival (AoA), Time of Arrival (ToA), Time Differences of Arrival (TDoA) and Received Signal Strength (RSS).

The *Global Positioning System* (GPS) is often inoperable in areas where satellite signals are blocked, such as indoors, built-up urban environments and covered areas. Thus, research is looking for alternative low-cost solutions in those areas. The performance parameters (accuracy, reliability, time-to-fix), costs and other practical issues (coverage, scalability and easy deployment) determine which location method is the best one.

In this work a *fingerprint-based* positioning system based on Cell ID is proposed. Such a method works better in the urban environment. For its implementation, no special equipments are requested. The existing mobile devices can retrieve from the network all the needed fingerprint parameters.

The main idea is to use cooperation among mobile devices to improve the localization process. In the investigated scenario, the mobile devices share their Cell ID information by using Bluetooth and send the collected data to a server that gives back the estimated position. If more Cell ID information is used, the area of localization gets reduced because it is given by the intersection of the cells of the cooperative devices.

In this context, Google has recently launched a new application called My Location [14] that exploits the cell ID to show the customer more or less where she or he is located on Google maps. The innovation of the proposed solution is in the use of cooperation [8] in this kind of network-based approach.

The main objective of this work is to prove the relevance of the proposed localization approach and to determine the order of magnitude of the achievable improvement in the location estimate.

During this research work, a prototypical system has been developed. Its limitations have been discussed and possible enhancements have been suggested.

The main result is that the proposed schema with three cooperative mobile devices can reduce the estimated area of more than five times compared to a non cooperative scenario. Moreover, for three cooperative mobile devices, the distance between the GPS position and the center of the estimated area is about 70 m and, as regard the range of the estimated area, this is about 145 m.

This report is structured as follows. In Chapter 1 the state of the art of the positioning solution in cellular network is outlined. In Chapter 2 the set up of the present project is depicted. In Chapter 3 the analysis of the accuracy results is made and the drawbacks of the present system are highlighted together with their possible solutions. In the appendixes some issues related to the development are described in more detail.

# Chapter 1

# State of the art of the positioning solution in cellular network

## 1.1 The starting point

Research in localization within cellular networks has been driven by the obligation set by the FCC (*Federal Communications Committee*) in the United States to provide Enhanced 911 (E911) wireless services. According to the obligation, cellular systems had to be able to locate a cellular phone in connection of an emergency.

In particular, the FCC report, issued in 1999, imposes that cellular carriers need to have network-based capabilities to estimate the location of the user with the accuracy of 100 m for 67% of calls and 300 m for 95% of calls [25]. In turn, the minimum required accuracy for mobile-based positioning solutions is 50 m for 67% of calls and 150 m for 95% of calls.

In Europe, the *Commission of the European Communities* started a similar initiative by publishing a recommendation on the processing of caller's location information in electronic communication networks in order to reach a location-enhanced emergency call services. The recommendation encourages the member states of the European Union (EU) to develop suitable technical solutions and practices for the provision of Enhanced 112 (E112) [6]. However, no specific accuracy requirement was declared.

## 1.2 Positioning in cellular network

As said in Section 1.1, the US FCC requirements for emergency location services has been the starting point for the development of accurate positioning methods. Today, there is an increasing interest towards mobile location technologies because of the plenty of commercial applications that can be delivered. The main purpose of this chapter is to outline the state of the art of the positioning solution in cellular network.

There are two main different approaches to retrieve the location of a Mobile Station (MS), namely *network-oriented* and *mobile-oriented* [9]. A network oriented approach uses information obtained from the network [23], [20]. Basically, two main techniques can be distinguished: the *radio waves-based* techniques and the *Cell ID-based* techniques. The first ones exploit the measuring of certain parameters of radio signals. The most used methods are: Angle of Arrival (AoA), Time of Arrival (ToA), path loss-based range estimation and Time Difference of Arrival (TDoA). The Cell ID-based techniques consider the position of the serving cell or the center of the serving sector as the location of the MS. In a mobile-oriented approach the MS, with its own forces, determines its location. This kind of localization is mostly based on GPS.

In the context of mobile positioning in cellular network, the term *fingerprint-based* method is frequently used. This refers to a positioning technique where usually the fingerprints, used for computing location, are measured (at the BSs or at the MS) and transferred to the location server for location determination (network-oriented approach). In a mobile-oriented approach, the MS makes measurements and determines its position alone.

Several positioning systems based on fingerprinting techniques have been developed in [18], [21], [28]. The main differences between these systems can be found in the types of fingerprint information and the pattern matching algorithms. Fingerprint can be any location-dependent signal information that can be measured by a MS. Lately, fingerprint-based system that uses Cell ID and WLAN scanning came up [23], [2], [26].

In general, implementing location methods requires some modifications, either software or hardware or both, to the cellular phone and/or the network. These modifications create various amounts of costs and new signalling to the network. Also the achievable accuracy of location methods varies. The requirements set by the applications determine which location method is the best or most cost-effective.

In Section 1.3 the main radio waves-based techniques are described. In Section 1.4 the Cell ID-oriented techniques are discussed. In Section 1.5 an overview on the satellites-based positioning system is made.

## 1.3 Radio waves-based techniques

This section provides an overview of the most common location techniques that are based on propagation properties of radio signals.

### 1.3.1 Time od Arrival (ToA)

In a ToA system [3], [13], the position of a MS is found by measuring the propagation time of a signal traveling from a MS to a fixed transceiver or vice versa. Geometrically, this provides a circular locus centered at the transceiver. In order to be able to resolve location ambiguity for a two-dimensional environment, two more measurements have to be made.

A ToA system has two major disadvantages. Firstly, it requires all transceivers (either at the network-side or the MS itself) to have precisely synchronized clocks. Secondly, the transmitted signal has to be time-stamped in order for the receiver to discern the distance that the signal has traveled. In addition, multipath propagation caused by signal reflections has a strong influence on the accuracy of the location estimate.

### 1.3.2 Timing Advance (TA)

In a GSM network a BTS sends synchronization information to a mobile station to ensure correct frame synchronization with the serving BTS. Basically, mobile phones have to send their data slightly earlier depending upon how far they are from the BS so that they do not overlap the timeslot allocated to another mobile. This is known as Timing Advance (TA).

The timing advance is related directly to the distance between the MS and the BTS. Values are between 0 and 63, which corresponds to a runtime of up to 233 microseconds, which in turn corresponds to the maximum radius of a network cell of 35 km. The distance from the BTS to a MS is measured in steps of 550 m [16]. This is very useful for determining the position of a MS more precisely than just with cell or sector identification.

The positioning method can be classified as a TOA technique, and the nature of the architecture is network-based. The system can be implemented without modifications to the current generation of GSM mobile stations.

### 1.3.3 Time Difference of Arrival (TDoA)

The idea behind the time-differencing scheme is to determine the relative position of a sender by examining the difference in time (Time Difference of Arrival, TDoA) rather than the absolute arrival time. Through TA, the MS has information about the propagation delays to different BTS and, therefore, it can detect the time difference between them.

To understand the principle behind this technique, let us consider a platform that sends a pulse to two spatially separated receivers' sites. The pulse will arrive at slightly different times at the two sites. This is due to the different distances of each receiver from the platform. For given locations of the two receivers, a whole series of emitter locations would give the same measurement of TDOA.

In conclusion, given two receiver locations and a known TDOA, the locus of possible emitter locations is a one half of a two-sheeted hyperboloid. In simple terms, with two receivers at known locations, an emitter can be located onto a hyperboloid. Note that the receivers do not need to know the absolute time at which the pulse was transmitted. Only the time difference is needed.

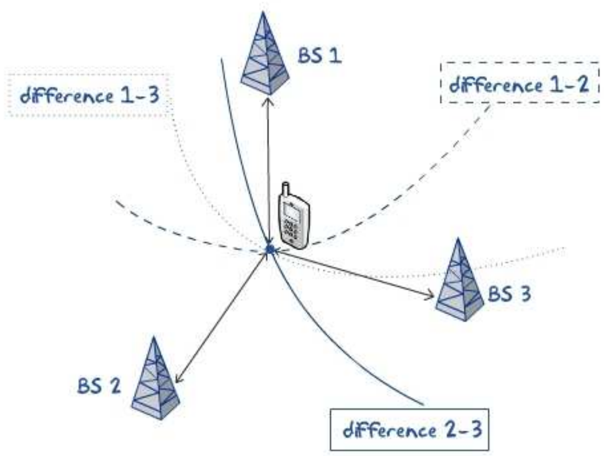



**Figure 1.1:** Time Difference of Arrival (TDoA)

Consider now a third receiver at a third location. This would provide a second TDOA measurement and hence locate the emitter on a second hyperboloid. The intersection of these two hyperboloids describes a curve on which the emitter lies. A third TDoA measurement resolves the remaining ambiguity. Therefore, the emitter's location should be fully determined in 3D. In practice, errors in the measurement of the time of arrival of pulses mean that enhanced accuracy can be obtained with more TDoA measurements (see Figure 1.1).

### 1.3.4 Observed Time Difference (OTD)

In GSM, the time difference measurements are called Observed Time Differences (OTDs). OTD measurements are made by the MS observing several BSs. The system can either work mobile-based or network-based depending on whether the GSM terminal calculates its position from two independent measurements or if the information is passed on to the network.

Enhanced OTD (E-OTD) technique has been standardized for GSM. The key of success of the E-OTD technique lies in the modified software for MS and use of sophisticated signal processing algorithms. In UMTS, OTD is standardized under the name Observed Time Difference of Arrival (OTDoA).

### 1.3.5 Angle of Arrival (AoA)

Angle of Arrival (AoA) measurement is a method for determining the direction of propagation of a radio wave that bears on an antenna array. AoA determines the direction by measuring the Time Difference of Arrival (TDOA) at individual elements of the array. From these delays the AoA can be calculated.

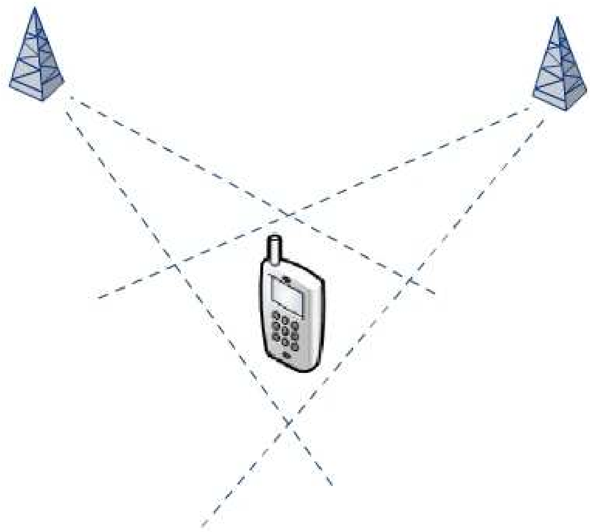


Figure 1.2: Angle of Arrival (AoA)

Signal Angle of Arrival (AOA) information, measured at the BS using an antenna array, can be used for positioning. Assuming a 2D geometry, an AoA measurement at two BSs is sufficient for unique location. This is illustrated in Figure 1.2, where the user location is determined as the point of intersection of two lines drawn from the BSs.

Also, the uncertainty in AOA measurement causes a position uncertainty that increases with MS-BS distance. Achievable accuracy depends on the number of available measurements, geometry of BSs around the MS and multipath propagation also. Since AOA technique requires line of sight between the MS and the BSs for accurate results, it is clearly not the best method for localization in dense urban areas where line of sight to two BSs is seldom present. The AOA technique could be used in rural and suburban areas.

A major barrier to implement AOA method in existing 2G networks is the need for an antenna array at each BS. It would be very expensive to build an overlay of AOA sensors to existing cellular network. However, since it is a network-based method and supports legacy handsets, it is developed by several companies as an E911 solution. In 3G systems AOA measurements may become available without separate hardware if adaptive BS antennas (arrays) are widely deployed. In addition to financial issues, AOA method may have a capacity problem. Multilateral measurement principle (measurement at several BSs) requires the co-ordination of almost simultaneous measurements at several BS sites, and it is difficult to serve a large number of users.

### 1.3.6 Distance estimation based on path loss model

An estimate of the range between a MS and a BS can be obtained by measuring the signal strength and applying one of the radio propagation models [12]. It is quite evident that the accuracy achieved from these path loss models lies on the accuracy of the radio prediction models used. The three most common propagation models are Hata-Okumura [11], COST 231 Walfisch-Ikegami [5], Intelligent Ray Tracing (IRT) [17].

For signal strength measurements, a number of error sources influence the accuracy of the position estimate. The primary source of error is multipath fading and shadowing by fixed and/or moving objects. The errors due to shadowing may be overcome by using pre-measured signal strength contours centered at each receiver. However, this approach assumes a constant physical environment, which cannot be guaranteed in every case. In addition, permanently changing environmental conditions such as rain, sleet, snowfall, humidity and temperature effect the signal attenuation and, therefore, the accuracy of the position estimate. Especially for GSM networks, the power control mechanisms for mobile stations add another level of difficulty to the employment of the SS method. In order to eliminate the effects of power control, it is necessary to disable the power control mechanism for the duration of the positioning.

## 1.4 Cell ID-based techniques

The coverage area of a transceiver station in a wireless communication network is referred to as *cell*. Its geographic extension is defined as the area within which the power level of the signal originated by the BTS does not fall below a predefined level. Environmental factors such as precipitation and a changing urban environment makes the cell's shape and extent variable and undefined even if radio signal propagation models, topological databases antenna's parameters are known. As a result, adjacent cells are required to overlap in order to provide uninterrupted coverage within a network's service area.

The relevance of a Cell ID-based method depends on the BTS density because this parameter defines the typical coverage area of a cell. While cells, ranging from 5 to 20 kilometers in radius, typically serve rural areas, a much denser network of cells covers urban regions. Furthermore, in order to increase traffic capacity, GSM networks often employ the concept of *Space Division Multiple Access* (SDMA) on a per-transceiver basis. This technique further distributes the available radio channels within the coverage area of one BTS consequently forming smaller units of 120 degrees and 180 degrees sectors. For dense cell layouts, the confidentiality of Cell ID information may exceed the results of conventional radio-location techniques. Consequently these techniques represent a very cost-efficient and straightforward GSM positioning option.

For large cells, an improved location confidence can be achieved with time information regarding recent handovers from one cell to an adjacent cell. Since a cell boundary zone crossing causes a handover event, the latter indicates that a mobile station is located somewhere along the boundary of the entered cell. Depending on several factors, including the handover algorithm, the direction of travel and the accuracy of signal strength measurements at the mobile station the boundary zones may be wider or smaller with an effect on the accuracy of this type of location information.

## 1.5 Overview of the satellite-based positioning system

*Global Positioning System* (GPS) is a satellite navigation funded by and controlled by the US Department of Defence (DoD). The system was designed for and is carried out by the US military personnel. GPS provides specially coded satellite signals that may be only processed by a *GPS receiver*, enabling the receiver to compute position, velocity and time.

The basic measurement performed by a GPS receiver is the time required for a signal to propagate from one point in space to another. Because in the general case, the speed that RF signals travel is known with relative accuracy this time measurement can easily be converted to distance from the RF source. If the range from the receiver to four satellites is calculated, the receiver can accurately determine his position anywhere on earth. The system allows the military users to make use of an enriched signal set, achieving a much better guaranteed accuracy than civilian receivers may achieve.

The system's operation relies primarily on the GPS satellites. A number of 24 LEO-SV (Low Earth Orbit - Satellite Vehicles) are positioned in such orbits as to cover almost the entire earth surface, while at any time 4 to 6 are on stand-by in orbit to replace malfunctioning. They complete one full rotation about the Earth every 12 hours.

