

POLITECNICO DI MILANO
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA



WhAC: a WiFi based Application for Indoor Customer Localization

Relatore: Prof. Antonio Capone
Correlatore: Ing. Alessandro Redondi

Tesi di Laurea di:
Roberto Molteni (matr. 734984)
Francesco Perini (matr. 735246)

ANNO ACCADEMICO 2010-2011

Abstract

Knowing the location of a user is very important for a great variety of services, especially in *context aware* and *ubiquitous* computing. Hundreds of applications that we use daily exploit informations about our position.

As for outdoor localization, GPS is the *de facto* standard, while, with regard to indoor localization, we are faced with different possible solutions, varying in the frame of reference, in type of signal used, in the algorithms, and may depend on inertial navigation systems.

So, nowadays there is no *silver bullet* in indoor localization, and the choice of a system with respect to another always involves a difficult compromise between accuracy and costs, besides user acceptance, scalability and security.

This work proposes our WhAC (*Where Are my Customers*) algorithm and application for indoor WiFi localization in *noisy* environments such as malls and supermarkets.

Our algorithm combines the benefits of simulating empirical propagation models in terms of lowering the *offline stage* effort with the unmatched precision of *fingerprinting* techniques, merging the two approaches together.

Our purpose is the real time tracking of customers moving around in the shopping area, in order to provide additional services, enhance profiling techniques and improve overall marketing effectiveness.

After showing and classify various localization techniques, we'll focus on those based on *Received Signal Strength Indication* (RSSI) like ours and we'll describe the algorithms present in literature and their performances, providing a

comparison with our obtained results.

Then we'll describe in details the context in which our solution works, exposing all the related issues arisen during the development of WhAC.

Finally, we will explain how our software application is structured and how it interacts with the final user.

Sommario

Conoscere la posizione di un utente è molto importante per un gran numero di servizi, specialmente nei campi del *context aware* ed *ubiquitous* computing. Centinaia di applicazioni di uso comune sfruttano informazioni legate alla nostra posizione.

Mentre per ciò che concerne la localizzazione in ambienti esterni il GPS è ormai lo standard *de facto*, per quanto riguarda la localizzazione in ambiente interno ci troviamo di fronte a diverse possibili soluzioni, che variano in base al tipo di segnale, agli algoritmi, al sistema di riferimento ed al possibile uso di tecniche di navigazione inerziale. Non è quindi ovvia la scelta di un sistema di localizzazione *indoor* in quanto bisogna considerare un difficile compromesso fra diverse componenti quali precisione, scalabilità, accettazione da parte degli utenti, sicurezza e costi di installazione, i quali non sono mai fattori concordanti.

Obiettivo del nostro lavoro è la creazione del sistema WhAC (*Where Are my Customers*), il cui scopo è la localizzazione ed il tracciamento degli utenti via rete WiFi in ambienti interni molto *rumorosi* dal punto di vista della propagazione del segnale, quali centri commerciali e supermercati.

WhAC, che si compone di un algoritmo ed un'applicazione, sfrutta la tecnologia WiFi al fine di permettere di localizzare clienti all'interno di spazi espositivi e offrire loro servizi personalizzati su dispositivi mobili, migliorare le tecniche di profilazione dei clienti ed aumentare l'efficacia complessiva delle tecniche di marketing.

Il nostro algoritmo combina i benefici nella prima fase del processo di loca-

lizzazione (*offline stage*) in termini di sforzi e tempi richiesti dalla simulazione di modelli empirici di propagazione con la sino ad oggi ineguagliata precisione delle tecniche di *fingerprinting*, fondendo assieme i due approcci, sinora considerati l'uno alternativo all'altro in letteratura.

Descriveremo in dettaglio il contesto in cui la nostra soluzione andrà ad operare ed i vincoli che esso impone, soffermandoci sulla risoluzione dei problemi emersi e fornendo un'analisi comparativa dei risultati ottenuti rispetto a quelli presenti in letteratura.

Infine, mostreremo come è strutturata la nostra applicazione software e come essa interagisce con l'utente finale.

Contents

1	Introduction	1
2	Localization systems and techniques	5
2.1	Localization	5
2.2	Applications of localization	8
2.3	Outdoor Localization	8
2.3.1	Global Positioning System	9
2.3.2	Cellular Networks	10
2.4	Indoor Localization	11
2.4.1	Signal Types	11
2.4.2	Signal Metrics	13
2.4.3	Metric Processing Methods	15
2.4.4	Inertial Navigation Systems	16
3	Localization based on Received Signal Strength Indication	21
3.1	Wireless Channel and RSSI	21
3.1.1	IEEE 802.11	21
3.1.2	RSSI: Received signal strength indication	24
3.1.3	Problems of Wireless Channel	24
3.2	Wireless LAN	28
3.2.1	Architecture of a Wireless LAN	29
3.2.2	Types of wireless LANs	30
3.3	Indoor Localization	32
3.3.1	Static Scene Analysis: Fingerprinting	32

3.3.2	Propagation Model	33
3.3.3	Classical Location Estimation Algorithms	37
3.4	Simulators and Planning Tools for Indoor Wireless Systems	38
3.4.1	WinProp by AWE Communications	38
3.4.2	WiSE: a Wireless System Engineering Tool by AT&T Bell Laboratories	41
3.4.3	RaPsor	42
3.4.4	WinPlanner	43
3.4.5	CINDOOR by University of Cantabria, Spain	43
3.4.6	EDX Signal Pro by Comarco Wireless Technologies . . .	44
4	The application scenario	49
4.1	User profiling in loyalty programs	49
4.2	New user-profiling techniques in retail supermarkets	50
4.3	The typical supermarket scenario	51
4.3.1	Typical store architecture	52
4.4	Evaluating possible technologies	53
4.4.1	Environmental Issues	54
4.4.2	Field requirements	55
4.4.3	Technology tradeoff	56
4.4.4	WiFi constraints	56
5	WhAC: the algorithm	61
5.1	Reducing fingerprinting offline stage effort	61
5.1.1	Impact of the Number of Data Points	62
5.1.2	Impact of the Number of Samples	63
5.1.3	Impact of User Orientation	64
5.2	Direct comparison of Simulator vs Fingerprinting in static local- ization	64
5.3	Static localization in noisy supermarket-like environments	68
5.4	Dynamic localization	71
5.5	Improving localization with Particle Filtering	73

6	WhAC: the software architecture	79
6.1	Problem Description	79
6.1.1	General Description	79
6.1.2	Detailed Description	80
6.1.3	Glossary	80
6.2	Actors of the system	81
6.3	Use case and scenarios	82
6.3.1	System Administrator	83
6.3.2	Client	84
6.4	System Architecture	88
6.4.1	Architectural Details	90
6.5	Supermarket Map	91
6.5.1	Database Design	91
6.6	Client Application	93
6.6.1	Server Service	93
6.6.2	Client App	97
6.6.3	Service Client	98
6.7	Test cases	101
7	Conclusion and future works	103
	Bibliography	106

List of Figures

3.1	Schema of the IEEE 802.11 standard	22
3.2	Signal recombination due to the multipath propagation	26
3.3	Schema of an indoor localization system	30
3.4	Interaction between Server and a Mobile Device during the offline stage	33
3.5	Interaction between Server and a Mobile Device during the online stage	34
3.6	Example of KNN classification	37
3.7	WinProp: Sample prediction with an indoor prediction model . . .	40
3.8	WiSE: Predicted coverage of a small office with two base stations	41
3.9	Illustration of the RaPSor Look and Feel	43
3.10	WinPlanner: example of signal coverage results in a section of a building	44
3.11	CINDOOR: coverage map	45
3.12	EDX Signal Pro: Predicted coverage area	46
4.1	Distance-depending path loss affected by signal fading.	54
4.2	RSSI variation over time.	58
5.1	The error distance versus the size of the empirical data set (on a log scale).	63
5.2	Linksys WRT54G	65
5.3	Fingerprinting: impact of the number of nearest neighbours on localization error	67

5.4	Cumulative density function of the localization error in a test area of approximately $160m^2$ with 7 AP.	67
5.5	Schema of our dynamic localization algorithm: Training phase . .	70
5.6	Simulator in noisy environment: impact of the number of nearest neighbours on localization error.	70
5.7	Cumulative density function of the localization error in a supermarket area of approximately $100m^2$ with 7 AP.	71
5.8	LAURA: Cumulative density function of the localization error in a test area of approximately $100m^2$	72
5.9	Cumulative density function of the tracking error in a supermarket area of approximately $100m^2$ with 7 AP.	73
5.10	Schema of our dynamic localization algorithm: Online Stage . . .	74
5.11	Cumulative density function of the tracking error in a supermarket area of approximately $100m^2$ with 7 AP.	76
5.12	Cumulative density function of the tracking error for different values of the anchor node density in a test area of about $250m^2$	77
6.1	Use Case: Administrator	85
6.2	Use Case: Client Web Site	86
6.3	Use Case: Client into the supermarket	87
6.4	Client Sequence Diagram	89
6.5	Client Activity Diagram	89
6.6	Supermarket Map	92
6.7	Design of Database Map	93
6.8	Design of Server Service	94
6.9	Functionalities of Server Services	96
6.10	Example of the tracking of a customer	98
6.11	The Android device collects RSSI readings and sends them to the server	99
6.12	Synchronization of the shopping list on the Android device	100
6.13	The products located in the same cell as the customer are shown on the Android device	101

List of Tables

2.1	Comparison of localization technologies	13
-----	---	----

Chapter 1

Introduction

There is currently a growing demand for localization services, ranging from *context aware computing* [1] to fleet management and military applications, and even to emerging fields of patients monitoring in medical institutes [22]. Mobile computing has evolved over the last several years and has aimed at providing mobile users an anytime and anywhere access to the right information at the right time, thus enabling value-added services. Knowing the location of users is an important component of mobile computing, in order to assist people with their desired goals, and making either the smart workplace or home really as such. The knowledge of the position of users combined with their profiles could significantly help in network planning, in load balancing, in caching of information closer to the user, radio resource management and designing other performance enhancement methods. Nowadays Position location services are also receiving increased importance due to public safety issues.

RSSI-based indoor localization systems are grown in importance among other indoor positioning techniques, mainly because of the availability of signal strength informations on all wireless equipments at no extra cost. Chosing WiFi 802.11 technology brings the advantage of avoiding the deployment of an *ad hoc* infrastructure just for localization, since it is possible to leverage the same equipments used for providing network access to the tracked users.

Through this thesis we are going to provide an introduction to the Wifi IEEE

802.11 technology, and to the problems that arise in RSSI localization techniques in noisy environments, mainly due to scattering and fading effects. We'll describe classical location estimation algorithms, both in case of static or moving targets, focusing also on most popular signal propagation simulators.

In particular we aim at finding a solution to indoor wifi localization for the purpose of tracking customers moving around in malls or supermarkets, in order to provide additional services, enhance profiling techniques and improve marketing effectiveness.

Developing such a system, which enables the combination of real-time information with a pre-existing profile of the customers, involves taking into account parameters as *user comfort* and *acceptance*, besides accuracy, scalability and installation efforts and costs. WiFi 802.11 turned out to be most suitable option to our purpose, although many issues arisen in the project of our infrastructure. Among these, major issues are due to the strong fluctuations of the RSSI values typical of WiFi even in line of sight conditions, the impossibility of using inter-AccessPoint measure in order to estimate a propagation model, and the different gains of eterogeneous types of antennas.

We will propose our algorithm for indoor wifi localization that combines the benefits of simulating empirical propagation models in terms of lowering the off-line stage effort with the unmatched precision of fingerprinting techniques, merging the two approaches together.

It is designed to suit fast-changing signal propagation conditions and to satisfy all requirements and constraints derived from noisy environments and our particular field of adoption explained through this thesis.

Moreover, in dynamic localization, we have shown how it is possible to improve it further with the adoption of a *Particle Filter* in order to incorporate the knowledge of cinematic models of the moving target in our localization algorithm.

We will also provide a comparative analysis of our method with respect to other approaches found in literature, presenting the obtained results both in static and dynamic localization efforts.

We designed an application infrastructure called WhAC (*Where Are my Customers*)

that exploits our localization algorithm in order to provide additional service based on the location of customers into the store. However, the application part is independent from the chosen localization algorithm. This means that even changing the algorithms used for localization, the software infrastructure will still work.

After giving a description of the problem, we will identify the actors of the system and all possible scenarios through the use of UML diagrams. Then we will present the design of our database used to support the application layer, and we will describe the concept of the *map of the environment* that will reside on the server. Finally, we are going to describe the client-side application and how it interacts with a service that resides server-side and uses the location informations.

In particular we'll see how the WhAC application can improve an everyday activity such as shopping.

Organization of the thesis

This thesis is divided as follows: Chapter 2 provides an introduction to the basic concepts and techniques involved in *localization*. Chapter 3 introduces the state of the art and related works about localization based on *Received Signal Strength Indication* (RSSI). In Chapter 4 we'll see in detail our application scenario: a mall, focusing on the description of the issues and limits arisen in performing WiFi localization in such context. Our proposed approach will be described in Chapter 5, yielding the description of our localization algorithm with empirical results followed by their analysis and comparison with other solutions found in literature. Finally, in Chapter 6 we'll describe our WhAC software architecture and how it uses localization in order to provide new services to the mall customers and enhance user profiling techniques. At the end, the conclusions and the possible future works will be described.

Chapter 2

Localization systems and techniques

Mobile computing has evolved over the last several years and has aimed at providing mobile users an anytime and anywhere access to the right information at the right time. Knowing the location of users is an important component of mobile computing, in order to assist people with their desired goals, and making either the smart workplace or home really as such. The knowledge of the position of users combined with the user profiles could significantly help in network planning, in load balancing, caching of information closer to the user, radio resource management and designing other performance enhancement methods. Nowadays Position location services are also receiving increased importance due to public safety issues. In this chapter we will span a comprehensive view of all main techniques for enabling localization services.

2.1 Localization

Localization process involves several issues and characteristics that need to be defined before we can start discussing the several approaches. Each approach may be suitable in one application over the other. According to the following comparison criteria, a choice regarding which location sensing system could be chosen would be made.

- **Physical Position and Symbolic Location:** Physical position is the posi-

tion obtained from the system using physical coordinates such as those obtained from a GPS device. The GPS device usually provides the user with coordinates in the form of lateral and longitudinal degrees as well as altitude with respect to the sea. On the other hand, a symbolic location is a location that has a meaning only in its specific context. For example, a symbolic location system could inform users that they are in the living room instead of giving them coordinates. Usually, a physically positioning system can provide the coordinates to another interface that would query a database of symbolic locations in order to emulate a symbolic location system.

- **Absolute vs Relative:** In an absolute positioning system, if two mobile devices are located at the same spot, their position will be the same (for example, in a GPS system). On the other hand, if a relative positioning system is used, the device will report its location with respect to a specific reference point or in case of locating other devices, with respect to itself. As the case with physical position and symbolic location, either systems; absolute or relative, can be augmented to simulate the other, though with limited success. In case of an absolute positioning system, the system could emulate relative positioning by searching for a reference point and modifying the reported location to be with respect to that reference point. In case of relative positioning, one could use the knowledge of the position of other reference points in order to determine the absolute location.
- **Location Computation:** The location computation parameters defines whether the client is the one that calculates the location using a specific algorithm or that it is a centralized server that manages this calculation. This choice would depend upon the size as well as the designated power consumption of the client. If the client computes its own location (for example using triangulation from GPS signals), it would need to have sufficient computational power in order to calculate its position in adequate time. This computational power may translate in more electronic power consumption. On the other hand, a device that relies on an external server for the localization proce-

ture (by receiving the locations directly from the server) would not need much computational power but, on the other hand, this solution entails that the server would have to undertake more responsibilities which might affect the number of clients that could be served simultaneously. On another note, having the server compute the client locations and possibly storing them, raises privacy concerns.

- **Accuracy and Precision:** Accuracy represents how much is the estimated location deviated from the actual position. Precision is the percentage of time the system is able to determine the location and achieve the specified accuracy. Both accuracy and precision are important when determining the effectiveness of a specific positioning system. Depending on the intended application, a 50m accurate system might be sufficient, for example in an outdoor city tour guide. In contrast, the same system might not be effective at all if the intended application was indoor positioning since this 50m window of error is simply too large that it might span the whole building. New research is directed towards the use of information from several sources by aggregating them into a statistical model in effort to increase both the accuracy and precision of the system.
- **Responsiveness and Latency:** For the application to be productive, it has to be responsive which means that the location of the user would have to be calculated in real-time. This can sometimes pose a challenge because if the user device relies on the server for location calculation, there might be some latency when obtaining the information due to the effects of network propagation delays. The other option is letting the device calculate its position, but this choice can also be challenging since the device might not have enough processing power in order to calculate the information in real time.

2.2 Applications of localization

Several interesting location-aware applications are emerging in everyday life, for example:

- **Fleet management** for having always under control the exact position of the company's cars, trucks and airplanes.
- **Indoor asset tracking and monitoring** of objects such as projectors for ease of finding as well as theft protection.
- **People tracking** inside a facility, like doctors in a hospital to quickly route the nearest doctor to emergencies. Similarly is possible to track persons in an open environment, i.e. soldiers on a battlefield.
- **Shopping assistance** in shopping malls by giving the users the ability to find the locations of specific items in stores as well as displaying information about those items such as the price and where is the cheapest place to get them.
- **Museum guide systems** that inform the visitors of a museum with the location of the various attractions as well as providing specific information about those attractions once the user is near them.

The following section presents one of the possible classifications of the different techniques that can be used in localization as well as their implementations. In general, localization techniques can be broadly classified into two main categories: outdoor and indoor.

2.3 Outdoor Localization

Outdoor localization involves locating users in a relatively open environment with no or few obstacles. The applications of outdoor localization include mapping services as well as providing location sensitive information where the exact location

of the user is not necessarily known. Current technologies for outdoor localization can be classified into 3 main classes:

- Handset Based
- Cellular Network Based
- Hybrid Approaches

2.3.1 Global Positioning System

The Global Positioning System (GPS) is a space-based global navigation satellite system (GNSS) that provides location and time information anywhere on or near the Earth, provided that there is an unobstructed line of sight to four or more GPS satellites. A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites high above the Earth. Each satellite continually transmits messages that include:

- The time the message was transmitted
- Precise orbital information (the *ephemeris*)
- The general system health and rough orbits of all GPS satellites (the *almanac*)

The receiver uses the messages received to determine the transit time of each message and computes the distance to each satellite. These distances along with the satellites' locations are used with the possible aid of trilateration, depending on which algorithm is used, to compute the position of the receiver. This position is then displayed, perhaps with a moving map display or latitude and longitude; elevation information may be included. Many GPS handset show derived information such as direction and speed, calculated from position changes.

Three satellites might seem enough to resolve a position since *space* has three dimensions and we can assume a position near the Earth's surface. However, even a very small clock error multiplied by the very large speed of light results in a

large positional error. Therefore receivers use four or more satellites to solve for the receiver's location and time. The very accurately computed time is effectively hidden by most GPS applications, which use only the location. A few specialized GPS applications do however use the time; these include time transfer, traffic signal timing, and synchronization of cell phone base stations.

Application of GPS: Originally conceived as a military project, GPS is considered a dual-use technology, meaning it has both military and civilian applications. GPS has become a widely deployed and useful tool for commerce, scientific uses, tracking, and surveillance. GPS's accurate time facilitates everyday activities such as banking, mobile phone operations, and even the control of power grids by allowing a synchronized hand-off switching.

2.3.2 Cellular Networks

Cellular networks such as GSM (Global Standard for Mobile Communications) and UMTS (Universal Mobile Telecommunication System) are network systems used to provide mobile telecommunication services. Locating a user on one of these networks can be done differently as follows. The first method is the cell-of-origin method, the coverage area is divided into cells. At any given time, the mobile user, using the cellular network, has to register with the cell he is currently in, which enables the system to locate him at any given time. The radius of the range that a cell could cover is in the order of a few kilometers. Positioning accuracy of up to 1km can be achieved using GSM. The other two methods that can be used with cellular networks are somehow similar to the methods used indoors. The first one is based on a trilateration time-of-flight approach as described before. The second method is based on the *fingerprinting* approach where the signal characteristics at several areas are measured and stored in a database. Both of these methods have a better accuracy than the cell-of-origin method averaging between 150m to 300m

2.4 Indoor Localization

Systems designed for indoor localization can be classified based on the signal types (infrared, ultrasound, ultra wideband, and radio frequency), signal metrics (AOA: angle of arrival, TOA: time of arrival, TDOA: time difference of arrival, and RSS: received signal strength), the metric processing methods (algorithms and scene profiling), and inertial navigation systems.

2.4.1 Signal Types

Infrared

Infrared (IR) light is electromagnetic radiation with a wavelength longer than that of visible light, measured from the nominal edge of visible red light at 0.7 micrometres, and extending conventionally to 300 micrometres. These wavelengths correspond to a frequency range of approximately 430MHz to 1 THz and include most of the thermal radiation emitted by objects near room temperature. The limited range of an IR network, which facilitates user location, is a handicap in providing ubiquitous coverage. Moreover, the IR network is often deployed for the sole purpose of locating people and does not provide traditional data networking services [2].

Ultrasound

Ultrasound is cyclic sound pressure with a frequency greater than the upper limit of human hearing. Although this limit varies from person to person, it is approximately 20,000 Hertz in healthy, young adults. The production of ultrasound is used in many different fields, typically to penetrate a medium and measure the reflection signature or supply focused energy. The reflection signature can reveal details about the inner structure of the medium, a property also used by animals such as bats for hunting. The most well known application of ultrasound is its use in sonography to produce pictures of fetuses in the human womb. There are a vast number of other applications as well. Location systems using ultrasound

are always hybrid. They use an RF signal which propagates 106 times faster than an ultrasonic one for the propagation time measurement of the ultrasonic signals. Due to the slow propagation speed of sound, the required sampling frequencies are much lower (100 KHz) and therefore, compared to RF, cheaper hardware can be used. However, the range of ultrasonic transducers is very limited (5m) resulting in a very high installation effort. Another disadvantage is that the accuracy drops perceptibly for moving objects. In addition, these systems also demand to carry hardware and furthermore, ultrasound can be bothersome for pets [2].

Ultra-Wideband

Ultra-wideband (also known as UWB, ultra-wide band, ultraband) is a radio technology that can be used at very low energy levels for short-range high-bandwidth communications by using a large portion of the radio spectrum. UWB has traditional applications in non-cooperative radar imaging. Most recent applications target sensor data collection, precision locating and tracking applications.

UWB communications transmit in a way that doesn't interfere largely with other more traditional narrowband and continuous carrier wave uses in the same frequency band. However first studies show that the rise of noise level by a number of UWB transmitters puts a burden on existing communications services. This may be hard to bear for traditional systems designs and may affect the stability of such existing systems. Since UWB radars have excellent spatial resolution it is supposed that can be advantageously applied in the field of localization and navigation. There are number of applications that would take advantage from the precise indoor positioning and navigation such as automatic storage, tacking of various targets, or people in dangerous environments and so on. Due to the shortness of the measurement pulses, the directly received signal can be distinguished more easily from reflected signals. However, due to the high signal propagation speed and the required high sampling frequencies, costly hardware is necessary. Usually an UWB-based system achieves an accuracy of about 15cm [3].

Radio Frequency

Radio frequency (RF) is a rate of oscillation in the range of about 3 kHz to 300 GHz, which corresponds to the frequency of radio waves, and the alternating currents which carry radio signals. RF usually refers to electrical rather than mechanical oscillations, although mechanical RF systems do exist.

There are several realizations using RF signals for localization including WLAN and Bluetooth technologies. The advantage of Wireless LAN and Bluetooth is, that many cellualars and PDAs are already equipped with adequate hardware and that the required infrastructure is also already installed in many places. Unfortunately, the accuracy of all of these systems is more or less detracted by multipaths propagation.

As a summary, table 2.4.1 illustrates the mentioned advantages and disadvantages of the discussed technologies.

System	Comfort	Acceptance	Accuracy	Scalability	Installation	Cost
Infrared	○	○	-	○	○	+
Ultrasound	-	-	○	○	-	○
Ultrawideband	○	-	+	○	○	-
Radio Frequency	○	-	-	+	○	-

Table 2.1: Comparison of localization technologies

2.4.2 Signal Metrics

TOA: Time Of Arrival

Time of Arrival (TOA or ToA), also named Time of flight (ToF), refers to the travel time of a radio signal from a single transmitter to a remote single receiver. The goal is to measure the time of arrival of the acoustic paths that link the source to the destination (i.e. a microphone). In order to do so, the loud-speaker produces

a known time-continuous sequence $s(t)$. The time-continuous signal $x(t)$ acquired by sensor i synchronized with source is the delayed replica of the signal $s(t)$:

$$x_i(t) = hs(t - \tau_i) + v_i(t) \quad (2.1)$$

The coefficient h is the attenuation of the direct path from source to receiver; τ is the corresponding delay; finally $v(t)$ is an additive noise that alters our measurement. After sampling by A/D converter we write the time-discrete signal as:

$$x_i(k) = x_i(kT) = hs(k - i) + v_i(k) \quad (2.2)$$

where T is the sampling period; k is the time-discrete index; i is the discrete version of τ . For the sake of simplicity TOAs are estimated by picking peaks in the cross-correlations of $x(t)$ and $s(k)$. In particular the lag i is the location of the first relevant peak in the cross-correlation.

TDOA: Time Difference Of Arrival

Differently from ToA, when measuring TDOAs we remove the hypothesis that sources are synchronized with receivers, while synchronism among receiver microphones still holds. This scenario accounts for situations in which arbitrary source (e.g. a person uttering a sentence, a ring of mobile phone, etc.) is active in the environment. Obviously, this is a more interesting case because it is more realistic to have a source not synchronized with sensors. Even if we cannot measure the time of arrival, many cues about the source position can still be extracted from the joint knowledge of the synchronized signals available at sensors. After sampling, the signal acquired by sensor i in the array is

$$x_i(k) = hs(k - i) + v_i(k) \quad (2.3)$$

The model in the above equation is equivalent to the model in equation 2.3. Under the assumption that the attenuation h does not depend on the microphone

index and that additive noises at different microphones are uncorrelated, the cross-correlation of $x(k)$ and $x(k)$ gives:

$$r_{ij}(k) = h^2 s(k - i_i) \otimes s(k - i_j) \quad (2.4)$$

where $\otimes$ denotes the cross-correlation operator. If the source signal is a white noise, we obtain:

$$r_{ij}(k) = h^2 \delta[k - (i_i - i_j)] \quad (2.5)$$

which exhibits a global maximum for $k = i_i - i_j$. For this reason we can extract TDOAs from cross-correlations searching for the global maximum locations [4].

RSSI: Received Signal Strength Indication

RSSI is the relative received signal strength in a wireless environment, usually expressed in milliWatt (mW) or in decibel (dBm). In particular, RSSI is an indication of the power level of a radio signal being received by the antenna. Therefore, the higher is the RSSI number, the stronger is the signal. RSSI can be used internally in a wireless networking card to determine when the amount of radio energy in the channel is below a certain threshold at which point the network card is allowed to transit (*Clear to Send* - CTS). Once the card is clear to send, a packet of information can be sent. The end-user will likely observe a RSSI value when measuring the signal strength of a wireless network through the use of a consumer wireless adapter.

2.4.3 Metric Processing Methods

Scene Analysis and Fingerprinting

The *scene analysis technique* depends on analyzing the location to obtain features that are easily compared and represented. A *scene* can be described either by images for the location or by the signal strength at that location. In static scene analysis, the observed features are looked up in the database that maps objects to

locations. In *differential scene analysis*, the difference between successive scenes is tracked. Those differences will imply that the user has moved. If features are known to be at specific locations, the observer can compute its location relative to them. The advantage of such system is that it can act independently from an external server by storing the database locally, which means that less power is required for data transmission and the privacy of the user could be easily maintained. The disadvantages include the need for a pre-built lookup database of the features that might need to be rebuilt if the features of the scene change.