The system is not as well adapted for urban use as the system will need to have direct visibility (Line-of-Sight conditions) with the satellites used for the position calculation. Further, the time-to-first-fix for a conventional GPS receiver from a cold start can take several minutes.

*Hybrid GPS-based techniques* can provide location information in areas that the GPS signals do not cover [22]. This can be achieved in various ways. Most of the GPS systems additionally involve/fuse with a Dead Reckoning (DR) technique to correct the position during motion. DR is the process of estimating the global position of a vehicle or MS by advancing a known position using course, speed, time and distance to be traveled. That is, in other words, figuring out where you momentarily are or where you will be at a certain time if you hold the speed, time and course you plan to travel. DR would be able to predict the estimated location during a short time of unavailability of the GPS signal. For estimations over a long time period, the system results in an error accumulation.

Further enhancements are the techniques known as *Differential GPS* and *Assisted GPS*. These techniques improve the system performance parameters such as accuracy, time-to-first-fix and coverage especially in the case where the system will be used in dense urban environments to provide location information and location-based services.

*Differential GPS* technique [15] is used to enhance the accuracy obtained by GPS. The idea behind differential positioning techniques is to correct systematic bias errors at one location, based on measured bias errors at a known position. In the case of DGPS, a reference receiver (DGPS Base Station), computes corrections for each satellite signal received. Then, the DGPS Base Station transmits the corrections to the co-observing receivers. However, there are still problems with the availability of the GPS signal.

An alternative solution is the *Assisted GPS* (A-GPS) [7]. It is a hybrid Satellite-Cellular solution that has been standardized for location in cellular networks. Such technologies make it possible to receive GPS satellite data even at signal levels below known thresholds, allowing in some cases the estimation of users' positions even when user is indoors. However, new handsets should be applied which involve GPS receivers. The main purpose of this system is to assist the GPS receiver in the position estimate by providing information about the visible satellites.

The main idea is that the information about the satellites in sight of every cell are stored in an *Assistant Server* and refreshed constantly. When the A-GPS needs its position, it uses the cellular network to connect to the *Assistant Server* that gives back the list of satellites in sight. A-GPS measures relative times of arrival of signals, sent simultaneously from at least three satellites.

The Russian analogue of GPS is termed as "GLONASS". The system is also of high accuracy, but the service provided is not trustworthy since the system may be turned on and off without warning. Furthermore, GLONASS is prone to satellites' malfunctioning while low financing seems to prohibit regular satellite replacements. Beginning in 2001, Russia committed to restoring the system, and in recent years has diversified, introducing the Indian government as a partner, and accelerated the program with a goal of restoring global coverage by 2009.

ESA is on its way however with its own European Constellation, called "GALILEO". GALILEO is a global navigation system composed of 30 dedicated navigation satellites and a ground infrastructure with the main control centres in Europe and a network of dedicated stations deployed around the world. The overall programme objective for Galileo is the deployment, by 2013, of a European navigation system providing five main services, namely the Open Service, the Safety of Life Service, the Commercial Service, the Public Regulated Service, and the Search and Rescue Service [1].

# Chapter 2

# Cooperative Localization Set-Up

## 2.1 Motivations

Location based services are breaking new ground on mobile devices. As shown in Chapter 1, there are two main different approaches to retrieve a mobile's location, namely *network-oriented* and *mobile-oriented* [9].

*Mobile-oriented* localization is mostly based on GPS. Even though the reached accuracy with this technology is quite high, GPS has several drawbacks such as limitation to outdoor, immense energy consumption, and the cost factor, which limits this technology to high class devices. From a customer point of view, the number one complaint is that it needs too long to get an initial localization and, once a GPS is hooked up, loss of signal or 'fly away' positioning in congested areas frustrates users. For all these reasons the research is looking for an alternative way to localize mobile devices among the network-oriented approaches [4].

A *network-oriented* approach uses information obtained from the network and can work on most mobile devices, it uses nearly no extra energy and it is not limited to the outdoor environment like GPS. It does not have any special hardware requirements. Thus, from a vendor's point of view, this should be the most convenient solution.

In this context, Google has recently launched a new application called My Location [14] that exploits the cell ID to show the customer more or less where she or he is located on Google maps. This approach gives a coarse information about the user's geographical location. We have improved the estimate by using cooperation.

The core idea is that each single mobile device shares its cell information with other mobile devices within its short range communication coverage. The proposed approach is not dependent on any network operator and, therefore, no charges for the customers are involved.

In more detail, the scenario investigated in this paper is as follows. We will consider an *applicant device* (AD) which tries to derive its position from the cell ID and we will show how it can exploit the information coming from other *neighboring devices* (NDs).

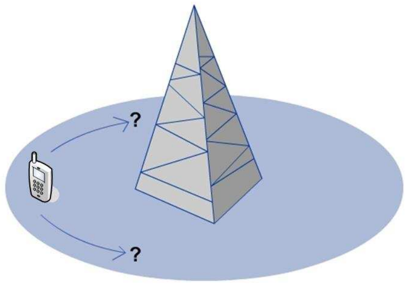



**Figure 2.1:** Area where is more likely to find a mobile device

We will show that this kind of approach can reduce the estimated area of more than five times.

In [24] the accuracy for mobile devices with multiple BS information is given (it is in the range between 70 and 200 m), if more BS information is available. Even though the mobile device has a full list of all surrounding BSs for handover purpose, from the programmer's point of view only the information of the currently selected one is available.

The rest of this chapter is organized as follows. In Section 2.2 the background idea of the project is depicted. In Section 2.3 the main tasks of the project are outlined. In Section 2.4 the preliminary measurement campaign is discussed. In Section 2.5 the architecture of the proposed platform is described.

## 2.2 The Main Idea

In this section we will describe the theory behind the whole project. The main idea of the proposed approach is to use cooperation among mobile phones to retrieve their geographical position by exploiting the knowledge of cell ID.

As said previously, even though the mobile device has a full list of all surrounding BSs for handover purpose, from the programmer's point of view only the information of the currently selected one is available. Therefore, the *applicant device* (AD) can retrieve the cell ID from one Base Station (BS) only. This information could be used to determine a zone where the mobile may be located. This means that it is assumed that the mobile is in a certain area (see Figure 2.1).

Secondly, a cell is sometimes subdivided to a certain number of sectors. A typical structure in a cellular network environment is the *trisector*, also known as *clover*, in which there are three sectors, each one served by one separate antenna. Every sector has a separate direction of tracking of 120 degrees with respect to the adjacent ones. Therefore, if we obtain from the AD its cell ID, we can define a smaller zone of localization (see Figure 2.2).

Let us suppose that the phone willing to retrieve the geographical position has two other mobile phones within its short range (10 - 15 m), which are connected to two different BSs. In a cooperative scenario, the two *neighboring devices* (NDs) can send their

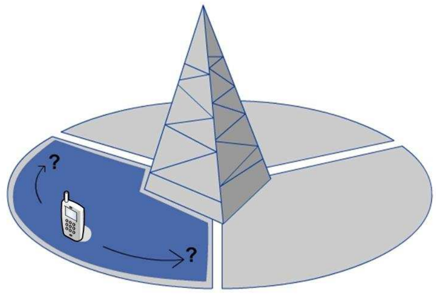



**Figure 2.2:** Area where is more likely to find the mobile device if its cell ID is known

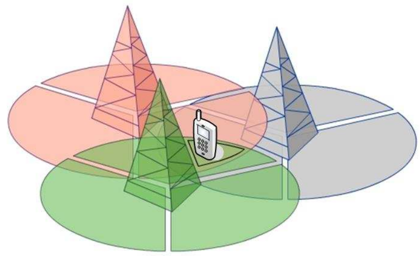


**Figure 2.3:** Area where is more likely to find the mobile device if cooperation is used

cell IDs using a short range communication technology, e.g. Bluetooth. By exploiting the information received, the area of localization is now given by the intersection of the zones individuated by each cell IDs exchanged by cooperating (see Figure 2.3). Therefore, by using cooperation [8] we can reduce the estimated area and improve by far the performance of the localization.

In this work, we have also tried to exploit the signal strength information. The main idea is basically the same. The AD and NDs exchange not only their cell IDs but also the signal strength parameter. Thus, they exchange couples (Cell ID, signal strength). Each couple individuates a zone that is no more a whole cell but a portion of it (see Figure 2.4). The intersection between smaller zones will be smaller as well. Thus, the use of signal strength could be a further source of accuracy gain.

## 2.3 Tasks

As described in Section 2.2, the approach presented in this paper is based on cooperation among mobile phones to retrieve their location by exploiting cell IDs. The main purpose of this work is to prove that cooperation can improve the location estimate. The main task of this job can be summarized as follows:

- To prove the potentiality of this approach,

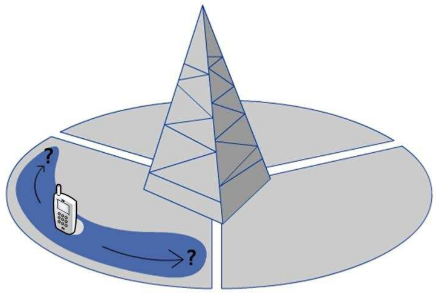



Figure 2.4: Area where is more likely to find the mobile device if its cell ID and signal strength are known

- To build a prototypic system to figure out eventual problems,
- To give an idea of the achievable accuracy.

These tasks have been traduced in the following:

**Phase 1: Gathering of the data for the elaboration (related to the BSs' placements)**

- *Symbian Developer Certificates for three Nokia N95:* Before starting, it has been necessary to prepare the mobile phones. Indeed, they could not have delivered investigation without being signed for special capabilities that let them to retrieve by Python code the GPS information, the BS ID and the signal strength. In the meantime, it has been changed hence this procedure will not be treated in this report.
- *Acquisition of the basics of Python Programming:* Python has been used a lot in the development of the whole system. For starting it has been useful attending the lessons of the Mobile Phone Programming course at Aalborg University.
- *CoopLoc - Measurements:* Development of an application for Nokia N95 for carrying out the measurement campaign (see Appendix A for the Python code).
- Measurements Campaign for CBB operator in Aalborg's downtown.
- Measurements Campaign for TDC operator in Aalborg's downtown.
- Measurements Campaign for Telia operator in Aalborg's downtown.
- *Easy-PHP:* Setup of a MySQL and Apache servers.
- Creation of a MySQL database to store the data for the elaboration.
- *CoopLoc - Processing:* Development of two Python scripts (for PC) to process the gathered data and fill in the MySQL database (see Appendix B for the Python code).

### Phase 2: Development of the tools for the analysis of the collected data

- *Acquisition of the basics of PHP and Javascript programming and of the way to make them interact:* PHP and Javascript have been used not only for this phase of the work but also for the development of the server side script and for the accuracy analysis.
- *Web View:* Setup of a web application to observe the stored data.

### Phase 3: Development of the server side script

- *CoopLoc - Server Side Algorithm:* Development of a PHP script that accepts $(BS, SS)_i$ and return the estimated position $r, (Lat_c, Long_c)$ (See Appendix C).

### Phase 4: Development of *CoopLoc*

- *Google Maps:* Understanding how to use Google Map data in mobile applications (see [19]).
- *CoopLoc:* Development of a Python application that looks for neighbors, exchanges data with them and sends the collected data to the server in order to retrieve its own location (see Appendix D).

### Phase 5: Accuracy Analysis

- *CoopLoc - AccuracyTests1:* Development of a Python application for accuracy analysis for Nokia N95.
- Measurement Campaign for testing purpose.
- Creation of a MySQL database to store the accuracy data.
- *CoopLoc - AccuracyTests2:* Development of a Python application to test the algorithm and fill in a database made for storing the accuracy data.
- *Web View:* Development of some Web pages (using PHP, Javascript and, of course, HTML) for analyzing the accuracy results.

### Phase 6: Web site of the application

- *Web site:* http://coooploc.es.aau.dk.

This is shortly what has been done. More details about code and development can be found in the appendixes. The accuracy results have been collected in Chapter 3. In the rest of the chapter more about the first four phases of the work is said. In particular, in Section 2.4 the first two phases are better treated. In Section 2.5 the functioning of the final system is described (Phases 3 and 4).

## 2.4 Measurements Campaign

When we started, the BS's placements were not available. Therefore the measurement campaign was delivered. In this section we discuss the first two phases of this project: gathering of the data for the elaboration and development of the tools for their analysis. These were two essential phases of the project. Indeed, we could not have started without having the data for the elaboration (in other words without having the BSs' placements). In addition to this, we needed to know if the main idea was applicable in the scenario where we were working. Therefore, the second phase was important because we figured out how many base stations are in the downtown of Aalborg and how they are placed.

First of all, we developed *CoopLoc - Measurements*, an application that runs on Nokia N95, using Python for S60 [10]. *CoopLoc - Measurements* waits for the availability of some satellites' data and when it receives them, it asks for the name of the path for the survey and it starts with the measurements. See Figure 2.5 for some screenshots of this application. We walked along the main streets of Aalborg while GPS position, BS ID and signal strength of every mobile phone had been storing for post processing. This is what we call *Measurement Campaign*. We did this with three N95s equipped with SIM cards belonging to different Danish operators (CBB, TDC, Telia) and run the above mentioned application. For the Python code of *CoopLoc - Measurements* see Appendix A.

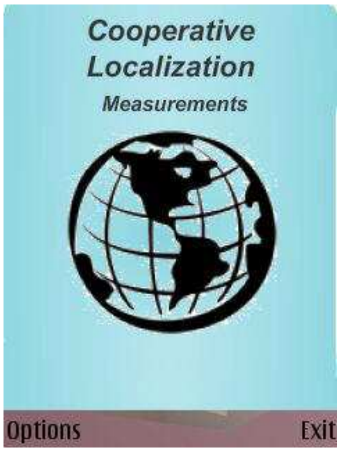
Afterwards, we developed two Python scripts for processing the stored data (for the Python code see Appendix B). The main task of these two applications was to fill in a MySQL database that had been created before. Basically, this database contains: latitude, longitude, Mobile Country Code (MCC), Mobile Network Code (MNC), Local Area Code (LAC), Cell ID (CID), signal strength, name of the path, name of the operator (even though this information is included in the MNC).

At that point, a MySQL database with all the information was available. A web application, made using PHP and Javascript, was developed for visualizing the retrieved data on a Google Map. In particular, Google Maps have been used to show the area covered by measurements (see Figure 2.6) using points of different colours according to the different operators (*CBB*: green; *TDC*: red; *Telia*: blue). To examine the base stations coverage and placement, a view of the different cells for each operator has been created (see Figure 2.7).

Thanks to the maps views we certified that the base stations have different placements and the cells intersect each other and, therefore, the main idea was quite well applicable.

## 2.5 Architecture

In this section we are going to describe the architecture of the proposed platform which is server-client based (see Figure 2.8). On the client side, the mobile runs an application which looks for neighbors and exchanges data with them. Once the BS IDs and the signal strengths are obtained, they are sent to the server through a socket connection.