2.4.4 Inertial Navigation Systems

An *inertial navigation system* (INS) is a navigation aid that uses a computer, motion sensors (accelerometers) and rotation sensors (gyroscopes) to continuously calculate via *dead reckoning* the position, orientation, and velocity (direction and speed) of a moving object without the need for external references. It is used on vehicles such as ships, aircraft, submarines, guided missiles, and spacecraft. Other terms used to refer to inertial navigation systems or closely related devices include inertial guidance system, inertial reference platform, inertial instrument, inertial measurement unit (IMU) and many other variations.

inserire immagine Accelerometer suggerita da Roberto

Accelerometer

An accelerometer is a device that measures the proper acceleration of the device. This is not necessarily the same as the coordinate acceleration (change of velocity of the device in space), but is rather the type of acceleration associated with the phenomenon of weight experienced by a test mass that resides in the frame of reference of the accelerometer device [6]. A system that do localization using accelerometer works in this way:

- The first step initializes the coordinates of the tracking object in a three-dimensional space. Normally, when the objects in a static state, it's X axis and Y axis acceleration value is 0, while there is gravity on the Z axe.

- The second step gathers accelerometer information, i.e. values of the three axis at a fixed frequency.
- The third step is to calculate location informations through looping the following method: the microprocessor transmits the three axis acceleration values, and then, according to the acceleration values, it infers the displacements by calculation using Newton theory.

Dead Reckoning

Dead reckoning (DR) allows a navigator to determine his present position by projecting his past courses steered and speeds over ground from a known past position. *Dead reckoning* can also try to determine a future position by projecting an ordered course and speed of advance from a known present position. Obviously, the position estimated via DR is only an approximate position, because it does not allow for the effect of leeway, current, helmsman error, or gyroscope error. *Dead reckoning* helps in predicting landfall, predicting arrival times and in evaluating the accuracy of electronic positioning information. It also helps in predicting which celestial bodies will be available for future observation. The navigator should carefully tend his DR plot, update it when required, use it to evaluate external forces acting on his vehicle, and consult it to avoid potential navigation hazards [7].

SurroundSense

SurroundSense is a localization technique based on the hypothesis that the combined effect of ambient sound, light, and color (i.e., the overall ambience of a place) can be unique enough to be considered a particular type of *fingerprint* and thus exploited for localization.

The device that gathers data for the localization procedure (i.e. a phone), continuously senses the ambience automatically. The sensed values are recorded, pre-processed, and transmitted to a remote SurroundSense server. Once the sensor values arrive at the server, they are forwarded to the fingerprinting factory, that

classifies the type of sensor data (sound, color, light, WiFi, accelerometer) and distributes them to respective fingerprinting modules. These modules perform a set of appropriate operations, including color clustering, light extraction and feature selection. The individual fingerprints from each module are logically inserted into a common data structure, called the ambience fingerprint. The matching/filtering module is expected to select one candidate fingerprint that best matches the test fingerprint [8].

Radio Frequency IDentification (RFID)

The Radio Frequency IDentification (RFID) is a technology that allows the automatic identification of objects, animals and persons. It is based on the possibility to store and access datas from a distance using particular devices called *TAGS* that are able to send their stored datas when polled by a RFID-reader device.

In detail, a RFID system is composed by a reader and at least one TAG, which can be either *active* or *passive*. The reader has at least one antenna to send the reading signal and receive responses. If the Tag is passive, it is simply a microchip with no power supply: it will be activated by the radio signal of the reader that must be strong enough to provide sufficient energy to send the answer by law of induction. active Tags are usually more expensive and have also a power supply unit, i.e. a battery, to grant the possibility of more complex computations and stronger signal.

Reference Tags are the Active Tags whose locations are fixed to help location calibration, and they serve as reference points in the system. This approach has three major advantages. First, there is no need for a large number of expensive RFID readers, and use extra, cheaper RFID tags instead, thus can be more practical due to cost constraint. Second, the environmental changes can easily be accommodated. The signal strength and detecting range of a same tag can be various in a complicated indoor environment, so it is hard to estimate the distance between Reader and Tag only by the detected signal strength. But for the reference tags are subject to the same effect in the environment as the tags to be located, this approach helps offset many environmental factors that contribute to the variations

in detected range. Third, the location information is more accurate and reliable.

The main focus of this system is to find out the tracking tags nearest neighbours of Reference Tags in the signal space by comparing the signal strength of tracking tags and reference tags detected by reader. Selected reference tags whose signal strength is close to the tracking tags should be granted higher weighting value, and then obtain the object location by computing the weighting value and the reference tags real positions.

Chapter summary

In this chapter we have seen how localization of people have grown in importance in many sectors of everyday life, becoming more and more pervasive and enabling value-added services. Besides this, we have provided a bird-eye view on a comprehensive variety of enabling techniques, ranging from exploitation of radio-waves to inertial navigation systems.

Chapter 3

Localization based on Received Signal Strength Indication

RSSI-based indoor localization systems are growing in importance among other indoor positioning techniques, mainly because of the availability of signal strength informations on all wireless equipments. Moreover, WLAN networks do not require to deploy an *ad hoc* infrastructure but it is possible to leverage the same equipments used for providing network access to the tracked users.

Through this chapter we are going to provide an introduction to the Wifi IEEE 802.11 technology, and to the problems that arise in RSSI localization techniques in noisy environments, mainly due due to scattering and fading effects.

Also, we'll provide a description of classical location estimation algorithms and a bird's-eye view on most popular signal propagation simulators.

3.1 Wireless Channel and RSSI

3.1.1 IEEE 802.11

IEEE 802.11 is a set of standards for implementing wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands. They are created and maintained by the IEEE LAN/MAN Standards Committee

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

(IEEE 802) [9].

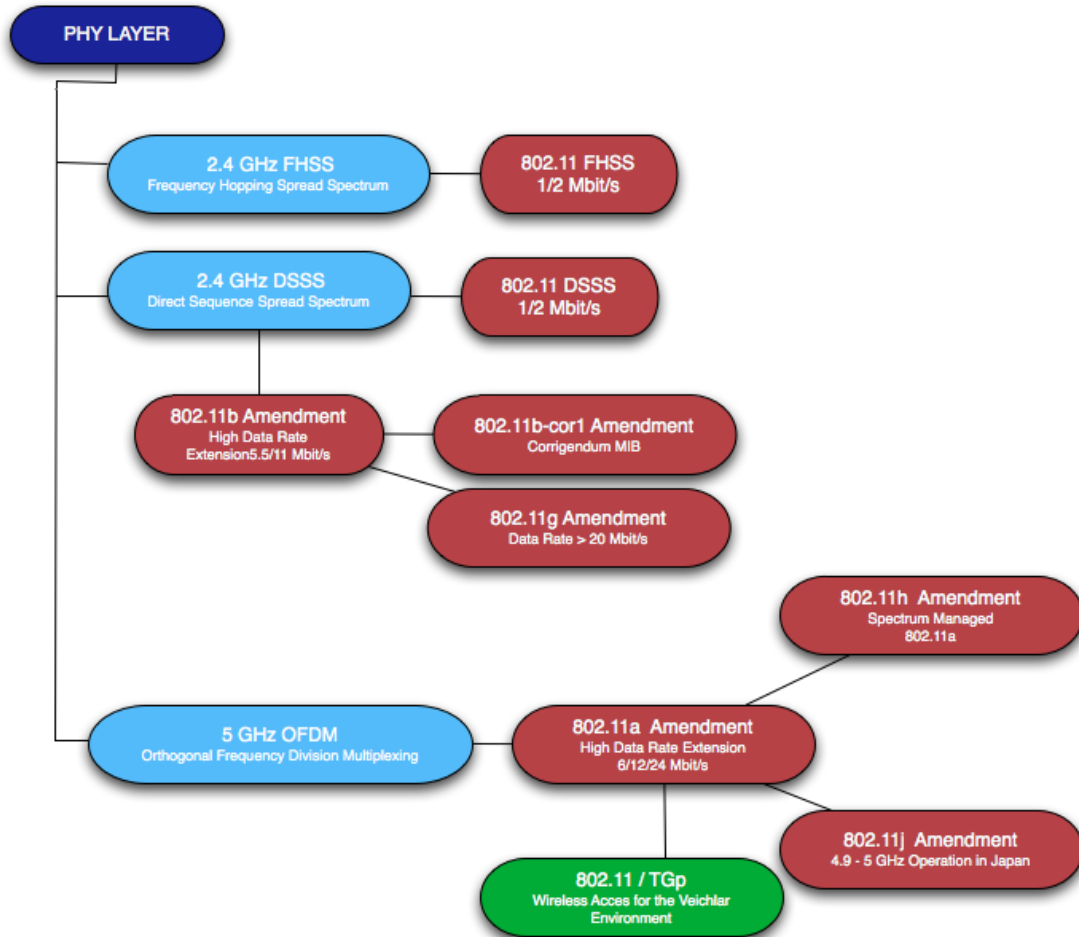


Figure 3.1: Schema of the IEEE 802.11 standard

- **802.11a:** The 802.11a standard uses the same data link layer protocol and frame format as the original standard, but an OFDM based air interface (physical layer). It operates in the 5 GHz band with a maximum net data rate of 54 Mbit/s, plus error correction code, which yields realistic net achievable throughput in the mid-20 Mbit/s. Since the 2.4 GHz band is heavily used to the point of being crowded, using the relatively unused 5 GHz band gives 802.11a a significant advantage. However, this high carrier

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

frequency also brings a disadvantage: the effective overall range of 802.11a is less than that of 802.11b/g. In theory, 802.11a signals are absorbed more readily by walls and other solid objects in their path due to their smaller wavelength and, as a result, cannot penetrate as far as those of 802.11b. In practice, 802.11b typically has a higher range at low speeds (802.11b will reduce speed to 5 Mbit/s or even 1 Mbit/s at low signal strengths). However, at higher speeds, 802.11a often has the same or greater range due to less interference.

- **802.11b:** 802.11b has a maximum raw data rate of 11 Mbit/s and uses the same media access method defined in the original standard. 802.11b products appeared on the market in early 2000, since 802.11b is a direct extension of the modulation technique defined in the original standard. The dramatic increase in throughput of 802.11b (compared to the original standard) along with simultaneous substantial price reductions led to the rapid acceptance of 802.11b as the definitive wireless LAN technology. 802.11b devices suffer interference from other products operating in the 2.4 GHz band. Devices operating in the 2.4 GHz range include: microwave ovens, Bluetooth devices, baby monitors and cordless telephones.
- **802.11g:** In June 2003, a third modulation standard was ratified: 802.11g. This works in the 2.4 GHz band (like 802.11b), but uses the same OFDM based transmission scheme as 802.11a. It operates at a maximum physical layer bit rate of 54 Mbit/s exclusive of forward error correction codes, or about 22 Mbit/s average throughput. 802.11g hardware is fully backwards compatible with 802.11b hardware and therefore is encumbered with legacy issues that reduce throughput when compared to 802.11a by 21. The then-proposed 802.11g standard was rapidly adopted by consumers starting in January 2003, well before ratification, due to the desire for higher data rates as well as to reductions in manufacturing costs. By summer 2003, most dual-band 802.11a/b products became dual-band/trimode, supporting a and b/g in a single mobile adapter card or access point. Details of making b and

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

g work well together occupied much of the lingering technical process; in an 802.11g network, however, activity of an 802.11b participant will reduce the data rate of the overall 802.11g network. Like 802.11b, 802.11g devices suffer interference from other products operating in the 2.4 GHz band, for example wireless keyboards.

- **802.11n:** 802.11n is an amendment which improves upon the previous 802.11 standards by adding multiple-input multiple-output antennas (MIMO). 802.11n operates on both the 2.4GHz and the lesser used 5 GHz bands. The IEEE has approved the amendment and it was published in October 2009. Prior to the final ratification, enterprises were already migrating to 802.11n networks based on the Wi-Fi Alliance's certification of products conforming to a 2007 draft of the 802.11n proposal.

3.1.2 RSSI: Received signal strength indication

Received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal. RSSI is generic radio receiver technology metric, which is usually invisible to the user of the device containing the receiver, but is directly known to users of wireless networking of IEEE 802.11 protocol family. In an IEEE 802.11 system RSSI is the relative received signal strength in a wireless environment, in arbitrary units. RSSI is an indication of the power level being received by the antenna. Therefore, the higher the RSSI number the stronger the signal.

3.1.3 Problems of Wireless Channel

Wireless signals propagating in air are subject to phenomena:

- Distance attenuation
- Fading Multipath
- Shadowing

Distance attenuation

When a signal travels from a source to a receiver is subject to many factors of attenuation. The first of these is distance attenuation. Usually the source is an antenna that focuses the power radiated in a direction. The power density of an antenna is expressed by:

$$F = \frac{P_t g_t}{4\pi d^2} [W/m^2]$$

where P_t is the power transmission, g_t is the maximum gain. Then the power receiver is expressed by:

$$P_r = P_T g_T g_R \left(\frac{\lambda}{4\pi d}\right)^2$$

where P_t is the power radiated from the transmitter, and $g_r g_T$ gains of transmitter and receiver, λ the wavelength and the distance between the transmitter and receiver. Attenuation free space is represented by:

$$L = \left(\frac{\lambda}{4\pi d}\right)^{-2}$$

This attenuation is not the only affected but also other signal attenuation may be present because of the atmosphere (depending on the frequency and fog, rain ..) and obstacles (absorption, reflection, diffraction ..). In the simple case of propagation with two rays, one direct and one fully reflect the relationship between power received and transmitted power takes the form:

$$P_R = P_T g_R g_T \left(\frac{\lambda}{d^2}\right)^2$$

In the scenario of two-ray propagation by the received power decreases due to attenuation due to distance, much faster ($\sim 1/d^4$) in the case of propagation in free space ($\sim 1/d^2$). usually uses a similar formula but also in the general case where the exponent of the distance (propagation coefficient η) can take values between 2 (free space) and 5 (strong attenuation urban):

$$P_R = P_T g_T g_R \left(\frac{\lambda}{4\pi}\right)^2 \frac{1}{d^\eta}$$

Fading Multipath

During the propagation between source and destination, the signal can follow multiple paths due to total or partial reflection from obstacles. The behavior of the waves on the objects depends on the frequency of the signal and the characteristics and dimensions of objects. In general, low frequency waves can pass through many obstacles nearly without attenuation, while, increasing frequency, signals tend to be absorbed or reflected (at very high frequencies - more than 5 GHz - it is possible only the direct propagation).