(a)

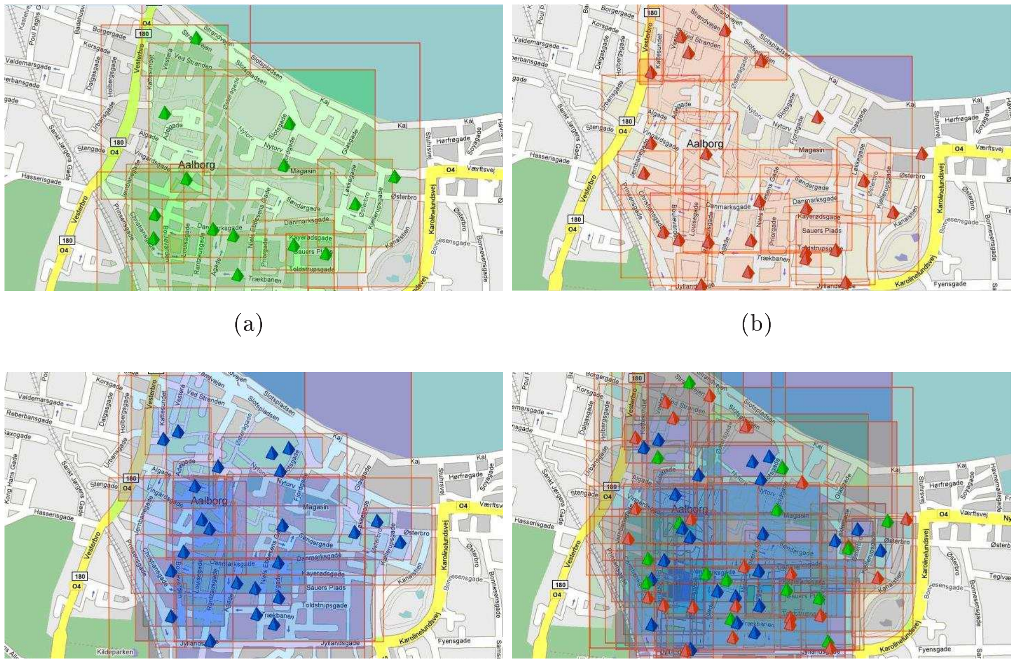

(b)


(c)


(d)

**Figure 2.5:** Screenshots of *CoopLoc* - *Measurements*


**Figure 2.6:** Measurements campaign in the center of Aalborg for CBB, Telia, TDC



(a) (b) (c) (d)

**Figure 2.7:** Cell Coverage over Aalborg

On the server side, an algorithm defines the smallest area where the mobile phone might be positioned according to the received data and sends the estimated position to the mobile device that visualizes the result on a map.

### 2.5.1 Server side algorithm

Because we want to understand if it is convenient to use the signal strength parameter in this kind of approach, two solutions have been developed named "signal strength + cell ID" solution and "only cell ID" solution. The two solutions will be best treated in Appendix C. In this section the logic behind both the solutions is outlined.

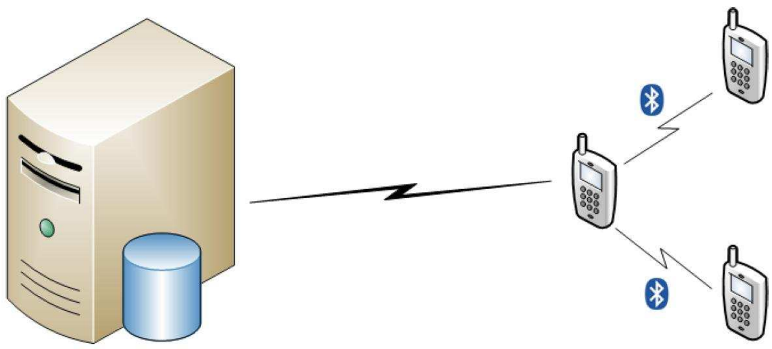


**Figure 2.8:** Architecture of the system


Figure 2.9: System as a black box

Let us imagine the server like a black box (Figure 2.9). It receives some input data and it gives back some output data. As the names suggest, the “signal strength + cell ID” solution accepts, as inputs, both the BS ID and the signal strength, while the “only cell ID” solution uses only the BS ID. The BS ID is given by a quartet of codes: the Mobile Country Code (MCC), the Mobile Network Code (MNC), the Local Area Code (LAC) and the Cell ID (CID). The MCC is related to the country (Denmark, Spain, ItalyĚ). The MNC is representative of the operator (CBB, TDC, Telia, VodafoneĚ) and, LAC and CID are linked to the cells’ subdivision in that area. The output of the whole system is, in both the cases, an estimated area that is defined from a circle. Thus, the algorithm returns a range and a couple of latitude and longitude.

The algorithm selects all the entries in the database that match the inputs. After this first selection, three sets of coordinates (one set for each operator) are obtained, see Figure 2.10.

We distinguish two kinds of set: the *Applicant Candidate Location Set* (ACLS) and the *Neighbor-Aided Location Set* (NALS). Let us consider the areas identified by these sets and let us assume they have non empty intersection. In this case, the algorithm aims to reduce the ACLS by selecting from it only the points which include in their 10 meters-range other points belonged to NALSs.

The distances between two points are calculated using the Vincenty’s formula [27] that calculates geodesic distances between a pair of points (latitude, longitude) on the earth’s surface, using an accurate ellipsoidal model of the earth. If the second step does not give results, it means that the initial sets of points do not intersect one another. Therefore the calculation is repeated increasing the 10 meters-range. This could happen for the areas not covered by measurements among which the indoor environment. After this filtering, a subset of the ACLS has been selected and the smallest circle that includes all its points is the estimated area.

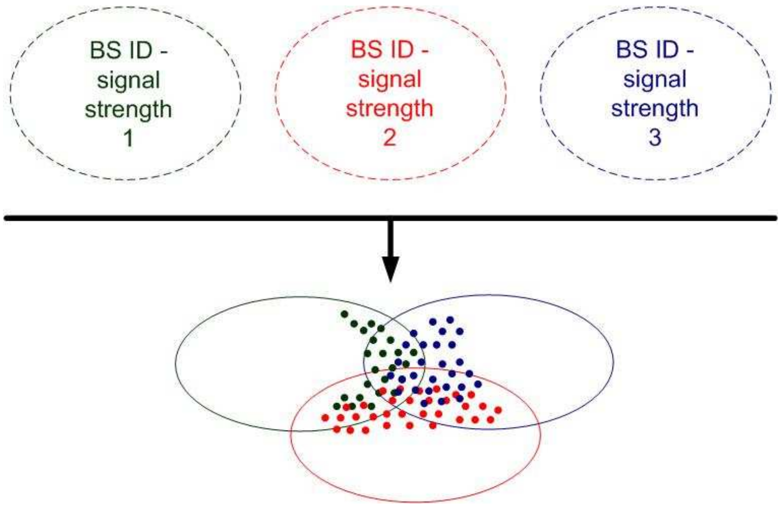



**Figure 2.10:** Server side Algorithm - First Filtering

### 2.5.2 Client side application

On the client side there is *CoopLoc*, a mobile application, whose main purpose is to exchange data with the neighbors and ask the server for the location sending the collected data. The application has been developed using Python for S60 (see Appendix D for more details).

From an end-user point of view the functionalities, accessible from the menu, are:

- *AP selection:* for selecting the favorite access point.
- *Locate Alone:* to retrieve the localization using only the data that the user can retrieve from the network.
- *Coop Location:* the application looks for neighbors, exchange data with them for obtaining a better localization.
- *Tracking:* the application looks for neighbors, exchange data with them constantly so the localization can be refreshed while, for example, the users are walking.
- *Help your friend:* to help a friend who wants to localize himself.
- *About:* to see the About window.
- *Exit:* to exit the application.

In Figure 2.11, some screenshots of the application working in Aalborg are shown.

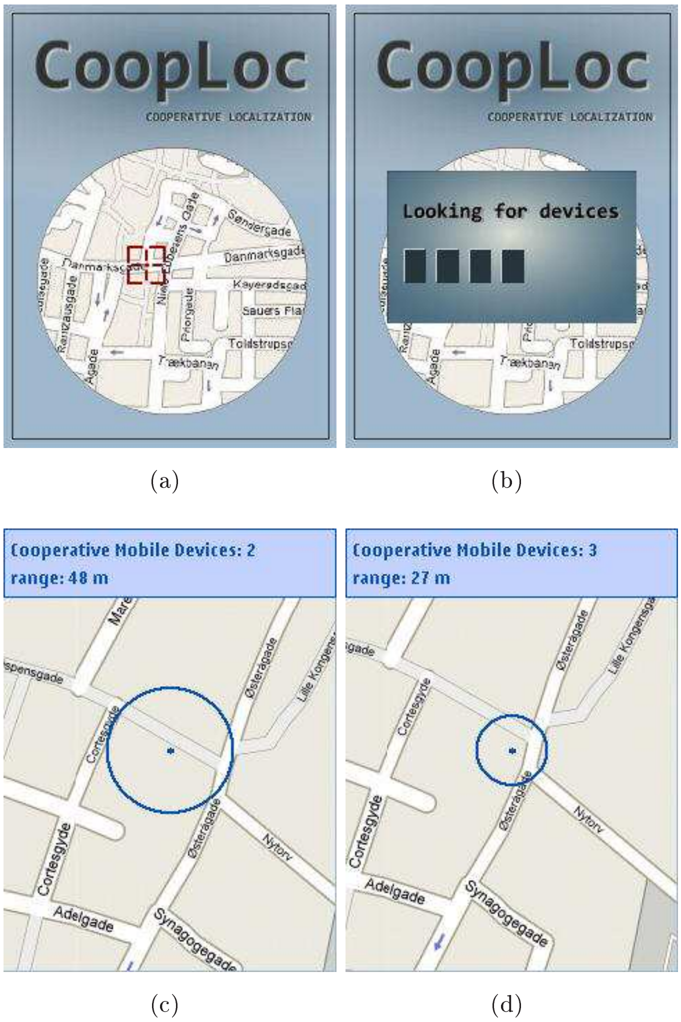



**Figure 2.11:** Screenshots of *CoopLoc* working in Aalborg

# Chapter 3

# Accuracy and Improvements of the Proposed algorithm

## 3.1 Introduction

As said in the , this work wants to demonstrate that cooperation can improve the location estimate. The purpose of this chapter is to analyze the data collected from some tests made in Aalborg and discuss the results achieved. Before starting we looked for the best way to carry out this investigation with the data that were available.

In this section the train of thought behind the whole discussion is outlined. Afterwards, the drawbacks of the present system are highlighted together with their possible solutions. Finally the reliability of the results is demonstrated and the use of the signal strength is discussed.

### 3.1.1 Train of thought

To evaluate the accuracy of the whole system, an application for Nokia N95 that stores the input data together with the estimated and the GPS positions has been developed. Thus, for each accuracy test the available information is:

- the estimated area that is given by a circle (so we have its center and its range),
- the GPS position.

As regards the real location of the mobile during the tests, the GPS position has the maximum grade of truth, so it has been considered as the real position.

Because the data used by the algorithm for the elaboration can not be considered 100% complete, the localization process can not always be a good one (this is better explained in Section 3.3.1). Thus, basically, two different cases can be distinguished:

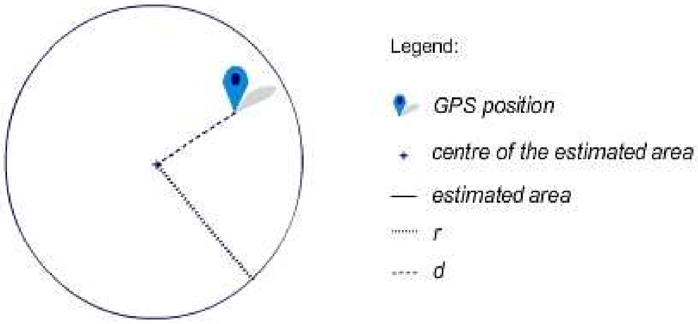



**Figure 3.1:** Good Localization - The GPS Position is in the estimated area

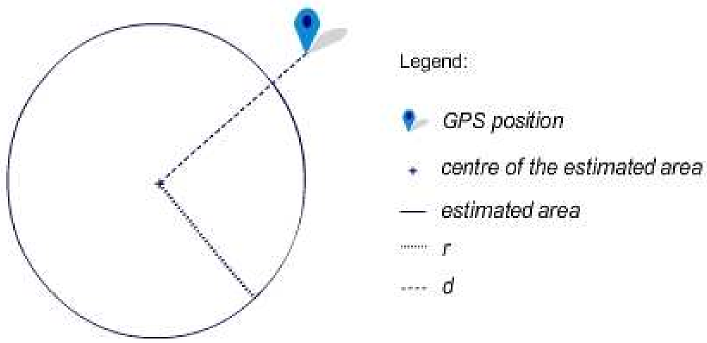



**Figure 3.2:** Bad Localization - The GPS Position is not in the estimated area

- the GPS position is included in the estimated area (see Figure 3.1). In the rest of the chapter we will refer to this situation with the terms 'good localization', 'good localization process'...,
- the GPS position is not included in the estimated area (see Figure 3.2). In the rest of the chapter we will refer to this situation with the terms 'bad localization', 'bad localization process'....

In the first case, it is interesting to observe how the range $r$ of the estimated area and the distance $d$ between the GPS position and the center of the estimated area vary as a function of the number of cooperative devices in order to understand how much the estimated area is focused on the real position. As regards the second case, it is shown how much the estimated area is far from the GPS position. Also in this case, the results will be shown as a function of the number of cooperative devices in order to understand if the cooperation reduces the error.

It is important to underline that the incompleteness of the data is not a relevant one, so it does not imply that the results can not be considered trustworthy. In Section 3.5 the reliability of the results is demonstrated.

What has been said until here will help to answer the question: *Does cooperation improve the location estimate?* But for gaining the maximum benefit from this work, in this chapter the answer to this other important question is given: *how should the algorithm be best implemented in the future?*

Because we want to understand if it is convenient to use the signal strength parameter in this kind of approach, during this chapter two sets of accuracy results are analyzed: the one related to the "signal strength + cell ID" solution and the one related to the "only cell ID" solution (see Appendix C).

In Section 3.2 the accuracy results for the "signal strength + cell ID" solution are given. In Section 3.3 a fundamental discussion of some localization processes is presented. In Section 3.4, through a discussion of the obtained results, it is outlined how the algorithm should be best implemented. In Section 3.5 the reliability of the results is demonstrated. In Section 3.6 the accuracy results for the "only cell ID" solution are given. In Section 3.7 fundamental considerations about the overall results will lead to the prediction of the performance of the system in optimal conditions. Section 3.8 concludes the chapter.

## 3.2 Accuracy analysis for the "signal strength + cell ID" solution

In this section the achieved results for the "signal strength + cell ID" solution are given. As said previously, to gather the accuracy data an application for Nokia N95 has been developed in order to store the GPS and the estimated positions in a database. The accuracy surveys were run in some of the main streets of Aalborg, Denmark.

Let $d$ be the distance between the center of the estimated area and the real position. This parameter does not distinguish good or bad estimations and it can be helpful to observe the global trend.

In Table 3.1 the maximum, minimum and medium values of this parameter are shown as a function of the number of cooperative devices. The average value for $d$ with three cooperative mobile devices is about 75 meters.

| **n. Mobile Devices** | 1 | 2 | 3 |
|---|---|---|---|
| **Maximum** | 668.9 m | 660.2 m | 243.6 m |
| **Average** | 168.6 m | 121.2 m | 74.5 m |
| **Minimum** | 14.9 m | 3.8 m | 5.3 m |

**Table 3.1:** Minimum, average and maximum value of $d$ for all the localization tests ("signal strength + cell ID" solution)

The results of the two cases introduced in Section 3.1.1 are shown in the rest of this section. Next, the conclusions are outlined.