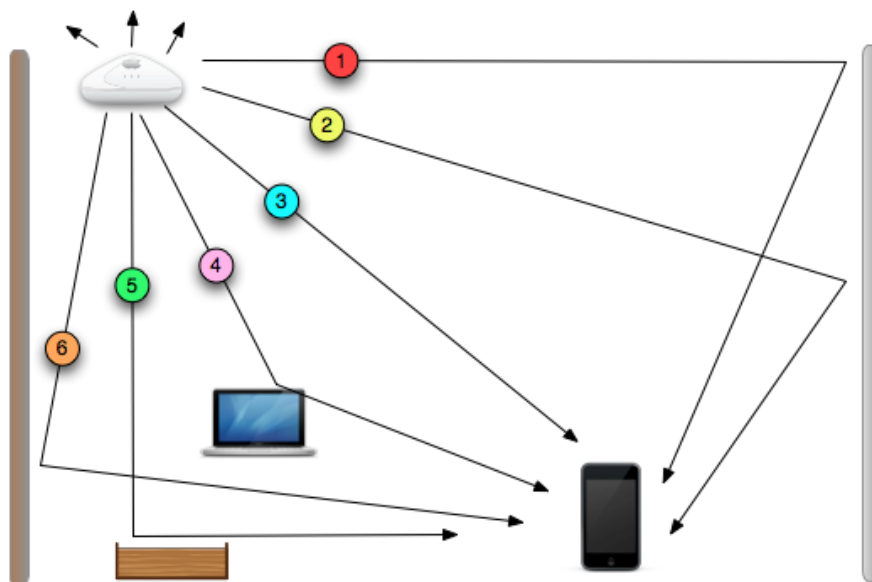


Figure 3.2: Signal recombination due to the multipath propagation

Signal replicas coming from different paths are recombined at the receiver. The result of recombination depends on:

- Replicas number
- Relative phases
- Amplitudes
- Frequency

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

In this way the resulting signal can be attenuated or amplified. If the terminal is moving the features of the recombination time varying. The fading holes tend to be spaced over the time taken to travel half the wavelength.

In reality, the multipath propagation can cause other more complex problems in the case of digital transmission. In this case, the different replicas of the transmitted signal delays (delay spread) cause a widening of the channel impulse response which can lead to intersymbol interference (ISI - Inter-Symbol Interference). The significance of the delay spread can be quantified by calculating its mean-square (RMS Delay Spread):

$$\tau_{RMS} = \sqrt{\frac{1}{\sum_{i=1}^N P_i} \sum_{i=1}^N (\tau_i^2 P_i) - \tau_d^2}$$

where

$$\tau_d = \frac{\sum_{i=1}^N (\tau_i P_i)}{\sum_{i=1}^N P_i}$$

τ_{RMS} RMS delay spread, τ_i delay path i , P_i received power path i , n number of paths.

The inverse of the delay spread of the band provides consistency. If the bandwidth is greater than the coherence bandwidth of the signal delay spread is not a problem. If, however, the coherence bandwidth is comparable with that of the signal delay spread causes intersymbol interference and non-negligible errors in reception. In this case, to compensate for the distortion in frequency channel must be equalized with a filter should be adaptable in receiving. In some scenarios you can write a simplified statistical representation of the attenuation due to fading from multipath. Two example are Rayleigh fading and Rice fading:

- **Rayleigh fading:** Rayleigh fading models assume that the magnitude of a signal that has passed through such a transmission medium (also called a communications channel) will vary randomly, or fade, according to a Rayleigh distribution the radial component of the sum of two uncorrelated Gaussian random variables. Rayleigh fading is viewed as a reasonable model for tropospheric and ionospheric signal propagation as well as

the effect of heavily built-up urban environments on radio signals. Rayleigh fading is most applicable when there is no dominant propagation along a line of sight between the transmitter and receiver. If there is a dominant line of sight, Rician fading may be more applicable.

- **Ricean fading:** Ricean fading is a stochastic model for radio propagation anomaly caused by partial cancellation of a radio signal by itself the signal arrives at the receiver by several different paths (hence exhibiting multipath interference), and at least one of the paths is changing (lengthening or shortening). Rician fading occurs when one of the paths, typically a line of sight signal, is much stronger than the others. In Rician fading, the amplitude gain is characterized by a Rician distribution.

Shadowing

In the signal propagation through or is reflected by obstacles that partially absorb the signal. This generates additional attenuation which is usually identified with the name of shadowing. The amplitude change caused by shadowing is often modeled using a log-normal distribution with a standard deviation according to the log-distance path loss model.

3.2 Wireless LAN

A wireless local area network (WLAN) links two or more devices using some wireless distribution method (typically spread-spectrum or OFDM radio), and usually providing a connection through an access point to the wider internet. This gives users the mobility to move around within a local coverage area and still be connected to the network [11].

3.2.1 Architecture of a Wireless LAN

Stations

All components that can connect into a wireless medium in a network are referred to as stations. All stations are equipped with wireless network interface controllers (WNICs). Wireless stations fall into one of two categories: access points, and clients. Access points (APs), normally routers, are base stations for the wireless network. They transmit and receive radio frequencies for wireless enabled devices to communicate with. Wireless clients can be mobile devices such as laptops, personal digital assistants, IP phones and other smartphones, or fixed devices such as desktops and workstations that are equipped with a wireless network interface.

Basic service set

The basic service set (BSS) is a set of all stations that can communicate with each other. There are two types of BSS: Independent BSS (also referred to as IBSS), and infrastructure BSS. Every BSS has an identification (ID) called the BSSID, which is the MAC address of the access point servicing the BSS. An independent BSS (IBSS) is an ad-hoc network that contains no access points, which means they can not connect to any other basic service set. An infrastructure can communicate with other stations not in the same basic service set by communicating through access points.

Extended service set

An extended service set (ESS) is a set of connected BSSs. Access points in an ESS are connected by a distribution system. Each ESS has an ID called the SSID which is a 32-byte (maximum) character string.

Distribution system

A distribution system (DS) connects access points in an extended service set. The concept of a DS can be used to increase network coverage through roaming be-

tween cells.

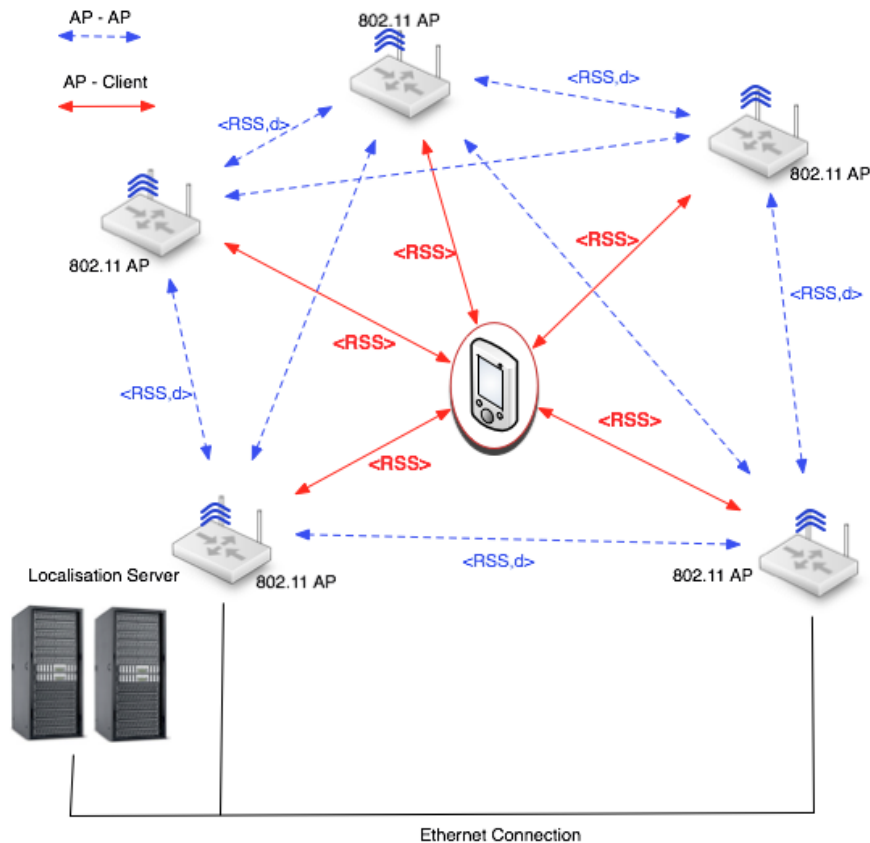


Figure 3.3: Schema of an indoor localization system

3.2.2 Types of wireless LANs

Peer-to-peer

An ad-hoc network is a network where stations communicate only peer to peer (P2P). There is no base and no one gives permission to talk. This is accomplished using the Independent Basic Service Set (IBSS). A peer-to-peer (P2P) network allows wireless devices to directly communicate with each other. Wireless devices within range of each other can discover and communicate directly without involving central access points. This method is typically used by two computers

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

so that they can connect to each other to form a network. If a signal strength meter is used in this situation, it may not read the strength accurately and can be misleading, because it registers the strength of the strongest signal, which may be the closest computer. IEEE 802.11 define the physical layer (PHY) and MAC (Media Access Control) layers based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). The 802.11 specification includes provisions designed to minimize collisions, because two mobile units may both be in range of a common access point, but out of range of each other. The 802.11 has two basic modes of operation: Ad hoc mode enables peer-to-peer transmission between mobile units. Infrastructure mode in which mobile units communicate through an access point that serves as a bridge to a wired network infrastructure is the more common wireless LAN application the one being covered. Since wireless communication uses a more open medium for communication in comparison to wired LANs, the 802.11 designers also included shared-key encryption mechanisms: Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA, WPA2), to secure wireless computer networks.

Bridge

A bridge can be used to connect networks, typically of different types. A wireless Ethernet bridge allows the connection of devices on a wired Ethernet network to a wireless network. The bridge acts as the connection point to the Wireless LAN.

Wireless distribution system

A Wireless Distribution System is a system that enables the wireless interconnection of access points in an IEEE 802.11 network. It allows a wireless network to be expanded using multiple access points without the need for a wired backbone to link them, as is traditionally required. The notable advantage of WDS over other solutions is that it preserves the MAC addresses of client packets across links between access points. An access point can be either a main, relay or remote base station. A main base station is typically connected to the wired Ethernet. A relay base station relays data between remote base stations, wireless clients or other re-

lay stations to either a main or another relay base station. A remote base station accepts connections from wireless clients and passes them to relay or main stations. Connections between "clients" are made using MAC addresses rather than by specifying IP assignments. All base stations in a Wireless Distribution System must be configured to use the same radio channel, and share WEP keys or WPA keys if they are used. They can be configured to different service set identifiers. WDS also requires that every base station be configured to forward to others in the system. WDS may also be referred to as repeater mode because it appears to bridge and accept wireless clients at the same time (unlike traditional bridging). It should be noted, however, that throughput in this method is halved for all clients connected wirelessly. When it is difficult to connect all of the access points in a network by wires, it is also possible to put up access points as repeaters.

3.3 Indoor Localization

3.3.1 Static Scene Analysis: Fingerprinting

Static scene analysis examines certain features such as RF propagation that can characterize various areas covered by a location system. Received signal strength indicator (RSSI) metric, which is measured from captured wireless LAN beacons, is our choice for static scene analysis although use of other metrics is also possible. In the remainder of this paper, we use two terms *RF metric* and *RSSI* interchangeably. Static scene analysis using such RF metrics consists of **offline** and **run-time** stages [12].

Offline Stage

Offline stage: during this stage, measurement of the RF features at known locations is carried out. Wireless access points deployed in the environment periodically transmit beacons. A signal metric measured from detected beacons, such as received signal strength, can be a useful RF feature. Signal metrics with respect to each access point are collected. The collected signal metrics are stored in a

location database to relate the signal information and coordinates of the known locations.

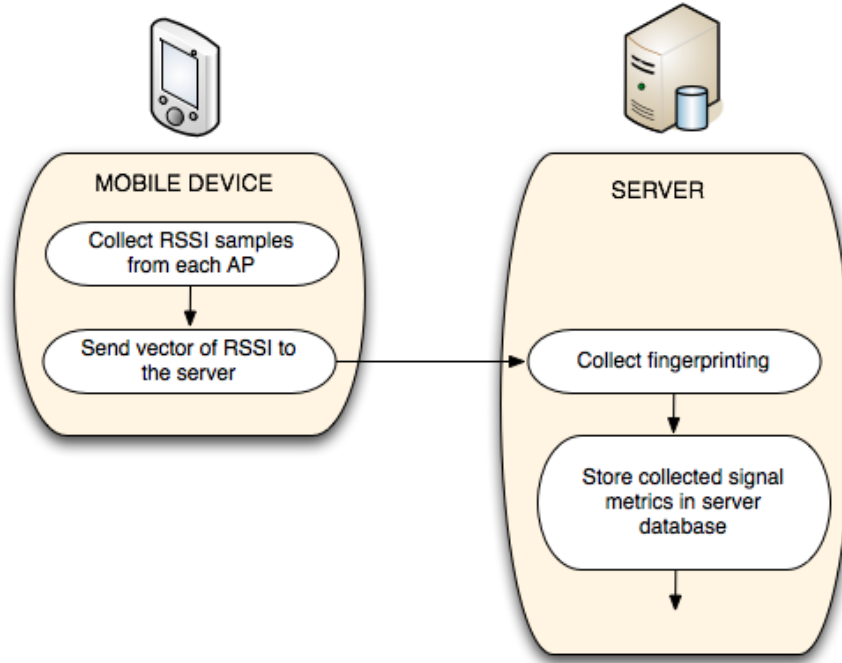


Figure 3.4: Interaction between Server and a Mobile Device during the **offline** stage

Run-time Stage

Run-time stage: measurement of the same signal metric as used in the offline stage is carried out. The location database is accessed to compare the signal metrics collected during the run-time (at an unknown location) with the stored entries. A location estimation algorithm is then applied to infer the location estimate for the unknown location.