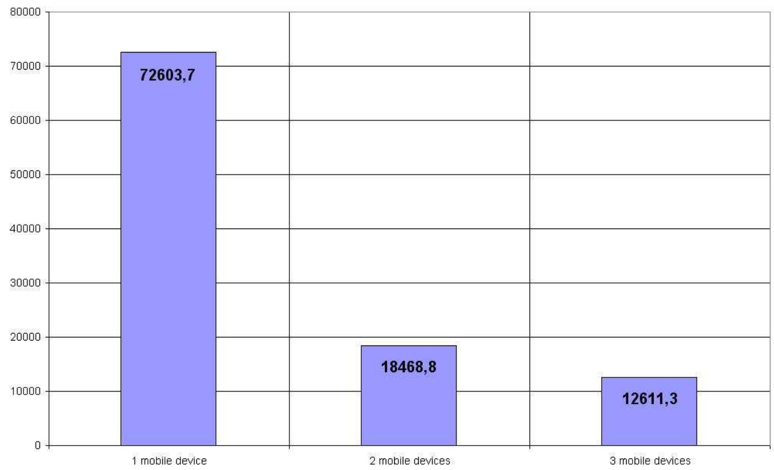



**Figure 3.3:** Estimated area as a function of the number of cooperative devices ("signal strength + cell ID" solution)

### 3.2.1 The GPS position is included in the estimated area

Let $r$ be the range of the estimated area (see Figure 3.1). As said in Section 3.1.1, if the GPS position is included in the estimated area, it is interesting to observe the variation of $r$ and $d$ in order to understand how the area of localization is focused on the real position. In Tables 3.2 and 3.3 the maximum, minimum and medium values respectively of $d$ and $r$ are shown as a function of the number of cooperative devices.

| **n. Mobile Devices** | 1 | 2 | 3 |
|---|---|---|---|
| **Maximum** | 319.5 m | 388.2 m | 147.9 m |
| **Average** | 110.0 m | 68.6 m | 51.1 m |
| **Minimum** | 14.9 m | 3.8 m | 5.3 m |

**Table 3.2:** Minimum, average and maximum value of $d$ for the good localizations ("signal strength + cell ID" solution)

| **n. Mobile Devices** | 1 | 2 | 3 |
|---|---|---|---|
| **Maximum** | 694.7 m | 556.0 m | 318.9 m |
| **Average** | 271.3 m | 135.9 m | 112.3 m |
| **Minimum** | 32.9 m | 10.0 m | 10.0 m |

**Table 3.3:** Minimum, average and maximum value of $r$ ("signal strength + cell ID" solution)

Figure 3.3 shows how the localization area decreases (in this case) according to the number of cooperative devices.

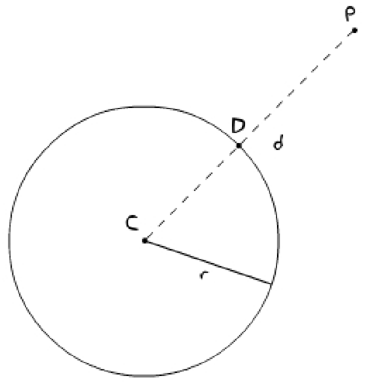



**Figure 3.4:** Distance point-circle

### 3.2.2 The GPS position is not included in the estimated area

If the GPS position is not included in the estimated area (see Figure 3.2), it is relevant to understand how much the estimated position is far from the “real” one. Therefore, the distance from a point to a circle (the estimated area) must be calculated.

From the analytical geometry, it is known that this distance, in this case (see Figure 3.4), is given by the distance between point P and the point of the circle, nearest to P, that belongs to the straight line defined by prolonging the segment PC.

In other words, the distance between the estimated area and the GPS position is given by $\overline{PD}$ that is equal to:

$$(d - r)$$

Where:

- $d$ is the distance between the center of the estimated area and the GPS position (PC in Figure 3.4),
- $r$ is the range of the estimated area.

In Table 3.4 the maximum, minimum and medium values of this parameter are shown as a function of the number of cooperative devices.

| n. Mobile Devices | 1 | 2 | 3 |
|---|---|---|---|
| Maximum | 319.5 m | 388.2 m | 147.9 m |
| Average | 110.0 m | 68.6 m | 51.1 m |
| Minimum | 14.9 m | 3.8 m | 5.3 m |

**Table 3.4:** Minimum, average and maximum value of *d-r* ("signal strength + cell ID" solution)

### 3.2.3 Conclusions

All the trends shown in this section have outlined that the larger the number of cooperative devices, the smaller the location error. With this solution, the proposed schema with three cooperative mobile devices has reduced the estimated area of more than five times compared to a non-cooperative scenario.

These results are related to the algorithm, explained in Appendix C, that uses the signal strength with the *threshold* parameter equal to zero. In the following section some important considerations are made during the discussion of some real localization processes.

## 3.3 Discussion of some localization processes

In this section two test processes are discussed (the "signal strength + cell ID" solution has been used):

- A good one to have an idea on how the system works,
- A bad one to explain why a localization process can not always be a good one.

### 3.3.1 A good case

Figure 3.5 shows the results of one localization process with our system. The mobile phone requesting the location was using a CBB SIM card. In the vicinity there were two other mobile phones with SIM cards of TDC and Telia. In Figure 3.5 (a) the cells of the three operators are shown for one given position of a mobile device. In Figure 3.5 (b) the mobile device uses only its own knowledge for a location estimate. The circle is centered on the estimated position and the second marker points the GPS position. The distance $d$ between the estimated and the real position is 32.56 m and the range $r$ of the circle is 318.17 m. In Figure 3.5 (c) the mobile device is locating itself using also the knowledge of its TDC neighbor ($d$ is 27.36 m and $r$ is 125.79 m). In Figure 3.5 (d) the data of both its neighbors are used ($d$ is 30.42 m and $r$ is 92.69 m). As a result, cooperation has improved the location estimate.


(a) Involved cells


(b) Localization made using only the requestor's data


(c) Localization with two cooperative mobile devices


(d) Localization with three cooperative mobile devices

Figure 3.5: Example of a Real Localization Process

### 3.3.2 A bad case

Figure 3.8 shows a localization process with the present system in which the GPS position is not in the estimated area. The possible causes of this failure could be found in:

- the algorithm (see Section *'Calibration Error'*)
- the data for the processing (see Section *'Incompleteness of the data'*)

In the first case, the failure is due to a *calibration error*. In the second case it is an error due to incomplete data. In the rest of the chapter we refer to this kind of errors with the term *data error* because it is related to a data matter.

#### Calibration error

The occurrence of a *calibration error* implies that, in the first set selection, the algorithm has selected a too small set of data. In other words what was named *threshold* parameter (see Appendix C) should be better calibrated.

The probability of this kind of error decreases if the number of mobiles increases, because the number of requisites that the localization has to respect increases too. A simple example will help to understand this better. If the AD sends a localization request to the server with only its input data, the algorithm first checks for the records in the database that match the input data. If it manages to find these entries, it gives the solution immediately. If the *threshold* parameter is not well calibrated what can happen is that a too small set of points is selected. Thus, the estimated area could not include the real position of the AD.

Let us suppose that a Neighboring Device (ND) starts to help the AD. The input data are given by the union of both the ND and AD data. So even if the algorithm manages to find a set of records that matches exactly the signal strength of the AD, it does not give that result immediately because it has to check if those records are "near" to those of the ND. If it does not happen, the *threshold* parameter is increased and the signal strength variation is better considered by the process.

To solve this possible problem, the *threshold* parameter for the first set selection process must be well calibrated.

#### Incompleteness of the data

In a cellular network environment an operator usually chooses to cover the same area with more than one cell to better supply the traffic load. So in the most common cases, the cells of the same operator intersect one another.

Let us consider the simple graphic of Figure 3.6 and let us suppose that the blue guy is doing the measurement campaign along the red path. He starts from point A where

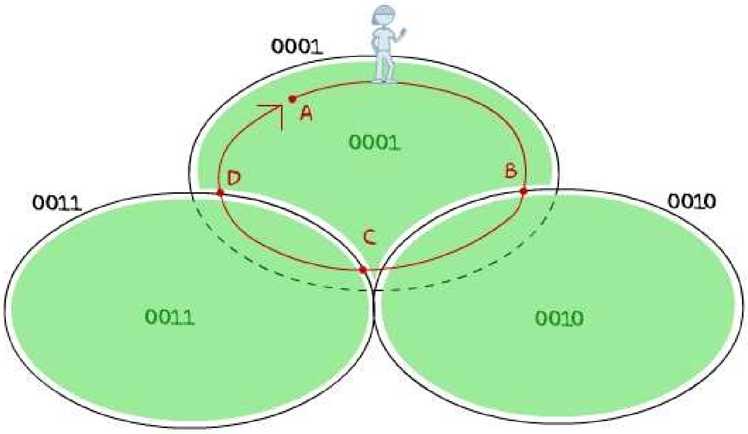



Figure 3.6: Explanation of mechanism that causes the incompleteness of the data - graphic 1

his mobile listens to cell 0001 until point B. Afterwards the mobile switches to other BSs. From point B to point C it is linked to cell 0011 and from C to D to cell 0010. Finally from D to A it switches back to cell 0001.

Therefore, if the measurement campaign is not run carefully, of course a sufficiently complete vision of the cells' placements will not be available. The black circumferences define cells 0001, 0010, 0011 of the operator A, while the green areas represent the system's vision of the cells. What does this imply?

To better understand this case, let us consider Figure 3.7. The black circumference defines the whole cell 1234 and the orange area is my understanding of this cell due to the circumstances explained before. What could happen if the AD is in the blue cross? If the AD listens to the cell with cell ID 1234, the algorithm will say that the AD is in the orange circle. This is because that one is the system's vision of the cell 1234.

### Discussion of the particular case

The example of localization in Figure 3.8 includes both the two contributes of errors explained in this subsection (*calibration errors* and *data errors*). The mobile phone requesting the location was using a Telia SIM card. In the vicinity there were two other mobile phones with SIM cards belonging to TDC and CBB.

As shown in Figure 3.8 (a) the GPS position is included in the area of intersection of two of the involved cells, thus it is a case of incompleteness of the data.

In Figure 3.8 (b) the AD uses only its own knowledge for a location estimate. The distance between the estimated area and the real position, ($d$-$r$), is 368.6 m. This is a case of *calibration error*. The algorithm has selected a very small area with exactly the signal strength of the AD.

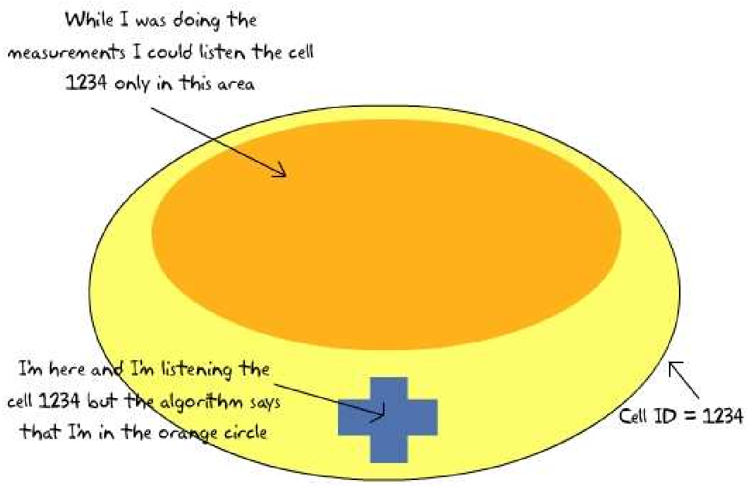
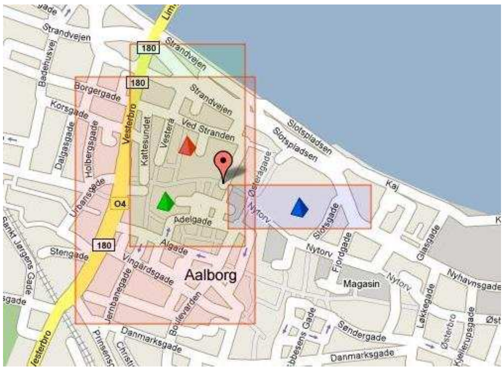



**Figure 3.7:** Explanation of mechanism that causes the incompleteness of the data - graphic 2



(a) Involved cells

(b) Localization made using only the requestor's data

(c) Localization with two cooperative mobile devices

(d) Localization with three cooperative mobile devices

**Figure 3.8:** Example of a bad localization process

In Figure 3.8 (c) the mobile device is locating itself using also the knowledge of its CBB neighbor (*d-r* is 39.0 m). The cooperation has reduced the location error. In Figure 3.8 (d) the data of both its neighbors is used (*d-r* is 30.8 m). In both the last two cases, even though cooperation has reduced the effects of the *calibration error*, the GPS position of AD is still not in the estimated area because of the incompleteness of the data.

In this section, the principles at the base of any eventual failure of the algorithm were shown. In the next section all the real and potential drawbacks of this system are highlighted and some possible solutions are proposed.

## 3.4 Drawbacks and possible solutions

The discussions carried out in this section are useful to answer the question: *How should the algorithm be best implemented in the future?* Two cases can be distinguished: we can or we can not retrieve somehow a complete set of data of the BSs' placements for the main operators of the world. The second one was the initial condition of this project.

At the beginning any BSs' placement was not available. Thus, this information was gathered from the network. Unfortunately the performance of this kind of localization system (based on cell ID fingerprint) is extremely linked to the completeness of the information. A lack in the information could cause a failure in the localization retrieval. Thus, a better understanding of the BSs' placements must be reached. To do more measurements is the only way we have for completing the elaboration data. Nevertheless, could this be done without any byproduct?

The best algorithm is supposed to work always, always fast and always well. For making the algorithm work always, we said that more measurements could be done. For making it work well it would be enough to keep cooperation because, as shown, this improves the performance of localization. Finally the algorithm efficiency as regards the response time needs to be considered.

As regards this last aspect, the overall result is that with the present algorithm the response time is variable. In particular, it depends on the number of records selected for the processing during the *set selection* phase. This is because the larger the number of selected records, the larger the number of times the *distance* function will be used (the *distance* function calculates the distance between two points according to the Vincenty's formula).

In other words sometimes the AD waits only few seconds (1 - 4 s), other times much more ($>$ 10 s). If such a behavior is acceptable for an algorithm made for research purpose, it would not be from an end-user point of view. If something "sellable" needs to be reached, a solution for this defect should be found. Therefore, it is significant to understand why this happens.

The information stored in our database has the shape shown in Figure 3.9. This implies that when an applicant device sends a couple (BS, SS), the only data the algorithm can work on are a set of records of this table. The number of records stored in

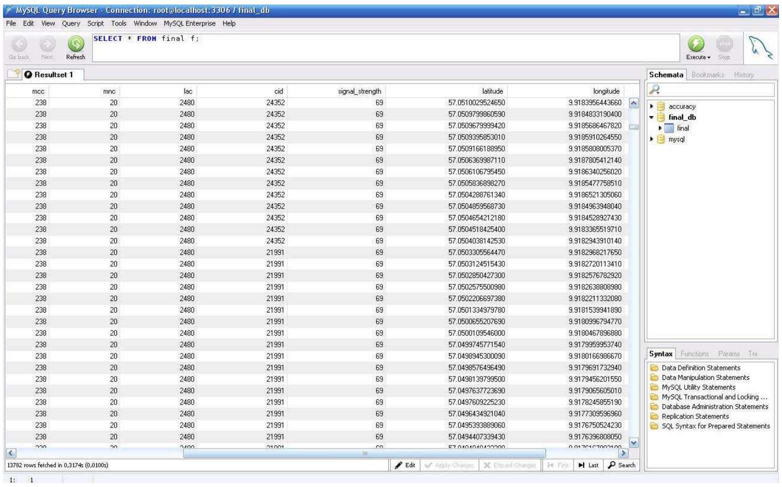


**Figure 3.9:** Table containing the data for the elaboration

the database varies for each couple (BS, SS). The response time varies according to the number of the selected records. Therefore the response time varies for each couple (BS, SS).

This means that on one hand for reaching a complete understanding of the BSs' placements more measurements must be done (this implies more records in the database). On the other hand for having a faster algorithm without a variable response time, it needs to have fewer records in the database. To solve this problem, one of the following solutions could be chosen:

1. Create a script for filtering the records in the database according to the amount of information they add,
2. Create another table with the possible BSs' placements. This solution needs a new (simpler) algorithm to be developed.

With the first solution, the algorithm is expected to be faster above all in the slowest cases because the number of records that will be selected decreases. Thus, the velocity would be less variable. This solution needs to be tested.

The second solution is potentially the best one. The eventual algorithm would make a sort of triangulation between the centers of the BSs. The response time would be always very short and there would not be any problem of variability.

Nevertheless, this could be a dangerous choice if the starting data have a high level of incompleteness. Indeed, in this case, there is a high risk that the calculation of the centers of the BSs will lead to wrong results. This means that we could not trust in the output of this system because, potentially, it will make a triangulation among three 'wrong' points. At least there would be high probability that they do not represent the real BSs' placements. As shown in Section 3.5.1 the data for the elaboration in our case are complete enough and, consequently, the results are valid.

This second solution (triangulation among BSs) is the one that should be used if the data were complete. Thus, if someone managed to retrieve somehow a complete set of data of the BSs' placements for the main operators of the world, this approach would lead to a new way of localization for mobile phones fast, quite efficient and economic. The algorithm in this case would have the potentiality to exploit all the information of the network (this prototype is taking into consideration a maximum of three operators). So potentially the performance in terms of localization area and response time could be much improved. There would not be any problem of variable response time on the server-side, because the algorithm would not need to calculate any distance between points.

## 3.5 Reliability

In this section the following fundamental questions has been answered:

1. *The available data are incomplete. How could this not invalidate the obtained results?* (see Section 3.5.1)

2. *Which are the expected results in the optimal conditions?* (see Section 3.5.2)

### 3.5.1 Reliability Confirm

The results are not invalidated because the level of partiality of the data is not a relevant one. In other words the available information of the BSs' placements in Aalborg is quite complete, even if it is not 100% complete.

Indeed, the measurement campaign was made carefully and the major part of the streets of Aalborg was included. This had implied that the same street (same zone) has usually been passed through more than one time.

Anyway it is important to prove somehow the reliability of the exploited data. For this reason the input data of the testing phase has been observed. The data from the network have been gathered two times: the first time for the measurement campaign and the second time, some months later, for testing purpose (to evaluate the accuracy of the system).

For each measurement tests it is relevant to observe where the GPS point is respect to the listened cells. The tests were run for three cooperative devices with SIM cards belonging to three different operators. It is possible to distinguish two cases:

- The GPS point is in all the three cells at the same time,

- The GPS point is not in all the three cells at the same time.

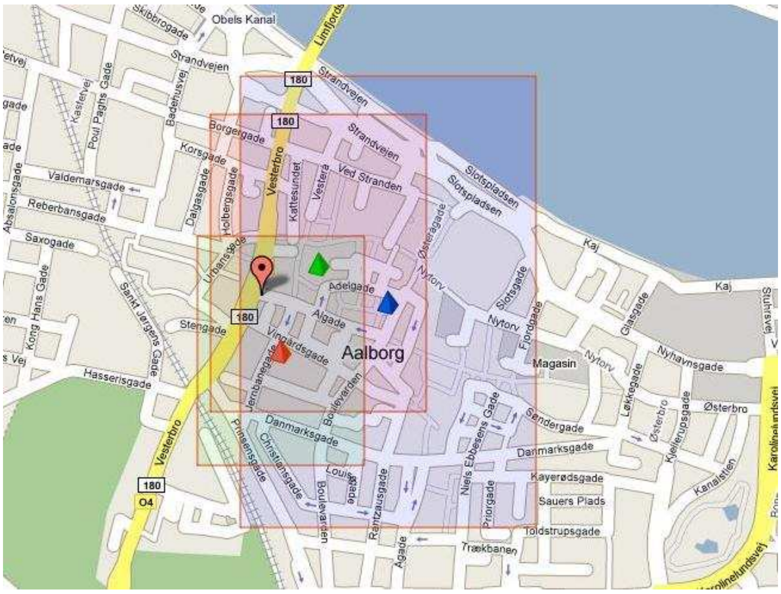



**Figure 3.10:** The GPS point is included in the intersection area of the three cells

In the first case the three cells intersect one other and the GPS is in the intersection area (see Figure 3.10). This case confirms that the data stored in the database for the elaboration are coherent with the new survey.

In the second case the GPS position, potentially, could not be in any cell or be in the area of only one (or two) of the three cells. As regards the listened cells in that point, these could also not even intersect one another. This means that this second case could be the symptom of both a relevant or irrelevant incompleteness of the data. For these reason, it is important to evaluate all the cases in order to understand the kind of incompleteness.

From the tests it results that in the 79.0% of the times the point is in all the three cells, while in the 21.0% it is not. By evaluating all the cases, it results that the incompleteness is an irrelevant one. Indeed, in this 21.0%, for only one case it happens that one of three cell IDs is not in the database and the remaining two cells does not intersect each other. Therefore, in this case the GPS point is included in only one cell. But this is a particular case, because it regards a marginal zone of the area covered by measurements (see Figure 3.11).

In all the other cases of this 21.0%, the GPS point intersects two of the three cells and the third one is really near to the GPS point (see Figure 3.12). In conclusion it is possible to claim that the incompleteness of the data used for the elaboration is not a relevant one and, therefore, it does not invalidate the accuracy results.

### 3.5.2 Impact estimate of the data incompleteness

The partiality of the data makes the 15% of the localization processes fail. It is possible to claim this because of the accuracy results for the "only cell ID" solution (see Section 3.7).


Figure 3.11: The GPS point is included in the area of only one cell

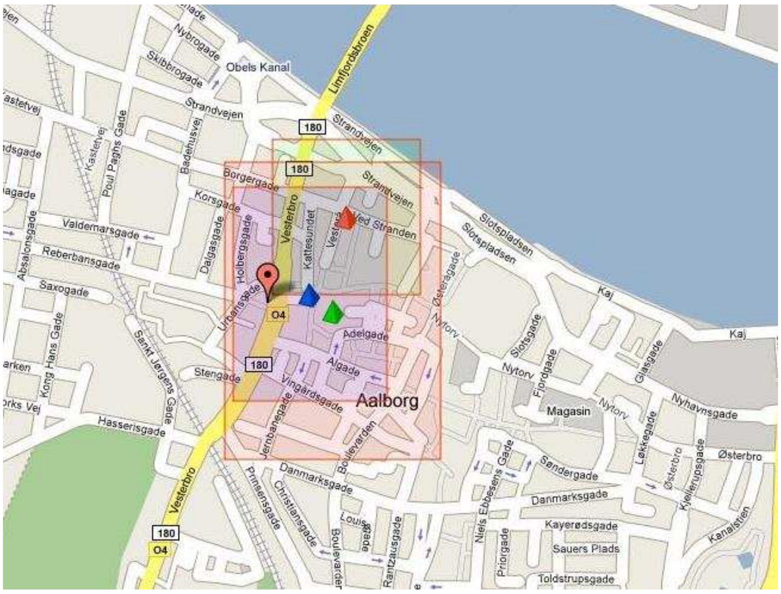



Figure 3.12: The GPS point is included in the intersection area of the two cells

As regards the influence on the remained 85%, how would the estimated area change in the optimal conditions?

In the optimal condition and considering an algorithm that does not use the signal strength, it is expected that:

1. There will not be any bad localization because, in the optimal conditions, the data will be complete.

2. In the localization processes with few cooperative devices the range of the estimated area could get worse. Indeed, in a complete vision of the cellular network the real area of the cells could be the same or larger than the ones of our vision. Therefore, the intersection of the cells could be the same or larger.

3. As the number of mobile devices increases, the performance of the algorithm are expected to get even better respect to the given accuracy results. That is because in the optimal conditions there would not be any failure and the amount of information that can be retrieved from the network would be larger, hence the area of localization could get extremely reduced.

## 3.6 Consideration related to the use of signal strength

As said in Section 3.1, the system was tested with two different algorithms:

- one that uses the signal strength parameter with initial *threshold* set to zero (*"signal strength + cell ID" solution*),
- one that does not use the signal strength parameter (*"only cell ID" solution*).

The accuracy results for the first algorithm have been given in Section 3.2. The accuracy results for the second solution are given in Section 3.6.1. In Section 3.6.2, the comparison between the two cases is made.

### 3.6.1 Accuracy results for the "only cell ID" solution

In this section the results for the "only cell ID" solution are given. For having a better comprehension of the train of thought behind this analysis, see Section 3.1 and Section 3.2.

Let $d$ be the distance between the center of the estimated area and the real position. In Table 3.5 the maximum, minimum and medium values of this parameter are shown as a function of the number of cooperative devices.

| n. Mobile Devices | 1 | 2 | 3 |
|---|---|---|---|
| **Maximum** | 522.20 m | 332.97 m | 274.94 m |
| **Average** | 134.83 m | 91.63 m | 73.53 m |
| **Minimum** | 15.53 m | 2.5 m | 1.07 m |

**Table 3.5:** Minimum, average and maximum value of *d* for all the localization tests ("only cell ID" solution)

| n. Mobile Devices | 1 | 2 | 3 |
|---|---|---|---|
| **Maximum** | 522.2 m | 333.0 m | 274.9 m |
| **Average** | 128.6 m | 83.0 m | 63.6 m |
| **Minimum** | 15.5 m | 2.5 m | 1.1 m |

**Table 3.6:** Minimum, average and maximum value of *d* for the good localizations ("only cell ID" solution)

#### The GPS position is included in the estimated area

Let $r$ be the range of the estimated area (see Figure 3.1). In Tables 3.6 and Table 3.7 the maximum, minimum and medium values respectively of $d$ and $r$ are shown as a function of the number of cooperative devices.

Figure 3.13 shows how the localization area decreases according to the number of cooperative devices.

#### GPS position is not included in the estimated area

In Section 3.2, it has been said that the distance between the estimated area and the GPS position is given by $(d - r)$, where $d$ is distance between the center of the estimated area and the GPS position and $r$ is the range of the estimated area.

In Table 3.8 the maximum, minimum and medium values of this parameter are shown as a function of the number of cooperative devices.

### 3.6.2 Comparison between the two solutions

For this comparison only the medium accuracy values have been considered because they are the most relevant ones. Using the "only cell ID" solution the following results get better:

1. $d$ calculated on all the tests (see Table 3.9),
2. $d$-$r$ calculated on the bad localizations (see Table 3.10),

| n. Mobile Devices | 1 | 2 | 3 |
|---|---|---|---|
| Maximum | 682.5 m | 460.9 m | 451.3 m |
| Average | 333.4 m | 190.9 m | 145.3 m |
| Minimum | 60.1 m | 15.5 m | 15.5 m |

**Table 3.7:** Minimum, average and maximum value of $r$ ("only cell ID" solution)

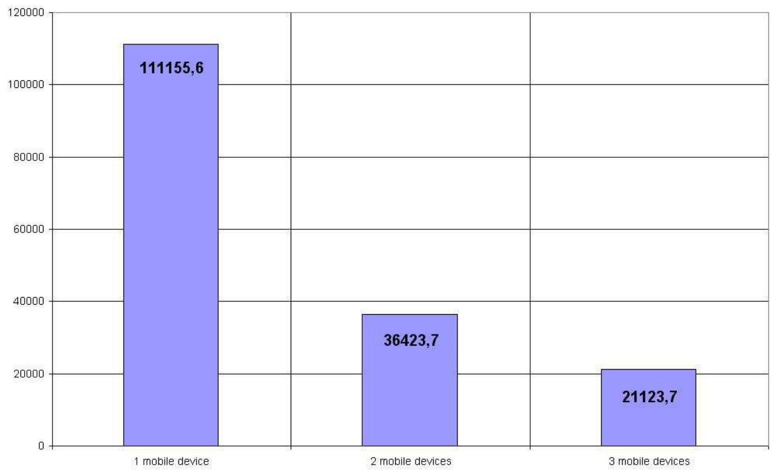



**Figure 3.13:** Estimated area as a function of the number of cooperative devices ("only cell ID" solution )

3. $d$ calculated on the good localizations (see Table 3.11).

Anyway, using the "only cell ID" solution the following parameters get worse:

1. $r$ calculated on the good localizations (see Table 3.12),
2. the response time.

Without using the signal strength, the results gain as regards the distance between the estimated and the real position, but the range of the estimated area is larger and the response time is longer.

Also for this solution, all the trends have shown that the larger the number of cooperative devices, the smaller the location error $d$ and the higher is the location accuracy. The proposed schema with three cooperative mobile devices can still reduce the estimated area of more than five times compared to a non cooperative scenario.

In the next section, the analysis of both the solutions will be completed, thus other considerations related to the use of signal strength will be done later.

| n. Mobile Devices | 1 | 2 | 3 |
|---|---|---|---|
| **Maximum** | 142.8 m | 209.6 m | 211.0 m |
| **Average** | 63.6 m | 40.4 m | 43.35 m |
| **Minimum** | 2.7 m | 0.7 m | 0.3 m |

**Table 3.8:** Minimum, average and maximum value of *d-r* ("only cell ID" solution )

| n. Mobile Devices | 1 | 2 | 3 |
|---|---|---|---|
| **Improvement** | -34 m | -30 m | -1 m |

**Table 3.9:** Improvement for *d* on all the tests

## 3.7 Final Discussion

Since the beginning of this chapter it has been distinguished the analysis for 'good' and 'bad' localization processes. If we knew the percentages of the two cases for both the implementations, we would be able to define an *index of validity* for the accuracy results.

During the analysis of this chapter two solutions has been treated. The "signal strength + cell ID" solution aims to reach the smallest estimated area and elaboration time. A calibration parameter lets us vary the range of the first set selection. By varying this parameter the weight of the *calibration errors* increases or decreases.

The "only cell ID" solution aims to reach the most reliable results. Because the signal strength parameter is not considered, it is like if the weight of the *calibration errors* has been set to zero. Thus, all the errors in this case are due to the above mentioned incompleteness of the data. Is there a way to vary the weight of the *data errors* too?

The definition of *failure* in a localization process, used so far, has been the following:

**Definition 3.7.1** Definition of failure (strict): *Let us consider a test process. The process fails if:*

$$(d - r) < 0 \tag{3.1}$$

*Where:*

- d *is the distance between the center of the estimated area and the GPS position,*
- r *is the range of the estimated area.*

This failure could be due to a *calibration error* or to a *data error*. The "signal strength + cell ID" solution has both the two kind of errors. In the "only cell ID" solution there is not any *calibration error*.

In Section 3.5.2 we said that if the data were complete, we would expect cells equal or larger than the ones of our vision. As a consequence, the expected area would be larger

| **n. Mobile Devices** | 1 | 2 | 3 |
|---|---|---|---|
| **Improvement** | -108 m | -78 m | -22 m |

**Table 3.10:** Improvement for *d-r*

| **n. Mobile Devices** | 1 | 2 | 3 |
|---|---|---|---|
| **Improvement** | -18 m | -15 m | -12 m |

**Table 3.11:** Improvement for *d* on the good localizations

too. If in this definition of *failure* we added a deviation parameter in order to reach a less strict definition, the *data errors* would be somehow ‘compensated’.

**Definition 3.7.2** Definition of failure (less strict): *Let us consider a test process. The process fails if:*

$$(d - r) < s \tag{3.2}$$

*Where:*

- d *is the distance between the center of the estimated area and the GPS position,*
- r *is the range of the estimated area,*
- s *is the deviation.*

It is like if we considered a larger range for the estimated area. Thus, with this new definition, a localization process is a good one if the GPS position is in the estimated area or ‘near’ to it.