3.3.2 Propagation Model

Wireless network design based on software aids, that have built-in propagation models, is much cheaper and easier than *site survey* approach, which is time con-

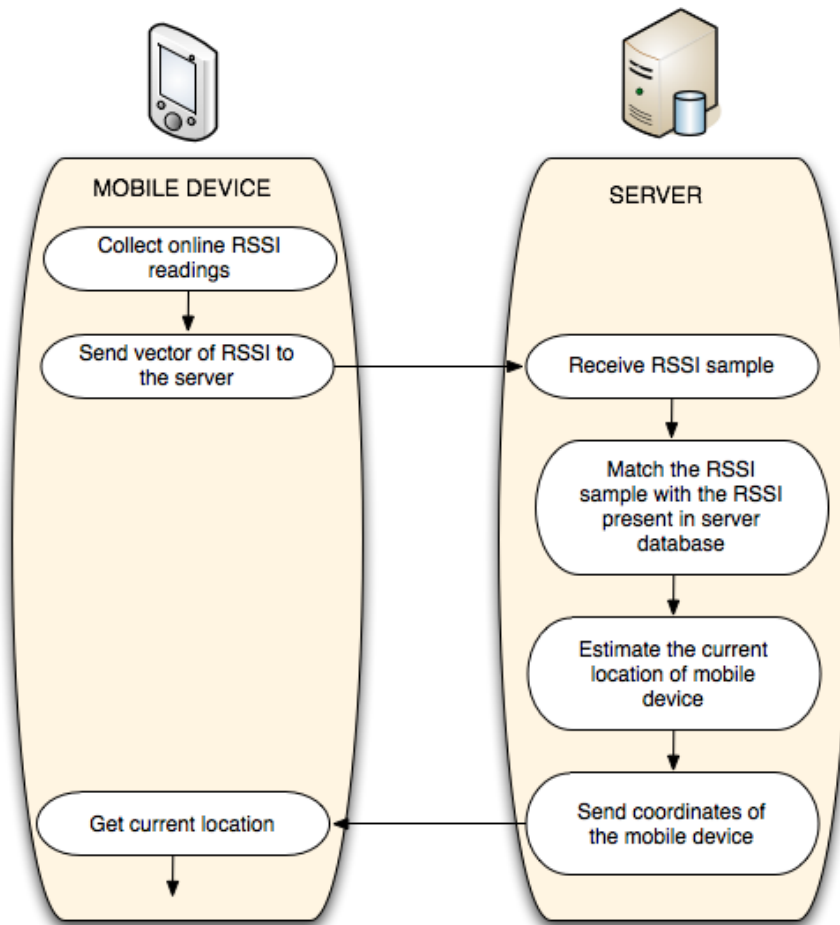


Figure 3.5: Interaction between Server and a Mobile Device during the **online** stage

suming and more expensive [10]. Propagation models could be classified as follows:

1. Empirical

- Free Space Loss
- One Slope Model
- Linear Attenuation Model
- COST 231 HATA

2. Semi-empirical

- Multi Wall Model
- COST WI

3. Optical

- Ray Tracing
- Ray Launching
- Dominant Path Prediction

4. Theoretical

- based on the field theory

Propagation models that have been verified [6] and found to be most prospective during current and future research on wireless network design methods are One Slope Model and Multi Wall Model. Both of them use simple Free Space Loss formula for calculating fundamental loss over the distance in free space.

Free Space Loss: Free Space Loss (FSL) model is derived from the Friis Equation which describes signal loss over the distance between the transmitter and the receiver:

$$P_R = P_T g_T g_R \left(\frac{\lambda}{4\pi d} \right)^2$$

where P_R is power received by the receiving antenna, P_T is power output to the transmitting antenna, g_T and g_R are the antenna gain of the transmitting and receiving antennas, λ is the wavelength and d is the distance between the transmitter and the receiver. Equation transformed and simplified with the assumption that gain of the antennas $G_T = G_R = 0$ dBi has the form:

$$PL(d) = 20 \log \left(\frac{4\pi}{\lambda} \right) + 20 \log d$$

where $PL(d)$ is the path loss over distance d .

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

After further transformations we get formula:

$$L_{FSL}(d) = 32.44 + 20 \log f + 20 \log d$$

where L_{FSL} is free space loss [dB], f is the frequency [MHz] and d is the distance [km].

One Slope Model. One Slope Model. One Slope Model (OSM) is one of the simplest models for calculating of signal strength level without considering structure of the building:

$$L_{OSM}(d) = L_0 + 10n \log(d)$$

Signal strength level L_{OSM} depends mainly on the distance d between the transmitter and the receiver. The environment-dependent values L_0 (signal loss on 1 meter distance) and n (empirical value representing degree of signal loss) are fixed for specified environment. Values of L_0 and n are given in the literature, e.g. for the office environment at frequency $f = 2.45$ GHz variables L_0 and n have values: $L_0 = 40.2$ and $n = 4.2$. OSM is not precise model because it does not incorporate structure of the environment. Sometimes the structure is not known so the model OSM could be useful in such cases.

Multi Wall Model. Multi Wall Model (MWM) uses FSL for basic calculation of the loss over the distance and more sophisticated sum of the losses on the obstacles, mainly walls:

$$L_{MWM}(d) = L_{FSL} + \sum_{i=1}^N k_{Wi} L_{Wi} + k_f L_f$$

where L_{FSL} is signal loss in free space (Eq. 3), d is the distance [m], N is number of types of the walls, k_{Wi} is number of the i -type walls with L_{Wi} loss [dB], k_f is number of the ceilings with L_f loss [dB]. Similar to OSM we need some measured values of variables L_{Wi} and L_f that are characteristic for modelled environment.

3.3.3 Classical Location Estimation Algorithms

We now describe three known, classical, location estimation algorithms.

Triangulation (TN)

Triangulation (TN): the algorithm forms circles centered at the RF access points, where the radius of each circle depends on the measured signal strength. The radius is approximated by comparing the run-time measurement with the information stored in the location database. A system of equations representing the circles is solved; typically, the vertices of a number of common areas shared by the circles are found using the solutions from the system of equations. Averaging the coordinates of the solutions that form the smallest area gives the final location estimate [13].

K-nearest neighbor averaging (KNN)

K-nearest neighbor averaging (KNN): the algorithm searches for K location entries from the location database having the smallest root mean square error in signal space with the given run-time measurement at the unknown location. Averaging the coordinates of the K locations gives the final location estimate [13].

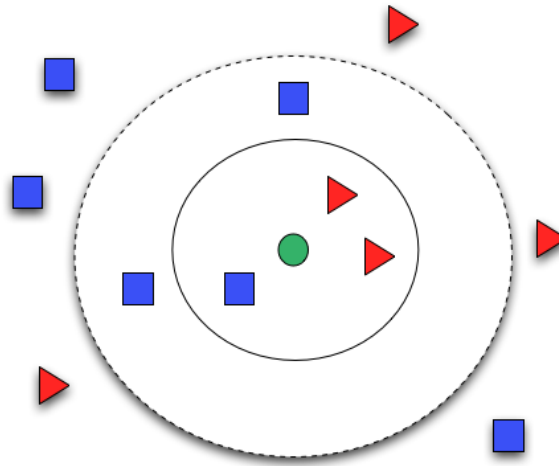


Figure 3.6: Example of KNN classification

The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $k = 3$ it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

Smallest M-vertex polygon (SMP)

Smallest M-vertex polygon (SMP): M candidate locations from each access point whose distance in the signal space with the given run-time measurement are searched from the location database. M-vertex polygons are formed by including at least one candidate location from each access point. The smallest polygon is the one having the shortest perimeter. Averaging the coordinates of vertices of the smallest polygon gives the final location estimate [13].

Besides this algorithms, recently a Machine Learning approach based on Support Vector Machine (SVM) has been proposed in [5] to model the complete fingerprint from few training data, and, as a pattern recognition algorithm, SVM is also utilized to estimate the user position. However, this technique doesn't suit fast changing scenarios.

3.4 Simulators and Planning Tools for Indoor Wireless Systems

3.4.1 WinProp by AWE Communications

One of the most sophisticated wave propagation and network planning CNP (Combined Network Planning) suits on the market. Developed in Germany, is a unique system that offers tools for many different standards for outdoor and indoor wireless signal propagation. Coverage, data throughput, interference and blocking can be predicted based on the path loss predictions obtained with ProMan or other

prediction tools. It uses pre-defined network planning modules for the following wireless standards:

- Broadcasting
- 2G and 2.5G Networks
- 3G and B3G Networks
- Wireless LANs (WLAN)
- WiMAX

Indoor propagation

Phenomena mentioned in previous sections, like multi-path propagation, reflection, diffraction and shadowing have a major impact on the received power. Based on 3D vector databases with planar objects WinProp is able to compute very fast path loss and wideband properties of the radio links inside buildings. To make it possible it uses different combinations of available propagation models for indoor scenarios. They can be broken down into two types:

- Empirical Models:
- Ray Optical Propagation Models

Modelling the multi-floor indoor propagation

The application includes also an option for multi floor buildings predictions. This type of calculation requires calculation on each floor. WinProp can compute predictions on an arbitrary number of heights and can display the result in a 3D views. Thus, interference between floors and coverage problems can be easily predicted.

Planning of W-LAN networks

WinProp offers fast and accurate propagation models integrated in a user- friendly GUI application. The user is able to define multiple access points' layout in the

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

building; as a result the coverage area can be computed for each access point separately or for an entire area as well. Based on the predictions of the signal propagation for each access point a set of radio network planning outputs is produced.

Network Optimization

Propagation of signal in the indoor environments requires detailed information about the walls and object inside the buildings. To make results as much accurate as it is possible WinProp uses 3D vector databases for its propagation models. Moreover, it includes an additional tool specially designed for building databases that are used in a process of signal propagation. For most indoor planning tools, the handling of the building data is the most critical part. This CAD tool allows the generation of building databases within a few minutes based on scanned bitmaps or CAD data.

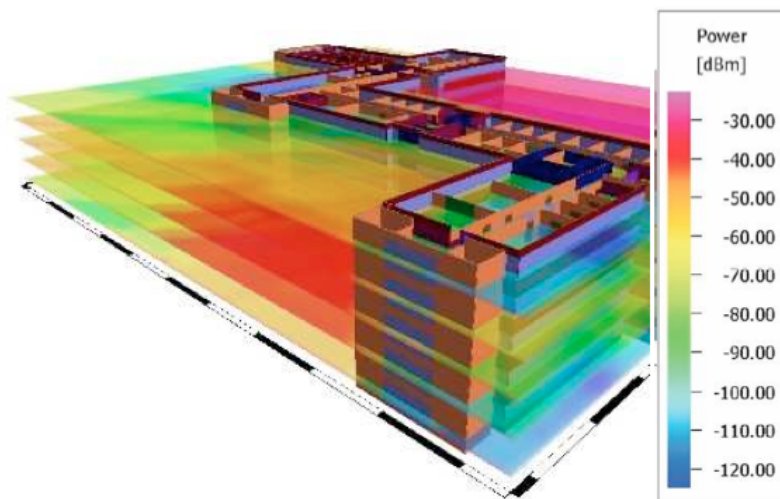


Figure 3.7: WinProp: Sample prediction with an indoor prediction model

3.4.2 WiSE: a Wireless System Engineering Tool by AT&T Bell Laboratories

WiSE is a simple system for signal wave propagation in indoor and microcell environments. Consist of three main functionalities: prediction, optimization and interaction. First option is responsible for computing predictions as an output, where building information (wall locations and composition), system parameters, and access points' locations are input data. Its main aim is to determine the wireless system performance. In a prediction computation process the basic propagation model with a several modifications is used. Second feature, given parameters and system requirements compute alternative layout of base-stations to optimize the network planning process. The last part of the system is the user interface. It displays plan, elevation, and perspective views of a building and shows in living colour the power received at each location for a given base-station site [14].