The results given so far are related to the Definition 3.7.1 of *failure*. The following are the percentages of validity for both the solutions:

**“Signal strength + cell ID” solution:**

- GPS position is included in the estimated area: 50.8%
- GPS position is not included in the estimated area: 49.2%

**“Only cell ID” solution:**

- GPS position is included in the estimated area: 84.3%
- GPS position is not included in the estimated area: 15.7%

If the Definition 3.7.2 of *failure* is considered (with $s = 35\ m$) the percentages of validity are:

| n. Mobile Devices | 1 | 2 | 3 |
|---|---|---|---|
| **Worsening** | +62 m | +55 m | +33 m |

**Table 3.12:** Worsening for *r*

**"Signal strength + cell ID" solution:**

- GPS position is included in the estimated area: 65.8%
- GPS position is not included in the estimated area: 34.1%

**"Only cell ID" solution:**

- GPS position is included in the estimated area: 94.2%
- GPS position is not included in the estimated area: 5.7%

Tables 3.13, 3.14, 3.15, 3.16 show the medium accuracy values for all the treated cases.

| **"SIGNAL STRENGTH + CELL ID" SOLUTION** | | |
|---|---|---|
| ***All the localization tests*** | | |
| Num. Mobile Devices | d | |
| 1 | 168.6 m | |
| 2 | 121.2 m | |
| 3 | 74.5 m | |
| ***GPS position is in estimated area* (50.8%)** | | |
| Num. Mobile Devices | d | r |
| 1 | 110.0 m | 271.3 m |
| 2 | 68.6 m | 135.9 m |
| 3 | 51.1 m | 112.3 m |
| ***GPS position is not in estimated area* (49.2%)** | | |
| Num. Mobile Devices | d - r | |
| 1 | 171.5 m | |
| 2 | 118.3 m | |
| 3 | 65.8 m | |

**Table 3.13:** Medium accuracy values for "signal strength + cell ID" solution (*Definition 3.7.1 of failure*)

| "SIGNAL STRENGTH + CELL ID" SOLUTION | | |
|---|---|---|
| ***All the localization tests*** | | |
| Num. Mobile Devices | d | |
| 1 | 168.6 m | |
| 2 | 121.2 m | |
| 3 | 74.5 m | |
| ***GPS position is in estimated area* (65.8%)** | | |
| Num. Mobile Devices | d | r |
| 1 | 118.2 m | 265.2 m |
| 2 | 68.2 m | 110.1 m |
| 3 | 52.0 m | 91.3 m |
| ***GPS position is not in estimated area* (34.1%)** | | |
| Num. Mobile Devices | d - r | |
| 1 | 197.7 m | |
| 2 | 161.0 m | |
| 3 | 115.0 m | |

**Table 3.14:** Medium accuracy values for "signal strength + cell ID" solution (*Definition 3.7.2 of failure, s = 35 m*)

What we can see is that the larger is the deviation $s$, the nearest to 100% is the percentage of successes in the localization. This quite obvious result confirms that, in the optimal conditions, there will not be any bad localization because the available data will be completed in that case (said in Section 3.5.2).

Anyway, we made this analysis not for having this confirmation but for reaching a major understanding of what should be expected in the optimal conditions. In other words, after all these results, we want to focus on the most reliable results of accuracy in the optimal conditions for three cooperative devices. 'Optimal condition' means that 100% complete data are available.

Of course this considerations refer to the "only cell ID" solution because it is the one without *calibration errors*.

The "only cell ID" solution with the s = 35 m is the most reliable one because there is not any *calibration error* and because the influence of the incompleteness of the data has been reduced using the $s$ parameter (the *index of validity* is the highest one: 94,2%).

As regards the $d$ parameter, the results of this (most reliable) solution say that $d$ is about 66 m. In this case, another quite reliable value for $d$ is the one calculated on all the localization tests (73 m). It is reliable because it considers also the worst cases. As a

| "ONLY CELL ID" SOLUTION | | |
|---|---|---|
| ***All the localization tests*** | | |
| Num. Mobile Devices | d | |
| 1 | 134.8 m | |
| 2 | 91.6 m | |
| 3 | 73.5 m | |
| ***GPS position is in estimated area* (84.3%)** | | |
| Num. Mobile Devices | d | r |
| 1 | 128.6 m | 333.4 m |
| 2 | 83.0 m | 190.9 m |
| 3 | 63.5 m | 145.3 m |
| ***GPS position is not in estimated area* (15.7%)** | | |
| Num. Mobile Devices | d - r | |
| 1 | 63.6 m | |
| 2 | 40.4 m | |
| 3 | 43.4 m | |

**Table 3.15:** Medium accuracy values for "only cell ID" solution (*Definition 3.7.1 of failure*)

consequence, it is reasonable to expect that the medium value of $d$ for three cooperative mobile devices will be included in the range [66, 73] m.

As regards the range $r$ of the estimated area, the value obtained in the "only cell ID" solution with s = 35 m is not that reliable. Indeed, it is inferior respect to the one obtained with the same solution, s = 0 m (Tables 3.15, 3.16). And that is because, the estimated areas in the added 9.9% of the cases (that has been added considering the s parameter equal to 35 m) were too small to include the GPS position. Thus, this 9.9% of cases has reduced wrongly the medium value of $r$. In conclusion, the most reliable result for $r$ is 145 m (medium value for three cooperative devices, obtained for the "only cell ID" solution with s = 0 m).

From this analysis it emerges that even though, on one hand the use of signal strength reduces the response time and the range of the estimated area, on the other hand it leads to a large number of errors in the localization tests if the *threshold* parameter has been not well calibrated. Therefore, for having a real improvement by the use of the signal strength, a very careful analysis should be made in order to build a model of its variation for the whole area covered by measurements.

| "ONLY CELL ID" SOLUTION | | |
|---|---|---|
| ***All the localization tests*** | | |
| Num. Mobile Devices | d | |
| 1 | 134.8 m | |
| 2 | 91.6 m | |
| 3 | 73.5 m | |
| ***GPS position is in estimated area* (94.2%)** | | |
| Num. Mobile Devices | d | r |
| 1 | 130.9 m | 327.0 m |
| 2 | 87.1 m | 181.2 m |
| 3 | 66.3 m | 130.7 m |
| ***GPS position is not in estimated area* (5.7%)** | | |
| Num. Mobile Devices | d - r | |
| 1 | 105.3 m | |
| 2 | 93.8 m | |
| 3 | 81.0 m | |

**Table 3.16:** Medium accuracy values for "only cell ID" solution (*Definition 3.7.2 of failure, s = 35 m*)

## 3.8 Conclusions

The main result is that the proposed schema with three cooperative mobile devices can reduce the estimated area of more than five times compared to a non cooperative scenario. For three cooperative mobile devices, the most reliable accuracy results are 145 m for the range of the estimated area and, as regards the distance between the GPS position and the center of the estimated area, this is about 70 m.

From this analysis it emerges that even though, on one hand the use of signal strength reduces the response time and the range of the estimated area, on the other hand it leads to a large number of errors in the localization tests if the *threshold* parameter has been not well calibrated. Therefore, for having a real improvement by the use of the signal strength, a very careful analysis should be made in order to build a model of its variation for the whole area covered by measurements.

This system has been developed for research purpose. To make it work better and in a larger scale, the following solutions could be followed:

- a careful measurement campaign should be run,

- the most efficient filtering of the data should be developed to store in the database only the essential records,
- finally it could be useful to find a coarse function for calculating the distance between two points.

In alternative, after having collected a large amount of data from the network, it would be clever to evaluate the reached level of completeness in order to create a database with the possible BS placements. This solution needs a new algorithm to be developed. The eventual algorithm would make a sort of triangulation based on the centers of the BSs. The response time would be short and not variable at all.

This second algorithm is the one that should be used if someone managed to retrieve a complete set of data of the BSs' placements for the main operators of the world, this approach would lead to a new way of localization for mobile phone fast, quite efficient and cost-effective. The system in this case would have the potentiality to exploit all the information of the network (this prototype is taking into consideration a maximum of three operators). So potentially the performance in terms of localization area and response time could be improved a lot. There would not be any problem of variable response time because the algorithm would not need to calculate any distance between points.

# Conclusion and Outlook

During the present research work, we looked for the most appropriate solutions to demonstrate the validity of the proposed localization approach and to determine the order of magnitude of the achievable improvement. To develop and experimentally test a positioning cellular system and to reach accuracy results, the following has been done:

- gathering of the data related to the BS's placements,
- processing and analysis of these data,
- development of a prototypical system with a server-client based architecture that realizes cooperative localization,
- accuracy tests and analysis of the results,
- developing of some web application for observing the stored data and the results of the accuracy tests.

The main idea is, at the same time, very simple and efficacious. It is also free of charge and does not have special hardware requirements. Indeed, the minimal requirements are Bluetooth and GPRS. Furthermore, the higher the number of operators in a certain area, the better the performance.

The main result is that the proposed schema with three cooperative mobile devices can reduce the estimated area of more than five times compared to a non cooperative scenario. Moreover, for three cooperative mobile devices, the distance between the GPS position and the center of the estimated area is about 70 m and, as regard the range of the estimated area, this is about 145 m.

As regards the limitation of the developed prototype, it needs a database to be created first. However, as said in Chapter 3, all the problems of this system could be solved if the data for the elaboration were available. In this case, this solution would lead to a new way of localization for mobile phones fast, quite efficient and cost-effective.

A possible future enhancement could be an implementation of the client side application that can work in offline mode. At the moment, to localize itself, a mobile has to make a request to a server. Future research can investigate the possibility for an end-user to download the whole application, with the data for a given area, only once for later offline usage.

# Appendix A

# CoopLoc - Measurements

## A.1 Introduction

When we started, the BS's placements were not available. Therefore a measurement campaign was delivered. We, basically, gathered the data for the elaboration. This has been an essential phase of the project. Indeed, we could not have started without having the BSs' placements.

To deliver the measurement campaign, we developed *CoopLoc - Measurements*, the subject of this appendix. *CoopLoc - Measurements* is an application that runs on Nokia N95 and it has been developed using Python for S60 [11].

In Section A.2 the source code is given. In Section A.3 a short description of the code is made.

## A.2 Source Code

**Listing A.1:** CoopLoc-Measurements

```
import appuifw
import os
import e32
from time import *
from location import *
from sysinfo import *
from positioning import *
from graphics import *

###all the functions

def init(dirName):
    global path
    LIST = os.listdir("e:\\Measurements\\")
    if not dirName in LIST:
        os.mkdir("e:\\Measurements\\"+dirName+"\\")
    NewFolder = list(localtime())[0:5]
    NewFolder1 = NewFolder[0:3]  #first 3 - date
```

```
    NewFolder2 = NewFolder[3:5]  #last 2 - time
    NewFolder1 = "".join(str(NewFolder1)).replace(", ","-")
    NewFolder2 = "".join(str(NewFolder2)).replace(", ",".")
    NewFolder = NewFolder1[1:len(NewFolder1)-1]+" " + \
                NewFolder2[1:len(NewFolder2)-1]
    folders = os.listdir("e:\\Measurements\\" + dirName)
    if not NewFolder in folders:
        os.mkdir("e:\\Measurements\\" + dirName+"\\" + NewFolder)
        path="e:\\Measurements\\" + dirName + "\\" + NewFolder + \
          "\\" + str(imei()) + ".txt"

def measurements(file_name):
    timew = str(clock())
    gsm_loc= str(gsm_location()) ###BS ID
    signal_strength = str(signal_dbm()) ###dbm
    set_requestors([{"type":"service", "format":"application", \
                    "data":"test_app"}])
    #non blocking call
    GPS_pos = str(position(course=1, satellites=1))
    file_name.write( timew + '%' + gsm_loc + '%' + \
                    signal_strength + '%' + GPS_pos + '%\n')
    print gsm_loc, signal_strength

def start():
    global path
    global measurements
    appuifw.note(u"Survey of GPS data in progress", "info")
    set_requestors([{"type":"service", "format":"application", \
                    "data":"test_app"}])
    signal=str(position())#blocking call
    if(signal!=""):
        appuifw.note(u"GPS data detected", "conf")
    ask_confirm=appuifw.query(u'Select ok to start','query')
    LIST = os.listdir("e:\\")
    if not 'Measurements' in LIST:
        os.mkdir("e:\\Measurements\\")
    Places = os.listdir("e:\\Measurements\\")
    Places.append('Other')
    pos_Other = Places.__len__()
    pos_Other = pos_Other - 1
    Itemize = []
    m=0
    for i in Places:
        Itemize.append(u"" + Places[m])
        m=m+1
    selection = appuifw.popup_menu(Itemize, u"Select and press OK:")
    if selection == pos_Other:
        NewPlace = appuifw.query(u'Insert New Place','text')
        appuifw.note(NewPlace + " Inserted", "conf")
        init(NewPlace)
    else:
        init(Itemize[selection])
        appuifw.note(Itemize[selection], "conf")
    print "started"
    measurements = 'true'
    file_name = open(path, 'a')
    appuifw.note(u"Measurements start", "info")
    global old_body
    appuifw.app.body = old_body
    while (measurements == 'true'):
        e32.ao_sleep(2)
        measurements(file_name)
    file_name.close()

```

```
83 def stop():
84     global measurements
85     measurements = 'false'
86
87 def exit_key_handler():
88     app_lock.signal()
89
90 ### graphic aspects
91 global old_body
92 old_body=appuifw.app.body
93 appuifw.app.screen='large'
94 appuifw.app.title = u"Measurements"
95 appuifw.app.menu = [(u"Start Survey", start),(u"Stop Survey", stop)]
96 img=Image.open("e:\\MAIN.jpg")
97 appuifw.app.body=c=appuifw.Canvas()
98 c.blit(img,scale=1)
99
100 app_lock = e32.Ao_lock()
101 appuifw.app.exit_key_handler = exit_key_handler
102 app_lock.wait()
```

## A.3 Short code explanation

When the 'Start Survey' option is selected from the main menu of the application (see screenshot in Figure 2.5(b)), the *start()* function is called (lines 41-80).

The *start()* function waits for the GPS signal (see Figure 2.5(c)). This has been implemented through the *blocking call* of Python for retrieving the GPS signal. Indeed, the *positioning* module has two way of retrieving the GPS information: *blocking* and *non blocking* call. The first one (see line 47) makes the application wait until the mobile has received the information from the satellites. The second one (see line 36) gets the information available without waiting (thus if no information is available, it returns a Python dictionary with undefined values).

Afterwards, it is possible to select the path for the survey. If the *Other* option is chosen, you can insert the name of a new path (see screenshot in Figure 2.5(d)). Next, the *init()* function is called (lines 12-28). This function organizes the folder's structure under the root folder of the application (E:/Measurements) which, at the end, will contain a folder for each path, named with the path name. And each path folder will contain as many folders as the number of times that a survey has been run for that path. Each folder will be named with date and time of the survey.

Then the measurement starts and, who is using this application is supposed to walk along a path. In this phase, the application calls every two seconds the *measurements()* function (lines 30-40) that stores in a txt file a timestamp, the BS ID, the signal strength and the GPS information. In the meantime, the application displays on the screen the BS ID and signal strength.

To stop the survey, choose the option 'Stop Survey'. In this way the *stop()* function (lines 83-85) sets to false the global variable *measurements* and, consequently, the survey will be stopped.

We walked along the main streets of Aalborg while GPS position, BS ID and signal strength of every mobile phone had been storing for post processing. This is what we call *Measurement Campaign*. We did this with three N95s equipped with SIM cards belonging to different Danish operators (CBB, TDC, Telia) and run the above mentioned application.

# Appendix B

# CoopLoc - Processing

## B.1 Introduction

Before starting, we needed to know if the main idea was applicable in the scenario where we were working. The *processing* phase has been important because we figured out how many base stations are in the downtown of Aalborg and how they are placed.