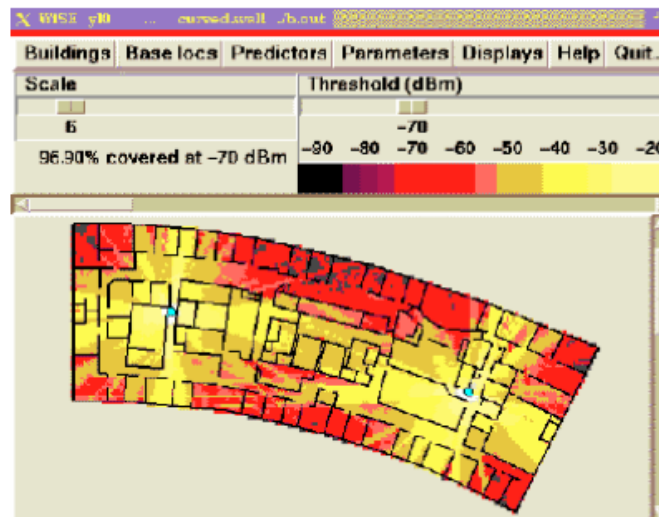


Figure 3.8: WiSE: Predicted coverage of a small office with two base stations

3.4.3 RaPSor

RaPSor is Radio Propagation Simulator developed in laboratory, used both in research and educational contexts. It succeeds three past versions, leading to an open and extensible tool. Based on Netbeans, a Rich Client Platform, it can be extended quite simply, using mainly four ways: Firstly, anyone can add a new antenna model, for instance to test (or learn) smart antenna. Secondly, new geometric primitives can be written. Thirdly, new simulation algorithms can be integrated inside RaPSor. Fourthly, any kind of simulation exploitation can be proposed. All these extensions can be realized as plugins, using mainly the inversion of control pattern coupled to the dependency inversion principle, and based on the Netbeans richness (eg. contextual Lookup). RaPSor ought to be an extensible, modular, and open-source tool. For portability, we choose the Java programming language, leading to an application running on many architectures and systems, like Unix, Linux, Windows or MacOS. In order to concentrate developers efforts on the business work, we first choose a Rich Client Platform (a.k.a. RCP). Due to the educational aspects of this project, we choose to work with Netbeans RCP. Such a platform provides all the redundant programming tools and method that any developer should write to its own application, as the file and window management, connecting actions to menu items, toolbars... Netbeans comes with many functionalities, and provides a reliable and flexible application architecture. Its modularity allows to select the ones a developer want to keep, and allows even the users to add or remove new or unnecessary tools. Moreover, users can write new tools, or plugins, using either the Netbeans IDE or another one, adding them later into RaPSor. RaPSor comes with some basic modules:

- The kernel, that provides all the necessary to write new simulation tools, like the antenna description (and interface), two simulators, an exploitation model, and the basic geometries with face and dihedron definitions.
- The windows-api, providing a main navigator, a narrow- band color panel, a main window to visualise in 2D or 3D the scenes, with or without simulation results, and a XML file format for input

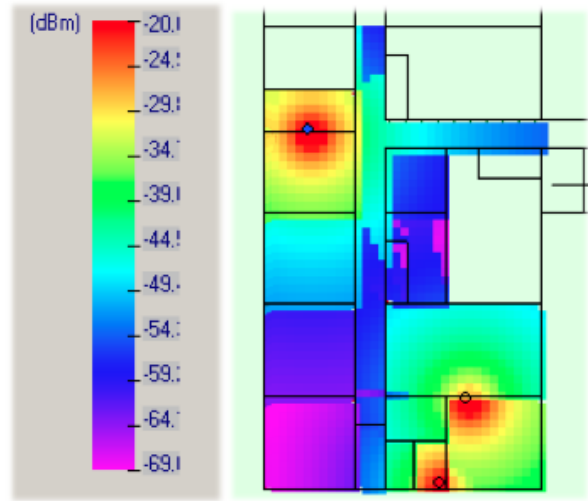


Figure 3.10: WinPlanner: example of signal coverage results in a section of a building

ments. It computes coverage and channel performance, explore indoor/outdoor interaction. In a prediction process application uses method based on a full three-dimensional implementation of GO/UTD (Geometrical Optics/Uniform Theory of Diffraction). Moreover, tool is capable of *ray tracing* carried out by combining Image Theory with Binary Space Partitioning algorithms. It provides features useful in a process of planning a wireless system such as coverage, fading statistics, power delay profile, and associated parameters, such as rms delay spread and the coherence bandwidth [14].

3.4.6 EDX Signal Pro by Comarco Wireless Technologies

EDX SignalPro is a complete software suit offering a complete set of planning tools for wireless communication systems. EDX SignalPro contains propagation prediction tools for wide area service prediction, link analysis, point-to-point and point-to-multipoint studies. One of the modules, called MIM (Microcell/Indoor Module), is specially designed for microcell and indoor wireless communication systems. It includes multi-site coverage and interference analysis, multiple point-to-point link analysis, a comprehensive set of propagation models, full mapping

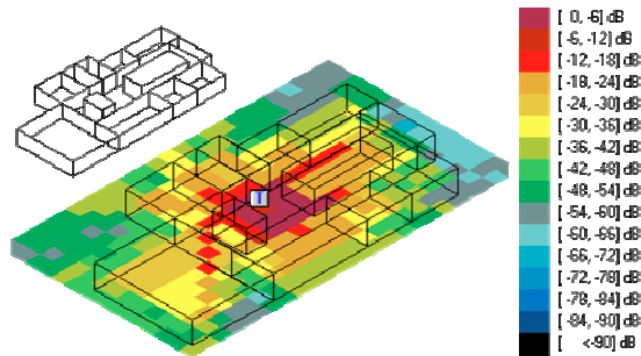


Figure 3.11: CINDOOR: coverage map

capabilities, and full access to terrain, groundcover, building, demographic, traffic, and other databases. It uses both two-dimensional and three-dimensional ray-tracing propagation models which have been optimized for accuracy and efficient performance in the indoor environment. Universal database provided with a suite includes detailed information about the RF characteristics of walls and other obstructions. Results of the computations can be displayed in a different ways, and several studies can be displayed simultaneously. For the best result EDX Signal Pro uses combination of different modifications of the following propagation models

1. 2D and 3D ray-tracing models for outdoor microcell and indoor wireless LAN/PBX/cell-extender environments
2. EDX Simplified Indoor Model for rapid, site-specific indoor signal strength calculations. Takes into account:
 - Line-of-sight rays
 - Wall transmission
 - Corner diffraction
 - Attenuation due to partial Fresnel zone obstruction

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

3. COST-231 Walfisch-Ikegami propagation model for simplified outdoor microcell studies.

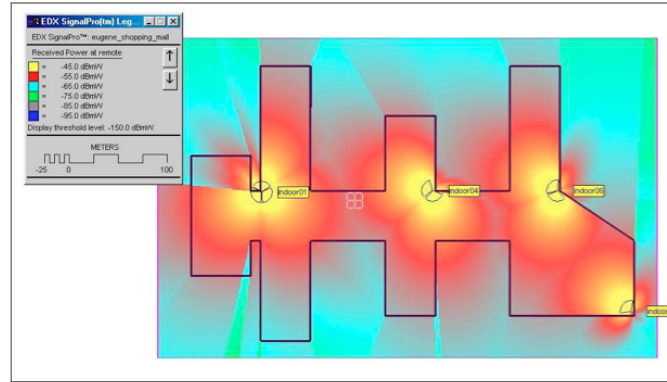


Figure 3.12: EDX Signal Pro: Predicted coverage area

What is more, application is capable of multi-level indoor signal propagation. Calculates attenuation between floors and plots results in 3D, multi-level graphs. In an addition to overall functionality of the system Time Signature Displays are available. Their main aim is to show the waveform of the received data pulse as a function of time. As a result, pulse shape distortions and echoes due to multipath are easily visible. Pseudo-animation mode shows the dynamic nature of the distortions [14].

Chapter summary

In this chapter we have seen how RSSI-based indoor localization systems are growing in importance among other indoor positioning techniques, especially in WLAN networks, since they do not require to deploy an ad hoc infrastructure but it is possible to leverage the same equipments used for providing network access to the tracked users.

We have provided an introduction to the Wifi IEEE 802.11 technology, and to the problems that arise in RSSI localization techniques in noisy environments with both static and moving targets.

CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH INDICATION

Finally, we have described classical location estimation algorithms and introduced most popular signal propagation simulators.

*CHAPTER 3. LOCALIZATION BASED ON RECEIVED SIGNAL
STRENGTH INDICATION*

Chapter 4

The application scenario

In this chapter we are going to describe the context where we need to apply indoor WiFi localization: the tracking of customers moving around in a supermarket or a hypermarket. We will discuss pros and cons of the RSSI localization techniques exposed in Chapter 3 with respect to our application demands and show why WiFi 802.11 turned out to be best option. Moreover we'll discuss the issues arisen in the project of our infrastructure for real time tracking of customers just as they move around the shop area and combine this information with a pre-existing profile of the customer in order to provide an enhanced loyalty program and more effective commercial proposals.

4.1 User profiling in loyalty programs

Loyalty programs are structured marketing efforts that reward, and therefore encourage, loyal buying behavior, which is potentially beneficial to the firm [16].

In marketing generally and in retailing more specifically, a loyalty card, rewards card, points card, advantage card, or club card is a plastic or paper card, visually similar to a credit card, that identifies the card holder as a member in a loyalty program. Loyalty cards are a system of the loyalty business model. In the United Kingdom it is typically called a loyalty card, in Canada a rewards card or a points card, and in the United States either a discount card, a club card or

a rewards card. Cards typically have a barcode or magstripe that can be easily scanned, and some are even chip cards. Small keyring cards (also known as key-tags) which serve as key fobs are often used for convenience in carrying and ease of access.

A retail establishment or a retail group may issue a loyalty card to a consumer who can then use it as a form of identification when dealing with that retailer. By presenting the card, the purchaser is typically entitled to either a discount on the current purchase, or an allotment of points that can be used for future purchases. Hence, the card is the visible means of implementing a type of what economists call a two-part tariff.

The card issuer requests or requires customers seeking the issuance of a loyalty card to provide a usually minimal amount of identifying or demographic data, such as name and address. Application forms usually entail agreements by the store concerning customer privacy, typically non-disclosure (by the store) of non-aggregate data about customers. The store uses aggregate data internally (and sometimes externally) as part of its marketing research. These cards can be used to determine, for example, a given customer's favorite brand of beer, or whether he or she is a vegetarian.

Where a customer has provided sufficient identifying information, the loyalty card may also be used to access such information to expedite verification during receipt of cheques or dispensing of medical prescription preparations, or for other membership privileges (e.g. access to a club lounge in airports, using a frequent flyer card).

4.2 New user-profiling techniques in retail supermarkets

In the previous section we have seen how main supermarket chains have built huge databases filled with data gathered from their customers' habits. Over the years, customers have been profiled according to their buying habits, family type, overall spending amount and so on, in order to create targeted, and thus more effective,

loyalty programs.

Nowadays, the next frontier in this scenario is the possibility of a real-time tracking of customers as they move around the supermarket area and fill their shopping trolleys.

This possibility would enable for the first time the study of the actual paths followed by customers during their shopping time. The knowledge of these paths could be quite important in order to perform aggregate analysis and determine the places where customers pass by more often, where they stop for a longer time, maybe because attracted by an effective commercial offer, or, on the other hand, if there are "mothballed" areas in the shop that could be employed in a better way.

What is more, thanks to modern barcode reader embedded in trolleys, it is possible to combine these information with the knowledge of what a customer has put in his trolley just as he grab the product from a shelf, so it is possible to propose to him special customized offers, thus resulting more effective.

In order to enable all these services, it is necessary to deploy a proper application and network infrastructure, as the one we have implemented.

4.3 The typical supermarket scenario

In this section we are going to describe the physical context where we want to operate our tracking infrastructure: a big-sized mall or supermarket.

Such supermarkets spans an exhibit surface of at least 2500 m^2 in which may transit much more than a thousand customers per day.

A supermarket or hypermarket is a self-service store offering a wide variety of food and household merchandise, organized into departments. It is larger in size and has a wider selection than a traditional grocery store, also selling items typically found in a convenience store.

The supermarket typically comprises meat, fresh produce, dairy, and baked goods departments, along with shelf space reserved for canned and packaged goods as well as for various non-food items such as household cleaners, pharmacy products and pet supplies. Most supermarkets also sell a variety of other

household products that are consumed regularly, such as alcohol (where permitted), medicine, and clothes, and some stores sell a much wider range of non-food products.

The traditional suburban supermarket occupies a large amount of floor space, usually on a single level. It is usually situated near a residential area in order to be convenient to consumers. Its basic appeal is the availability of a broad selection of goods under a single roof, at relatively low prices. Other advantages include ease of parking and frequently the convenience of shopping hours that extend far into the evening or even 24 hours a day. Supermarkets usually allocate large budgets to advertising, typically through newspapers. They also present elaborate in-store displays of products. The stores are usually part of corporate chains that own or control (sometimes by franchise) other supermarkets located nearby (even transnationally) thus increasing opportunities for economies of scale.

Supermarkets usually offer products at low prices by reducing their economic margins. Certain products (typically staple foods such as bread, milk and sugar) are occasionally sold as loss leaders, that is, with negative profit margins. To maintain a profit, supermarkets attempt to make up for the lower margins by a higher overall volume of sales, and with the sale of higher-margin items. Customers usually shop by placing their selected merchandise into shopping carts (trolleys) or baskets (self-service) and pay for the merchandise at the check-out. At present, many supermarket chains are attempting to further reduce labor costs by shifting to self-service check-out machines, where a single employee can oversee a group of four or five machines at once, assisting multiple customers at a time.

Other services offered at some supermarkets may include those of banks, cafés, childcare centres/creches, photo processing, video rentals, pharmacies and/or petrol stations.

4.3.1 Typical store architecture

Whilst branding and store advertising will differ from company to company, the layout of a supermarket remains virtually unchanged. Although big companies spend time giving consumers a pleasant shopping experience, the design of a su-

permarket is directly connected to the in-store marketing that supermarkets must conduct in order to get shoppers to spend more money while there.

Every facet of the store is mapped out and attention is paid to colour, wording and even surface texture. The overall layout of a supermarket is a visual merchandising project that plays a major role in retailing. Stores can creatively use a layout to alter customers perceptions of the atmosphere. Alternatively, they can enhance the stores atmospherics through visual communications (signs and graphics), lighting, colours, and even scents [17]. For example, to give a sense of the supermarket being healthy, fresh produce is deliberately located at the front of the store.

As explained in [18], supermarkets are designed to 'give each product section a sense of individual difference and this is evident in the design of what are called the anchor departments; fresh produce, dairy, delicatessen, meat and the bakery'. Each of these sections has different floor coverings, style, lighting and sometimes even individual services counters to allow shoppers to feel like there are a number of markets within this one supermarket.

4.4 Evaluating possible technologies

As mentioned in Chapter 2, the costs of a location-sensing system can be assessed in several ways:

- Time costs: are affected by the installation process's length and the system's administration needs;
- Space costs: include factors like the amount of installed infrastructure components and the hardware's size;
- Infrastructural costs: are mainly affected by the price per mobile device unit or infrastructure element.

Therefore Received Signal Strength Indicators (RSSI) are used as input of the localization system because the acquisition of the values has only low demands on the hardware components.

4.4.1 Environmental Issues

Macroscopic objects with metallic surfaces are characteristic for supermarkets environments. Therefore not only line of sight (LOS) connections but also non line of sight (NLOS) connections between transmitter and receiver have to be considered. As a consequence of the multipath propagation by reflection, diffraction and dispersion effects interferences of different multipath components occur at the receiver.

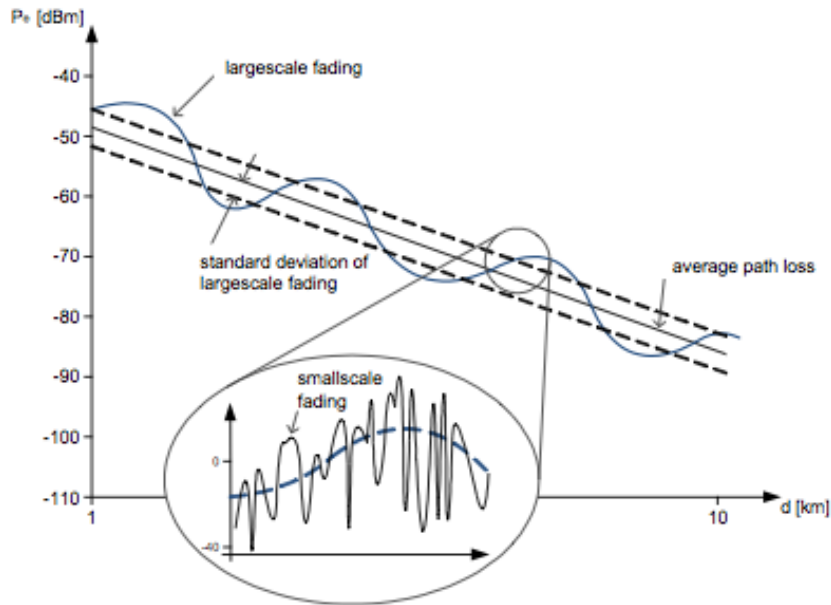


Figure 4.1: Distance-depending path loss affected by signal fading.

As represented in Figure 4.1, the distance-dependent path loss at the propagation of electromagnetic waves between transmitter and receiver is characterized by a superposition of different fading components. For this reason a complex model with distance and environment-dependent portions for the description of the path loss have to be assumed [19].

4.4.2 Field requirements

The requirements of a particular application could impact the choice of the architecture for a positioning system. For most commercial applications, it is desirable to have a system that is easily deployable and exhibits robustness against changes in the environments, interference and propagation model imperfections. Eliminating the offline measurement-based training phase is an essential step in providing robust methodologies that are implementable on low cost, low complexity infrastructure. This is an obstacle in practical implementation of positioning systems that are quickly deployable on commodity networks.

Moreover, localization technology should be unobtrusive and provide tangible advantages to the final user in order to gain success. So, beside costs and ease of deployment, we should consider factors such as *comfort* and *acceptance* [20]:

- Comfort: the system should be inconspicuous (for the user) and thus not demand to carry any hardware;
- Acceptance: besides the mentioned aspects (e.g. comfort and cost) which influence the degree of acceptance indirectly, the system should be developed under the consideration of additional aspects like electromagnetic pollution and privacy;
- Accuracy: The system should be accurate enough (precision, update rate) to track movements and to determine the position of a person;
- Scalability: the system should be able to track several persons per room with justifiable computational power;
- Installation effort: the system should be easy to install and unobtrusive (with respect to the environment);
- Cost: the system should be built with inexpensive standard components;

4.4.3 Technology tradeoff

From what we have explained up to now, WiFi emerges as the best candidate technology to our purpose, since it ensures sufficient localization precision and accuracy and, at the same time, it is low cost, widespread and widely accepted by users.

Moreover, it is often already deployed in many retail stores and it also grants the network capabilities necessary to support our computer infrastructure.

All other feasible RSSI technologies have proved to be unsuitable for our purposes, as we are briefly going to see:

- Ultra-wideband: although it could grant an unmatched precision as for localization (it is used in "see-through-the-wall" precision radar imaging technology precision locating and tracking using distance measurements between radios, and precision time-of-arrival-based localization approaches), it has still prohibitive costs.
- Bluetooth: its range is too low for our scenario, so it would be necessary too high density of anchors in the supermarket. Moreover Bluetooth uses a radio technology called frequency-hopping spread spectrum (FHSS), which chops up the data being sent and transmits chunks of it on up to 79 bands, which is not suitable for a reliable RSSI analysis aimed at localization.
- RFID: also the RFID range is very low. Moreover, the main limitation resides in the necessity of huge electric powered readers that are not suitable to reside on a shopping trolley.
- ZigBee: similarly to Bluetooth, it has a too low range and it is not widespread. It could not provide a reliable network infrastructure so it would force us to deploy a WiFi Network in parallel.

4.4.4 WiFi constraints

Once we have chosen to adopt WiFi, we have to face some additional constraints and issues generally not taken into account in the already existing literature.

A paramount requirement soon emerged during our tests is the need to find a way to keep the RSSI map updated with a proper frequency. Since the movement of customer with trolleys around the supermarket has a strong impact on the WiFi signal propagation, we need to continuously vary the expected RSSI value in a certain point, especially if we consider that we have to track even hundreds persons at the same time. In such scenario, building an offline fingerprint map of RSSI values is definitely unfeasible.

Secondly, during our testing measures we found out how RSSI values are subject to strong fluctuation of 10dBm in the span of a day, even in an empty room. In other words, our receiver terminals in testing points registered different power values in the same place of an empty room in different moments of a day, due to fluctuations of the room temperature and humidity.

Even standing still with the mobile terminal in hand and in line-of-sight with a near Access Point with no obstacles, we can observe significative RSSI fluctuations. This effect is also well known in literature, such as in [21], and can be clearly summarized in figure 4.2, showing fluctuations of the power level read by a WiFi terminal in line-of sight with an access point.

Moreover, due to the particular shelf disposition and power supply availability of a supermarket we are forced to assume that WiFi Access Points (AP) are placed in a high position, e.g. just below the roof, which means that we could not exploit inter-AP RSSI measures to have an estimated of the WiFi signal propagation between APs and mobile terminals.

The on-line RSSI measurements among APs are often used to capture (in real-time) the effects of RF multi-path fading, temperature and humidity variations, opening and closing of doors, furniture relocation, and human mobility on the RSSI measurements.

This techniques has been widely exploited in literature [22] [11] and grants excellent results in building a RSSI map of the environment.

It is also recommended that all Access Points deployed are of the same kind, (i.e. same chip, firmware and antenna), since taking into account different kind of antenna pattern leads to an increased complexity in building the RSSI map.

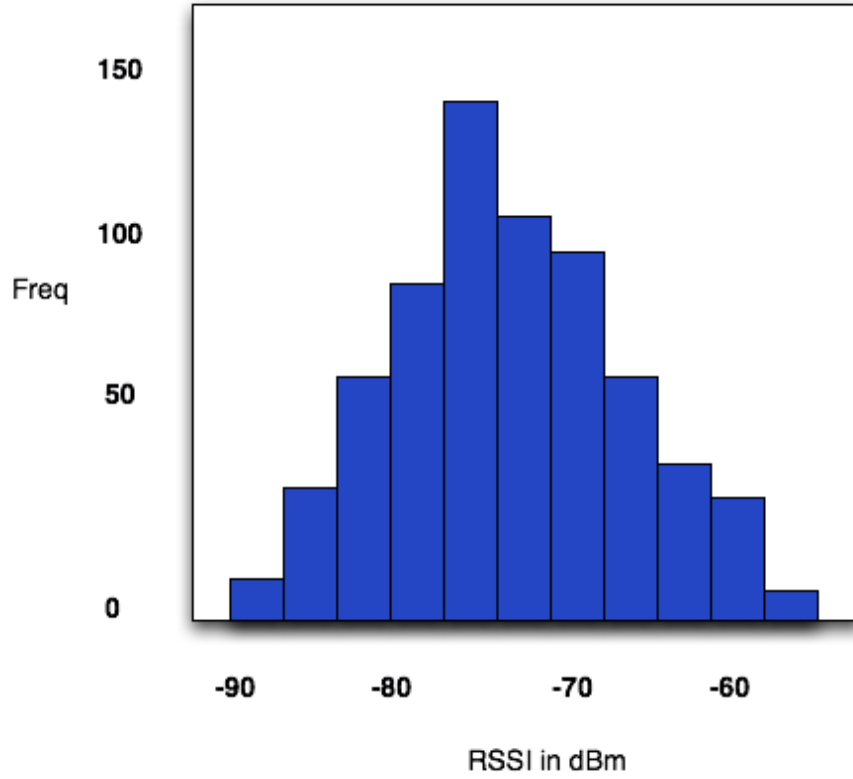


Figure 4.2: RSSI variation over time.

Moreover different chips and firmwares may have inconsistencies when regulating the transmitting power of the Access Point, leading to a situation where is very difficult to compensate offsets. In order to reduce scattering and reflections, we calibrated the transmitting power of all Access Points at no more than 10mW, which simplifies the prediction of the RSSI map. For the same reason, we assigned to each Access Point a different non-adjacent channel in order to reduce interference.

A similar problem resides mobile terminal side: different mobile terminals with different antennas, although in the same position and orientation may register different RSSI values from the same AP up to 15dBm. Again, using homogeneous devices greatly simplifies the calibration a propagation model and the compensation of offsets. This is particularly important if we think a human body is a strong

barrier for the microwaves of a WiFi signal. So keeping a mobile terminal in one hand may cause an attenuation of even 20dBm with respect to having the mobile terminal in the same place and orientation but laid on a table. So, it is vital to have at least good offset compensation in order to cope with the ever-present natural signal attenuation factors.

Chapter summary

In this chapter we have provided a detailed portrait of the of indoor WiFi localization oriented to track customers moving in a supermarket. Our solution enables the combination of real time information with a pre-existing profile of the customer in order to provide an enhanced loyalty program and more effective commercial proposals.

Also, we have discussed pros and cons of all possible RSSI localization techniques w.r.t. our application demands and showed why WiFi 802.11 was chosen. Finally we have provided a comprehensive explanation of major issues arisen in dealing with WiFi networks during the development of our infrastructure, focusing on signal propagation and antennas concerns.

In the following chapters we will go in detail in the development of our solution from the point of view of the algorithm and the architecture.

Chapter 5

WhAC: the algorithm

In this chapter we'll describe in detail our proposed algorithm for indoor WiFi localization, designed to suit fast-changing signal propagation conditions and to satisfy all requirements and constraints derived from noisy environments exposed in the previous chapter. It combines the benefits of simulating empirical propagation models in terms of lowering the offline stage effort with the unmatched precision of fingerprinting techniques, merging the two approaches together.

Moreover, in dynamic localization, we'll show how it is possible to improve it further with the adoption of a *Particle Filter* in order to incorporate the knowledge of cinematic models of the moving target.

We will also provide a comparative analysis of our method with respect to similar approaches, presenting the obtained results both in static and dynamic localization efforts.

Our WhAC (*Where Are my Customers*) application will make use of this algorithm in order to locate customers moving around in a mall.

5.1 Reducing fingerprinting offline stage effort

As explained in chapter 2, *fingerprinting* is the most accurate technique for building a reliable RSSI map of the area of interest, since it provides actual power readings of mobile terminals. Once we have a RSSI map, we can run a location

algorithm of our choice, such as *Triangulation*, *K Nearest Neighbour*, *Smallest Vertex Polygon* or other forms derived from these 3 methods to determine the user location. In general, there isn't a location estimation algorithm best suited for all environments or technologies. In [23] is considered the combination of multiple algorithms and it is showed that combining them properly can indeed provide significant benefits.

5.1.1 Impact of the Number of Data Points

In [24] is studied how the accuracy of location estimation is impacted by the variation of the fingerprint density, in other words, they have studied what happens when they consider less fingerprints of the environment thus making the map less *dense* (e.g. they considered data from fewer distinct physical locations than the all the collected ones), and how is the localization accuracy afflicted. Varying the number of physical locations (ranging between 2 and 70), they conducted 20 runs of their analysis program. In each run, they picked n points at random from the entire data set collected during the off-line phase and used this subset to construct the search space for the their algorithm. They collected the error distance data from all the runs corresponding to the same value of n (5.1)

Similarly, in [25], it is showed the influence of *granularity* of the RSSI map, the database entries are collected on a grid of points within the building. The spacing between grid points influences the granularity of the position estimate. Decreasing the spacing (e.g. taking RSS measurements every foot) will increase the database size but are unlikely to yield a better accuracy because the RSS values measured a foot apart will be more or less the same. On the other hand, if the spacing is very large, it may reduce the search space but drastically decrease the accuracy. In other words: increasing the granularity of the grid only slightly increase the positioning accuracy. However, the size of the database is increased many times, so search time for matching the fingerprint is consequentially increased.

Their best results, one of the most promising we could find in literature [25], used a granularity of about $1.5[m]$ and in each point of the map was calculated

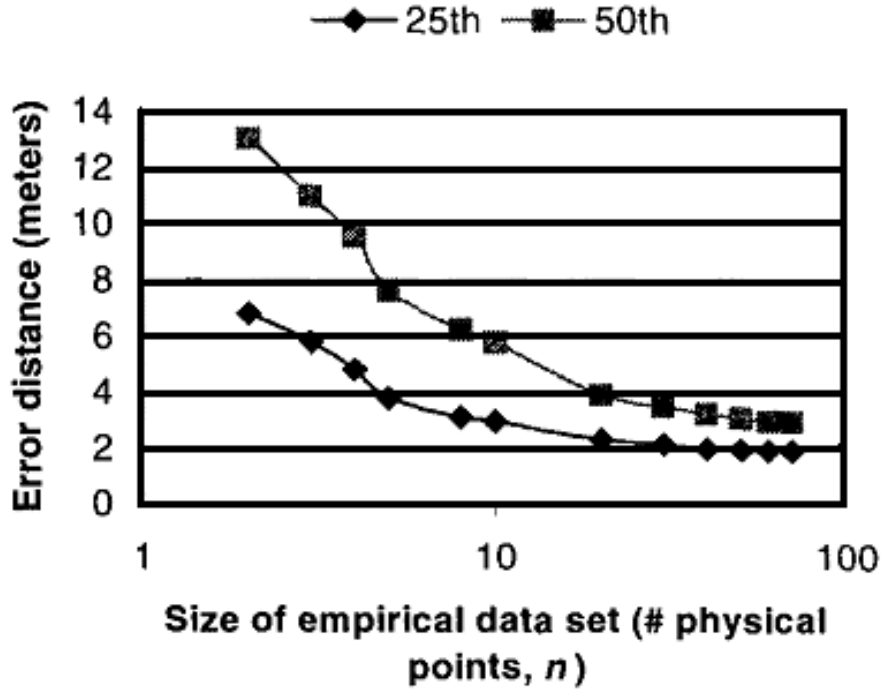


Figure 5.1: The error distance versus the size of the empirical data set (on a log scale).

the average of 40 samples, composed of 5 samples in each of the Mobile Stations orientation at two different times of the day. Moreover, they used four orientations at each location. With such a huge effort, they scored a localization error of $3[m]$ at the 75th percentile.