During this phase we developed two Python scripts for processing the stored txt files. The main task of these two applications was to fill in a MySQL database that had been created before. This database contains: latitude, longitude, Mobile Country Code (MCC), Mobile Network Code (MNC), Local Area Code (LAC), Cell ID (CID), signal strength, name of the path, name of the operator.

At that point, a MySQL database with all the necessary information was available and, therefore, it was possible to develop a web application using PHP and Javascript for visualizing the retrieved data on a Google Map.

In Section B.2 the source code of the two Python application is given. In Section B.3 a short description of the code is made.

## B.2 Source Code

**Listing B.1:** From the server side algorithm

```
import os
final_path = "C:\\Processing\\finalTXT.txt"
final = open(final_path,'a')

#each line of the final path has the following format:
#operator%path_name%date%time%timestamp%mcc%mnc%lac%cid%\
#signal_strength%latitude%longitude%altitude

def FindTXT(folder):
    #folder must end with '\\'
    LIST = os.listdir(folder)
    k = 0
```

```
    for i in LIST:
        if (check_extension(LIST[k]) == "true"):
            Filter(folder + str(LIST[k]))
            print "ok"
        else:
            FindTXT(folder + LIST[k] + "\\")
            print str(folder + LIST[k] + "\\")
        k = k + 1

#check_extension
def check_extension(element):
    if(element.find(".") == -1):
        return "false"
    else:
        dot_pos = element.find(".")
        extension = element[dot_pos + 1:len(element)]
        if (extension == "txt"):
            return "true"

def Find_third(line):
    pos = 0
    for i in range(3):
        pos = pos + line.find("%")
        line = line[line.find("%") + 1:]
    return pos

def time_bs_ss(line):
    final_pos = Find_third(line) + 3
    old_line = line
    pos = line.find("%")
    time = line[0:pos]
    line = line[pos + 1:]
    #BS info
    pos_par1 = line.find("(")
    pos_comma = line.find(",")
    mcc = line[pos_par1 + 1:pos_comma]
    line = line[pos_comma + 2:]
    mnc = line[0:line.find(",")]
    line = line[line.find(",") + 2:]
    lac = line[0:line.find(",")]
    line = line[line.find(",") + 2:]
    cid = line[0:line.find(")")]
    line = line[line.find("%") + 1 :]
    ss = line[0:line.find("%")]
    #output
    output = time + "%" + mcc + "%" + mnc + "%" + lac + "%" + cid \
             + "%" + ss + "%"
    return output, final_pos

def Filter(path):
    file_input = open(path, 'r')
    data = file_input.readlines()
    initial_data = op_path_date_time(path)
    for i in data:
        final.write(initial_data)
        string, pos = time_bs_ss(i)
        final.write(string)
        new_line = i[pos:]
        lat, lon, alt = filterGPS(new_line)
        final.write(lat + "%" + lon + "%" + alt + "%\n")
    file_input.close()

#other info (operator, path_name, date and time)
def op_path_date_time(path):
```

```
    print path
    string = "C:\\Processing\\Measurements\\"
    new_path = path[len(string):]
    operator = new_path[0:new_path.find("\\")]
    new_path = new_path[new_path.find("\\") + 1:]
    name_path = new_path[0:new_path.find("\\")]
    new_path = new_path[new_path.find("\\") + 1:]
    data = new_path[0:new_path.find(" ")]
    new_path = new_path[new_path.find(" ") + 1:]
    time = new_path[0:new_path.find("\\")]
    print operator + "\n" + name_path + "\n" + data + "\n" + time \
          + "\n"
    output = operator + "%" + name_path + "%" + data + "%" + time \
             + "%"
    return output

def filterGPS(string):
    lat = "'latitude': "
    pos_lat = string.find(lat)
    string = string[pos_lat + len(lat):]
    latitude = string[0:string.find(",")]
    alt = "'altitude': "
    pos_alt = string.find(alt)
    string = string[pos_alt + len(alt):]
    altitude = string[0:string.find(",")]
    lon = "'longitude': "
    pos_lon = string.find(lon)
    string = string[pos_lon + len(lon):]
    longitude = string[0:string.find(",")]
    return latitude, longitude, altitude

FindTXT("C:\\Processing\\Measurements\\")
final.close()
```

**Listing B.2:** CoopLoc-Processing2

```
import os
import MySQLdb

#read from the file that cames out from CoopLoc - Processing 1
final_path="C:\\Processing\\finalTXT.txt"
final = open(final_path,'r')
data = final.readlines()
db=MySQLdb.connect(host="localhost",user="root",passwd= "", \
                   db = "final_db")
c=db.cursor()
for i in data:
    pos=i.find("%")
    operator=i[0:pos]
    i=i[pos +1:]
    pos=i.find("%")
    path_name=i[0:pos]
    i=i[pos +1:]
    pos=i.find("%")
    data=i[0:pos]
    i=i[pos +1:]
    pos=i.find("%")
    ora=i[0:pos]
    i=i[pos +1:]
    pos=i.find("%")
    time=i[0:pos]
    i=i[pos +1:]
    pos=i.find("%")
```

```
28         mcc=i[0:pos]
29         i=i[pos +1:]
30         pos=i.find("%")
31         mnc=i[0:pos]
32         i=i[pos +1:]
33         pos=i.find("%")
34         lac=i[0:pos]
35         i=i[pos +1:]
36         pos=i.find("%")
37         cid=i[0:pos]
38         i=i[pos +1:]
39         pos=i.find("%")
40         ss=i[0:pos]
41         i=i[pos +1:]
42         pos=i.find("%")
43         latitude=i[0:pos]
44         i=i[pos +1:]
45         pos=i.find("%")
46         longitude=i[0:pos]
47         i=i[pos +1:]
48         pos=i.find("%")
49         altitude=i[0:pos]
50         c.execute("""INSERT INTO final (operator, path_name, date, \
51     time, timestamp, mcc, mnc, lac, cid, signal_strength, latitude, \
52     longitude, altitude)VALUES(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s,\
53     %s, %s, %s)""", (operator, path_name, data, ora, time, mcc, mnc,\
54                         lac, cid, ss, latitude, longitude, altitude))
55     db.close()
56     final.close()
```

## B.3 Short code explanation

These two script elaborate the output of *CoopLoc - Measurements*. As said in Appendix A, the output of this application is a *Measurements* folder with a certain structure. The names of the subfolders contain information about the survey (date, time, path name).

*CoopLoc - Processing* filters all the available and necessary information and stores it in another TXT file that in the code is called *finalTXT.txt* (line 2). Before using *CoopLoc - Processing*, we suggest to rename the *Measurements* folder of the three mobile phones that has run the measurement campaign, with the name of the corresponding operators and, to put all the three renamed folders in a new *Measurements* folder. When the FindTXT() function (line 9-20) is called, this folder is sifted out and all the TXT files are processing.

*CoopLoc - Processing 2* simply opens the filtered *finalTXT.txt* and insert the data in a MySQL database created before.

We divided this elaboration in these two steps for checking the data before inserting them in the database.

# Appendix C

# CoopLoc - Server Side Algorithm

## C.1 Introduction

As said in Chapter 2, in this project two solutions have been developed named "signal strength + cell ID" solution and "only cell ID" solution. We will deliver their analysis according to the following steps:

- Analysis of the available data for the elaboration (see Section C.2)
- Algorithm as a black box (see Section C.3)
- Flow sheet of the procedure

In Section C.4 the analysis of the "signal strength + cell ID" solution is made. In Section C.5 the "only cell ID" solution is shortly examined.

## C.2 Analysis of the available data for the elaboration

The data available for the elaboration were stored in a MySQL database, whose most relevant records are: latitude, longitude, Mobile Country Code (MCC), Mobile Network Code (MNC), Local Area Code (LAC), Cell ID (CID), Signal Strength (SS). The quartet (MCC, MNC, LAC, CID) identifies a BS of an operator in a country. In particular:

- the MCC (Mobile Country Code) is indicative of the country,
- the MNC (Mobile Network Code) is indicative of the network operator,
- the LAC (Local Area Code) is related to one operator and define a quite large area,
- the CID (Cell ID) individuates the smallest unit of an operator's network.

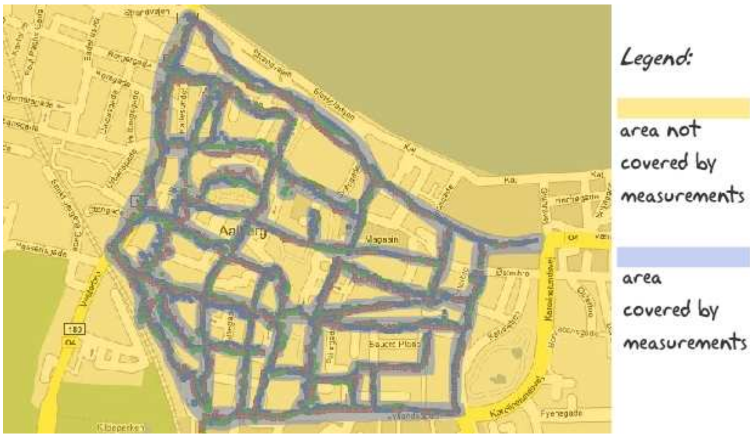



**Figure C.1:** Area covered and not covered by measurements


**Figure C.2:** The AD's position is included in the area covered by measurements

These data were our starting point. As shown in Figure C.1, the measurements points do not cover all the pixels of the map, indeed the measurements campaign has been run walking through the main streets of Aalborg. Thus, it is possible to distinguish between area covered and not covered by measurements. Consequently, two cases can occur:

- the AD's position is a point included in the area covered by measurements
- the AD's position is a point included in the area not covered by measurements (street not covered by measurements or indoor environment)

As regards the first case (see Figure C.2), saying that the AD's position is in an area covered by measurements implies that, in the database, there are some records that match more or less its GPS position. Our goal is to define an area that includes the AD's position. This means that it needs to select from the DB the measurements points near to the AD's position and define the area that includes all of them.

In the second case (see Figure C.3), there are no points in the database that match with the AD's position, so the goal is to find the nearest ones. Of course the area of localization will be larger.


Figure C.3: The AD's position is included in the area not covered by measurements

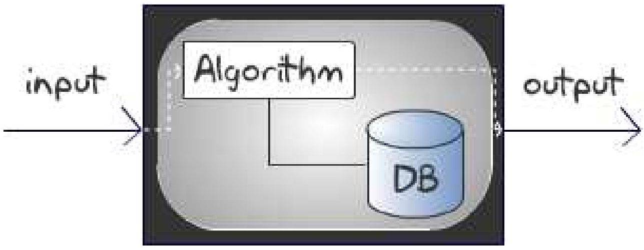



Figure C.4: Algorithm as a black box

If the data of the BS placement had covered all the pixels of the map (so the whole area of the city center of Aalborg), we would not have had this distinction and the accuracy would have been potentially the same in each point of the city.

## C.3 Algorithm as a black box

After these introductive considerations, let us consider our server as a black box (see Figure C.4) and let us go to analyze each element of the system (input, output and algorithm).

***Input*** After exchanging data with the NDs, the AD sends the collected data to the server according to the following format:

$$\{num, MCC1, MNC1, LAC1, CID1, SS1, \ldots, MCCn, MNCn, LACn, CIDn, SSn\}$$

Where:

- num is the number of cooperative devices

- (MCCn, MNCn, LACn, CIDn) is the BS information of the n-th device
- SSn is the signal strength of the n-th device

The first device is the AD.

***Output*** The estimated position is a circle. The algorithm returns the center and the range of the estimated area.

***Algorithm*** The algorithm is a PHP script that elaborates the inputs considering the data for the elaboration. See Section C.4 and C.5 for the flow sheets of the procedures.

## C.4 Flow sheet of the procedure for the "signal strength + cell ID" solution

The flow sheet is shown in Figure C.5. The steps of the procedure can be summarized as follows:

*STEP 1 - Check for BS ID:* First the algorithm checks if the BS IDs in input are in the database.

*STEP 2 - Set selection:* If BS is not in the database, that input is not considered. If BS is in the database a not empty set is defined.

**Listing C.1:** Set selection (with signal strength)

```
<?php
//(...)

$ss_m= $ss - $threshold;
$ss_M= $ss + $threshold;
$query = SELECT * FROM $table
         WHERE mcc = $mcc AND mnc = $mnc AND
               cid = $cid AND lac = $lac AND
               (signal_strength >= $ss_m AND
               signal_strength <= $ss_M);

//(...)
?>
```

At the beginning the *threshold* parameter is set to zero. If no points are found, the range of signal strength variation (the *threshold* parameter) is increased.

*STEP 3 - Nearest Points in the 10 meter range:*

As said in Section 2.5.1, we distinguish two kinds of set: the *Applicant Candidate Location Set* (ACLS) and the *Neighbor-Aided Location Set* (NALS). During this step of the procedure, the algorithm aims to reduce the ACLS by selecting from it only the points which include in their 10 meters-range other points belonged to NALSs. It is like

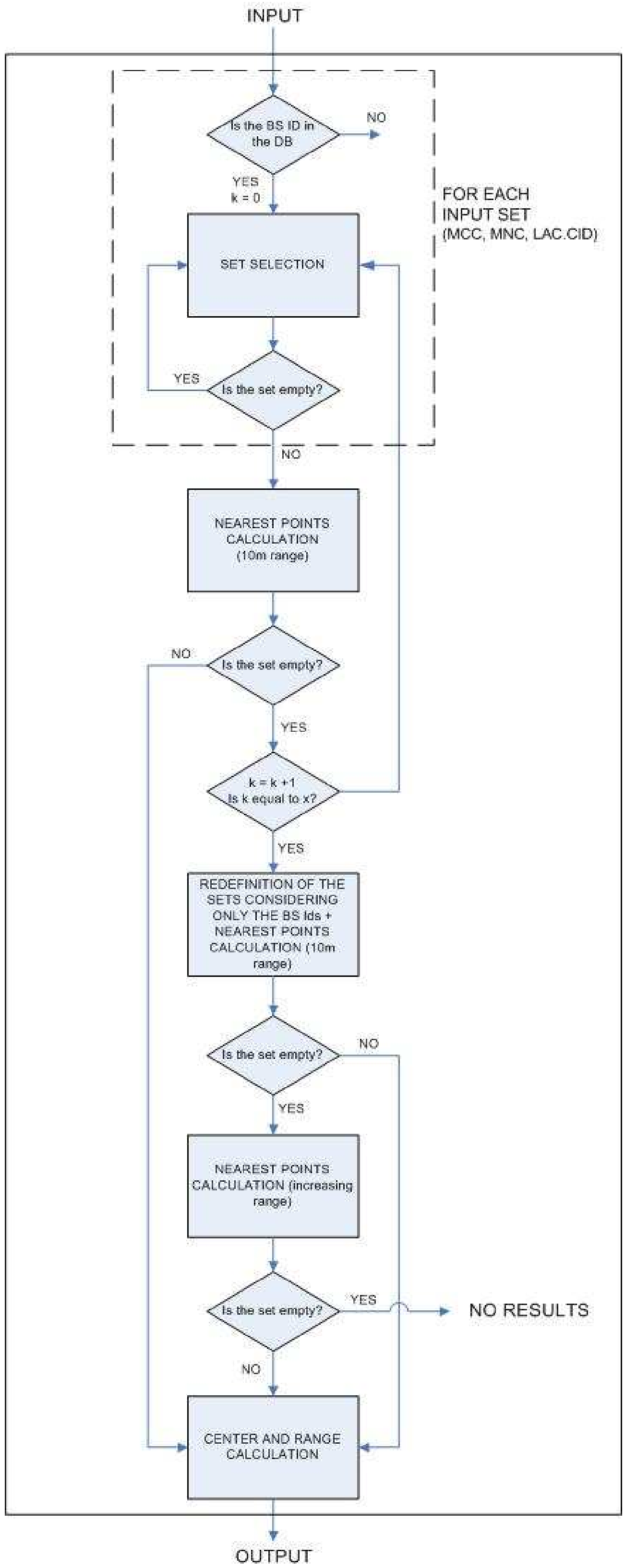



**Figure C.5:** Flow sheet of the procedure for the "signal strength + cell ID" solution if a 10m-range circle is drawn around each point of the ACLS. If one point, belonged to one of the other two sets, is included in this circle, the ACLS's point is selected for the output subset. In this moment it is supposed that the AD's position is in the area covered by measurements, thus we are looking for the points nearest to the AD's position.

*STEP 4:* if no points are found the range is increased. This leads to a new set definition (step 2). Afterwards the third step is repeated. Doing in this way the number of points in the sets is supposed to get larger. Therefore it is more likely that the nearest points function gives a result set.

*STEP 5:* If no points are found the sets are redefined without using signal strength variation, only exploiting the BS information. Thus, all the records that match the BS ID are selected. Afterwards the third step is repeated.

*STEP 6:* If the algorithm is arrived at this step, it means that it could not find any result. This could imply that the AD's position is not in an area covered by measurements. Thus, it needs to look for points further from the AD. Therefore, the 10 meters range is increased.

*STEP 7:* At this step, a filtered subset of points is available. The smallest area that includes all the points in the subset is the estimated area. Its center and range are the final output of the algorithm.

The algorithm could be calibrated to better consider the signal strength variation. The parameters that could be varied are the threshold parameter of the first set selection and the number x of times that the sets are rearranged before trying the solution without signal strength. For the analysis the threshold parameter has been set to zero and the x parameter ha been set to five.

## C.5 Flow sheet of the procedure for the "only cell ID" solution

The "only cell ID" solution changes respect to the above illustrated for the steps two and four. Indeed, the Set Selection phase is made without considering the signal strength parameter, only exploiting the BS ID. And, consequently, the fourth step cannot be included because it foresees to repeat the steps two and three after having increased the signal strength. The procedure starts directly according to the fifth step attempt.

Therefore, the new flow of sheet for this solution is the one in Figure C.6. The procedure can be summarized in the following steps:

*STEP 1 - Check for BS ID:* First the algorithm checks if the BS IDs in input are in the database

*STEP 2 - Set selection:* If BS is not in the database, that input is not considered. If BS is in the database a not empty set is defined.

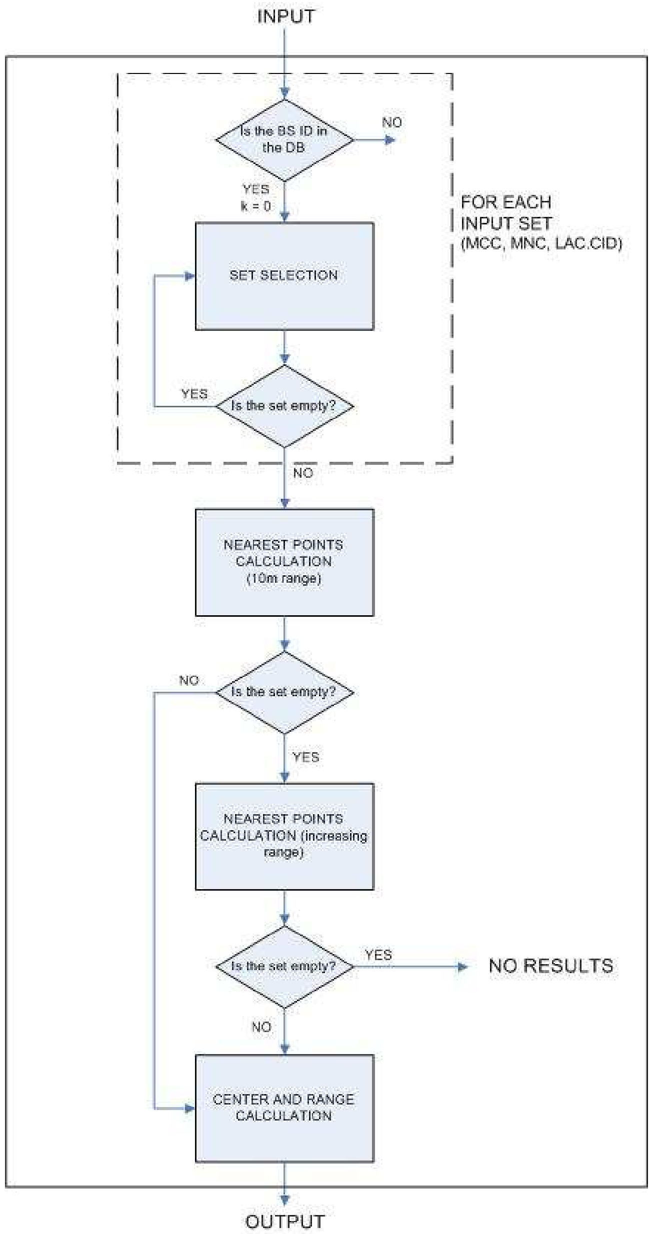



**Figure C.6:** Flow sheet of the procedure for the "only cell ID" solution

**Listing C.2:** Set selection (without signal strength)

```
<?php
//(...)

$query = SELECT * FROM $table
             WHERE mcc = $mcc AND mnc = $mnc AND
               cid = $cid AND lac = $lac;

//(...)
?>
```

The sets are defined without using signal strength parameter, only exploiting the BS information. Thus, all the records that match the BS ID are selected.

*STEP 3 - Nearest Points in the 10 meter range:*

During this step of the procedure, the algorithm aims to reduce the ACLS by selecting from it only the points which include in their 10 meters-range other points belonged to NALSs (see the third step (see the third step of Section C.4 for further information).

*STEP 4:* The 10 meters range is increased (see the sixth step of Section C.4 for further information).

*STEP 5:* At this step, a filtered subset of points is available. The smallest area that includes all the points in the subset is the estimated area. Its center and range are the final output of the algorithm.

# Appendix D

# CoopLoc

## D.1 Introduction

*CoopLoc* is the mobile application that uses cooperation among mobile devices to retrieve the location of the applicant device (AD). Its main task is to exchange data with the neighboring devices (NDs) and ask the server for the location with the collected data. The application has been developed using Python for S60.

In Section D.2 the procedures for the menu's options are described and the Python code for some issues is given.

## D.2 Procedures for the menu options

The options accessible from *CoopLoc*'s menu shortly are:

- AP selection
- Locate Alone
- Coop Location
- Tracking
- Help your friend
- About
- Exit

In the following subsections, motivations and procedures behind these options are outlined and the Python code for the main issues is given.

### D.2.1 AP selection

As said previously, the main idea is to send the data, collected by cooperating, to a server and wait for the location estimate. Thus, *CoopLoc* needs to access the network. By choosing the *AP selection* option, the end-user can select his favorite access point. The steps of this procedure are:

*STEP 1:* To show a list of all the available access points.

**Listing D.1:** List Available APs

```
import socket

apid = socket.select_access_point()
apo = socket.access_point(apid)
socket.set_default_access_point(apo)
```

*STEP 2:* To store the ID of the selected one in a txt file.

**Listing D.2:** Store the AP ID in a file

```
path_sett = DATA_DIR + '\\settings.txt'
f_sett = open(path_sett, 'w')
f_sett.write(str(apid))
f_sett.close()
```

In this way, all the time the application needs to access the network, it does not ask the user for the favorite AP.

### D.2.2 Locate Alone

If there is none for cooperating, the AD can locate itself using the option *Locate Alone*. The procedure contains the following steps:

*STEP 1:* To retrieve from the cellular network its own BS ID and signal strength.

**Listing D.3:** To retrieve BS ID and signal strength

```
import location
import sysinfo

location.gsm_position()
sysinfo.signal_dbm()
```

*STEP 2:* To send the retrieved data to the server and obtain the estimated location.

**Listing D.4:** To send data to a server

```
import httplib, urllib

#(...)

params = urllib.urlencode(diz)
headers = {"Content-type": "application/x-www-form-urlencoded", \
```

```
            "Accept": "text/plain"}
conn = httplib.HTTPConnection("XXX.XXX.XXX.XXX")
conn.request("POST", "/cooploc/script.php", params, headers)
response = conn.getresponse()
var = response.read()

#(...)
```

*STEP 3:* To visualize the result on a map (for *How to use Google Map data in mobile applications*, see [19]).

### D.2.3 Coop Location

If, in the AD's vicinity, there are some NDs, cooperation can be exploited. The NDs have to select the option *Help your friend*. The procedure of the option *Coop Location* contains the following steps:

*STEP 1:* To retrieve from the cellular network its own BS ID and signal strength (see Step 1 of Section D.2.2).

*STEP 2:* To look for NDs (using Bluetooth) and understand which of them are mobile phones.

**Listing D.5:** Look for NDs (mobile phones) and collect data from them

```
import lightblue

#(...)

devices = lightblue.finddevices()
for x in devices:
    device_class = lightblue.splitclass(devices[i][2])
    if (device_class[1] == 2): #this means that the BT device is a mobile
        addr = devices[i][0]
        i = i+1
        out = bt.connect(addr)
        if (out == 'OK'):
            GOOD_MAC.append(addr)
            resp = bt.readline()
            print "Received: " + resp
            (mcc,mnc,lac,cid,ss)= Filter_Data(resp)
            InsertDiz(mcc,mnc,lac,cid,ss)
            num_devices = num_devices + 1
        else:
            BAD_MAC.append(addr)
    else:
        i = i+1

#(...)
```

*STEP 3:* To collect data from the NDs.

*STEP 4:* To send the collected data to the server and obtain the estimated location (see Step 2 of Section D.2.2).

*STEP 5:* To visualize the result on a map (for *How to use Google Map data in mobile applications*, see [19]).

### D.2.4 Tracking

The *Tracking* option refreshes the localization constantly. Indeed, the application, after having found the NDs, asks for new information every seconds.

The steps of procedure are the same of the *Coop Location* option, but after the 5th step, the procedure starts again from the 3rd one until the end-user selects the option *Stop Tracking*.

### D.2.5 Help your friend

The NDs, for helping the AD, have to choose the option *Help your friend*. The application starts to listen on a certain port until a request arrives. When this happen, the ND retrieves its BS ID and signal strength from the network and sends them back to the applicant. Then, it starts to listen again.

**Listing D.6:** Work as a server

```
import socket

#(...)

server_socket = socket.socket(socket.AF_BT, socket.SOCK_STREAM)
server_socket.bind(("", p))
server_socket.listen(1)
socket.bt_advertise_service( u"CoopLoc", server_socket, True, \
                             socket.RFCOMM)
socket.set_security(server_socket, socket.AUTH)
(sock,peer_addr) = server_socket.accept()
my_bs = gsm_location()
my_ss = signal_dbm()
string = str(my_bs[0]) + "%" + str(my_bs[1]) + "%" + str(my_bs[2])\
         + "%" + str(my_bs[3]) + "%" + str(my_ss) + "%"
sock.send(string+'\n')

#(...)
```

# Bibliography

[1] E. S. Agency. http://www.esa.int/esana/sem3odshkhf_galileo_0.html.

[2] S. W. Boston. http://www.shyhookwireless.com/.

[3] J. J. Caffery. A new approach to the geometry of toa location. In *Vehicular Technology Conference, IEEE (VTS-Fall VTC)*. IEEE Computer Society, Tokyo, Japan, 2000.

[4] M. Y. Chen, T. Sohn, D. Chmelev, D. Hähnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. E. Smith, and A. Varshavsky. Practical metropolitan-scale positioning for gsm phones. In P. Dourish and A. Friday, editors, *Ubicomp*, volume 4206 of *Lecture Notes in Computer Science*, pages 225–242. Springer, 2006.

[5] COST-231. Urban transmission loss models for mobile radio in the 900 and 1800 mhz bands. Technical report, European Cooperation in the Field of Scientific and Technical Research (COST), 1991.

[6] (E112). http://www.europa.eu.int/comm/environment/civil/prote/112/112_en.htm.

[7] S. Feng and C. L. Law. Assisted gps and its impact on navigation in intelligent transportation systems. *International Conference on Intelligent Transportation systems*, Singapore, 2002.

[8] F. Fitzek and M. Katz, editors. *Cooperation in Wireless Networks: Principles and Applications – Real Egoistic Behavior is to Cooperate!* ISBN 1-4020-4710-X. Springer, April 2006.

[9] F. Fitzek and M. Katz, editors. *Cognitive Wireless Networks: Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications.* ISBN 978-1-4020-5978-0. Springer, July 2007.

[10] F. Fitzek and F. Reichert, editors. *Mobile Phone Programming and its Application to Wireless Networking.* Springer, 2007.

[11] M. Hata. *Empirical formula for propagation loss in land mobile radio services*, volume 29, pages 317–325. IEEE Trans. on Vehicular Technology, 1980.

[12] M. Hata and T. Nagatsu. *Mobile Location using Signal Strength Measurements in a Cellular System*, volume 29, pages 245–252. IEEE Trans. on Vehicular Technology, 1980.

[13] M. Ibnkahla. *Signal Processing for Mobile Communications, handbook.* CRC PRESS, Washington D.C., 2005.

[14] G. *MyLocation. http://www.google.com/gmm/mylocation.html.*

*[15] E. Kaplan.* Understanding GPS: Principles and Applications. *Artech House, Boston, 2006.*

*[16] J. Kriegl. Location in cellular networks. Diploma thesis, Institute for Applied Information Processing and Communications, University of Technology Graz, Austria, 2000.*

*[17] G. Liang and H. L. Bertoni.* A new approach to 3-D Ray tracing for propagation Prediction in Cities*, volume 46, pages 853–863. IEEE Trans. on Antenna and Propagation, 1998.*

*[18] D. R. M. Meurer, S. Heilmann and P. W. Baier. A signature based localization technique relying on covariance matrices of channel impulse responses. In* Workshop on positioning, navigation and communication (WPNC)*, Hannover, Germany, 2005.*

*[19] G. M. on mobile phone. http://wiki.forum.nokia.com/index.php/How_to...*

*[20] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate gsm indoor localization. In M. Beigl, S. S. Intille, J. Rekimoto, and H. Tokuda, editors,* Ubicomp, *volume 3660 of* Lecture Notes in Computer Science*, pages 141–158. Springer, 2005.*

*[21] H. M. T. O. R. Yamamoto, H. Matsutani and H. Ohtsuka. Position location technologies using signal strengths in cellular system. In* VTC-spring*, Rhodes Island, 2001.*

*[22] D. D. S. Kyriazakos and M. Theologou. Localization mobile terminals, based on a hybrid satellite-assisted and network-techniques. In* Wireless Communications and Networking Conference (WCNC)*, Chicago, USA, 2000.*

*[23] I. R. Seattle. http://www.placelab.org/.*

*[24] M. M. C. Takenga.* Received Signal Strength based Fingerprint Positioning in Cellular Networks involving Neural Networks and Tracking Techniques. *PhD thesis, Gottfried Wilhelm Leibniz University Hannover, 2007.*

*[25] F. A. to Promote Competition and P. S. in EnhancedWireless 911 Services. http://www.fcc.gov/bureaus/wireless/news_releases/1999/nrwl9040.doc.*

*[26] K. I. Tokio. http://www.placeengine.com/en.*

*[27] C. Veness. http://www.movable-type.co.uk/scripts/latlong-vincenty.html.*

*[28] J. Zhu and G. Durgin. Indoor/outdoor location of cellular handsets based on received signal strength.* IEEE Electronics Letters, *2005.*