5.1.2 Impact of the Number of Samples

In the analysis presented in [24], they have worked with the mean of all of the samples recorded during the off-line phase for each combination of location and orientation. While it may be reasonable to construct the empirical data set with a large number of samples (since it is a one-time task), there may be constraints on the number of samples that can be obtained in real-time to determine a user location. So they investigated the impact of a limited number of real-time samples (while retaining the entire off-line data set for search space) on the accuracy of location estimation. Their analysis shows that only a small number of real-time

samples are needed to approach the accuracy obtained using all of the samples. With just 1 real-time sample, the median error distance is about 30% worse than when all samples were considered. With 2 samples, it is about 11% worse and with 3 samples it is under 4% worse.

5.1.3 Impact of User Orientation

In [24] is showed how the user orientation has a significant impact on the signal strength measured at the base stations. They considered, in some sense, the worst case where the off-line data set only has points corresponding to a particular orientation (say north) while the real-time samples correspond to the opposite orientation (i.e., south). They computed the error distance for all four combinations of opposing directions: north-south, south-north, east-west, and west-east, and observed a fairly significant degradation in the accuracy of location estimation. For instance, for the north- south case, the 25th percentile of the error distance is 2.95 meters (54% worse than when having a RSSI map with 4 orientation per point) while the 50th percentile (median) is 4.90 meters (67% worse). This degradation underscores the importance of obtaining empirical data for multiple orientations to construct the search space. However, even in this worst case, the empirical method outperforms the strongest base station and random methods by a factor of 2 to 4.

5.2 Direct comparison of Simulator vs Fingerprinting in static localization

We conducted our own experiments in **static** localization with Fingerprinting and a simulator [26] based on the empirical *MultiWall propagation Model* (MWM).

We used 7 Access Points model Linksys WRT54G 5.2, on which was installed the OpenWRT firmware [27] in order to set their transmitting power to 10mW to reduce signal reflections and scattering effects. The Beacon Period was set to 250ms, in order to have at least 4 RSSI readings each second, and to each Access

Point was assigned a different non-adjacent channel in order to reduce interference. We collected data from a Samsung Galaxy Mini GT5800 [28] as both for building the database of the fingerprints and as mobile terminal to be localized. The mobile devices record Received Signal Strength (RSS) measurements corresponding to APs in their view at various (unknown) locations and report these to a localization server.

The following tests were carried out in an office of about $160m^2$.



Figure 5.2: Linksys WRT54G

We tried lowering the offline stage effort in building the fingerprint map making the grid less dense, in order to get a way to employ always an acceptable amount of time.

To this aim, we performed a static localization comparison between fingerprinting with a grid spacing of $2.5[m]$ and a simulation with a grid spacing of $0.1[m]$.

This "low-effort" fingerprinting proved to be a really time consuming effort anyway, although very reduced with respect to the procedure described in [24], averaging six power readings in each point, with the mobile terminal randomly pointing at one direction. However, our results show that it is not possible to

simplify the fingerprinting further in order to achieve an acceptable localization.

Despite building the fingerprint database required us more than half hour, our results in figure 5.4 show that in such case the performances of fingerprinting fall under those of the simulator. In particular, taking into consideration the 80th percentile as reference, it is evident that the localization error via simulator is even 1 meter lower than that derived from pure fingerprinting method, requiring a much lower effort.

As for the localization algorithm, we used *K-nearest neighbor*. *KNN* searches for K location entries from the location database having the smallest root mean square error in signal space with the given run-time measurement at the unknown location. Averaging the coordinates of the K locations gives the final location estimate. In literature, beside the simplest solution represented by a computationally intensive *full search*, there are optimized data structures based on partitioning the search space that greatly reduce the complexity of the Nearest Neighbour search, such as *K-D-Trees* and *R-Trees*. Using an appropriate nearest neighbor search algorithm makes k -NN computationally tractable even for large data sets.

We performed a comprehensive study of the impact of the number of considered nearest neighbours (the K of KNN) on the localization accuracy, our obtained results are shown in fig. 5.3.

From the analysis of the collected datas it clearly emerges that for high K , above $K=85$, the localization error tend to became stable, thus making useless increasing it. Interestingly, we have even smaller errors with K between 5 and 10: this behaviour can be easily exploited for speeding up computation greatly reducing the search effort. Despite the graph, we do not suggest to set $K=1$, since in some cases it could led to great error without the possibility to balance them taking into account other neighbours in signal space, thus getting unreliable position estimates. For the rest of this section we assume to use $K=90$.

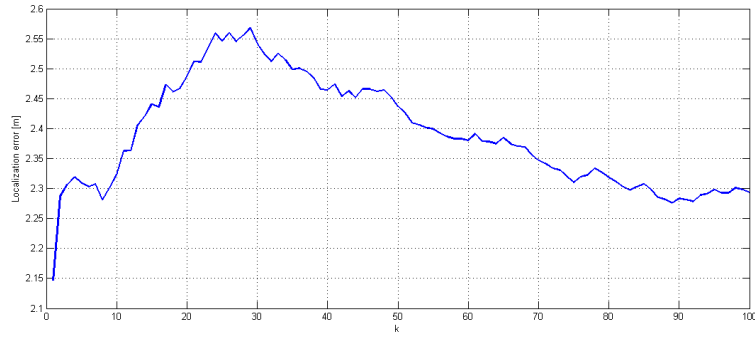


Figure 5.3: Fingerprinting: impact of the number of nearest neighbours on localization error

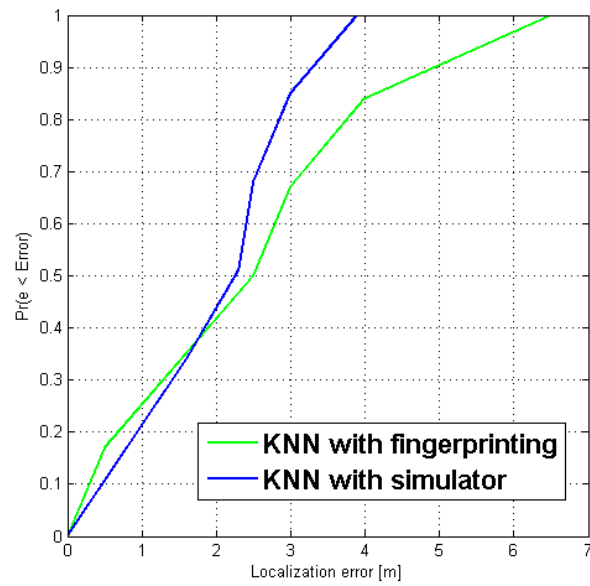


Figure 5.4: Cumulative density function of the localization error in a test area of approximately $160m^2$ with 7 AP.

5.3 Static localization in noisy supermarket-like environments

In this section we are going to see how static localization performance based on simulators decrease in "noisy" environments and what is our approach to deal with this problem.

The tests described in this section were carried out in a network laboratory of about $100m^2$, with high metal closets arranged in lanes, in a disposition similar to what we could usually find in supermarkets, characterized by non line-of-sight propagation due to the presence of walls and furniture.

We used the same 7 Access Point as before while our mobile device was a Samsung Galaxy Tab [29], laid on a metal trolley, like those used in supermarkets.

While we gathered datas, we had 6 other persons moving randomly around the laboratory in order to simulate moving customers.

This time, as shows image 5.6, localization error via simulator approach is monotonically decreasing at the increasing of the K considered nearest neighbours, hence we had always considered between 90 and 100 points in signal space and averaged their coordinates. This is because in a noisy environment we need to take into account many more points than in the previous experiments in order to compensate power readings altered by moving people with respect to what expected from the simulation. In our experiments we tried to consider even more than 100 points but it simply slowed down computation without benefits in localization accuracy, and for very great K the error began to increase again.

As expected, the overall localization error get slightly worse than in previous experiments.

At first step, we tried a simple approach to improve accuracy: scaling the propagation database obtained by the simulator by a different constant for each access point. In detail, our idea was to use a few mobile terminals as reference. We deployed these terminals sparsely in the laboratory area, while in a real supermarket, they should be placed in fixed known position between shelves or cashes. From their power readings, we calculate the average difference between the ac-

tually read value and the value predicted by the simulator for each access point. Then, we could literally scale the propagation database of such offsets in order to "calibrate" the propagation model. In this way we hoped to obtain predictions more similar to actual values, however, since for some terminals may result hindrance when the scaling is performed in a way that further increase the difference between the expected and the read value, in some cases with this procedure results get worse. So, the further step consists in scaling the propagation database in the radius of the reference mobile terminals under our control, weighting the read value with the predicted value. In detail, we applied formula 5.1, where P is the new expected power value, d is the distance of the predicted point from the position of the mobile terminal, r is the maximum radius in which we perform the correction, P_s is the expected value from the simulator with the scaling correction and Δ is the difference between the read and the expected power value from the terminal. The maximum radius r should be chosen in order to avoid interferences between adjacent reference terminals. Since the position of the reference terminals seldom changes, we can cache the points within radius r from a terminal in order to avoid a full search on the whole map, thus greatly speeding up computation. This *training phase* is summarized in schema 5.5.

$$P = ((1 - (d/r)) * (P_s + \Delta)) + (d/r * P_s) \quad (5.1)$$

Figure 5.6 clearly show that with our proposed method we can improve the accuracy with respect to using the simulator alone, and achieve a medium static localization error below $2.4[m]$ although in a noisy environment. Moreover, figure 5.7 shows that in 80% of the cases, static localization error is below $3.5[m]$, yielding better results that with simulator alone.

In the rest of this section it follows a comparison with the *LAURA* system [22], which performs localization, tracking and monitoring of patients hosted at nursing institutes by exploiting a wireless sensor network based on the IEEE 801.15.4 (*Zigbee*) standard.

The *LAURA* system was developed in our same laboratory and is similar to our system in the purposes and in using RSSI techniques for localization, since

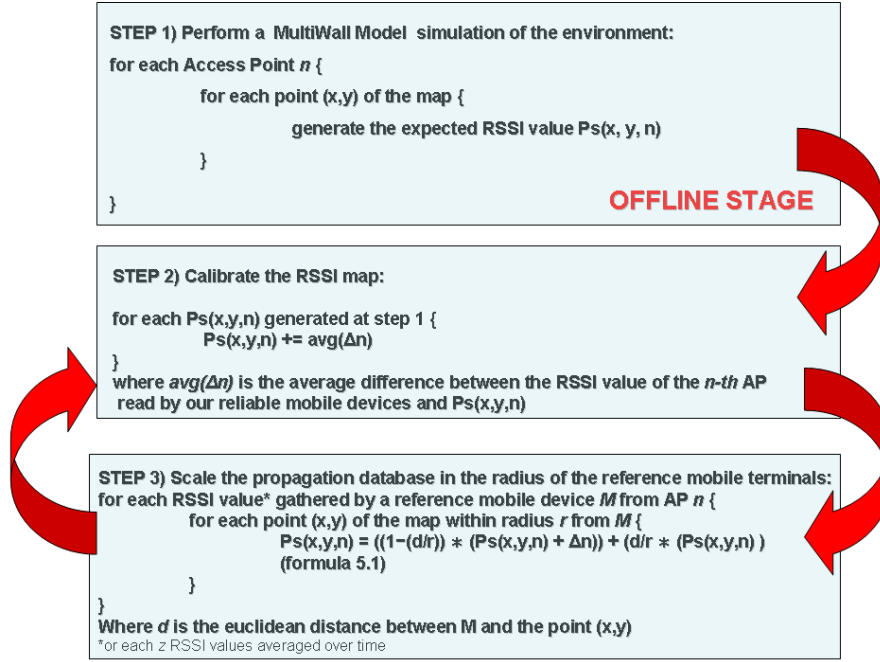


Figure 5.5: Schema of our dynamic localization algorithm: Training phase

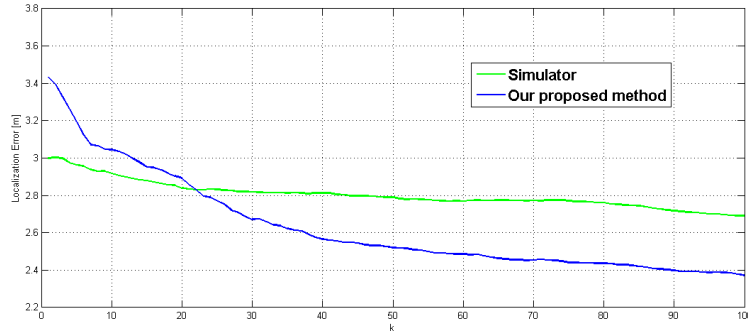


Figure 5.6: Simulator in noisy environment: impact of the number of nearest neighbours on localization error.

we have found ourselves facing many of the problems already faced by *LAURA* developers.

Differently from our solution, in *LAURA* anchors nodes collect inter-AP measures in order to create a propagation model of the current environment, that is a techniques we could not use due to our environmental constraints. *LAURA* also

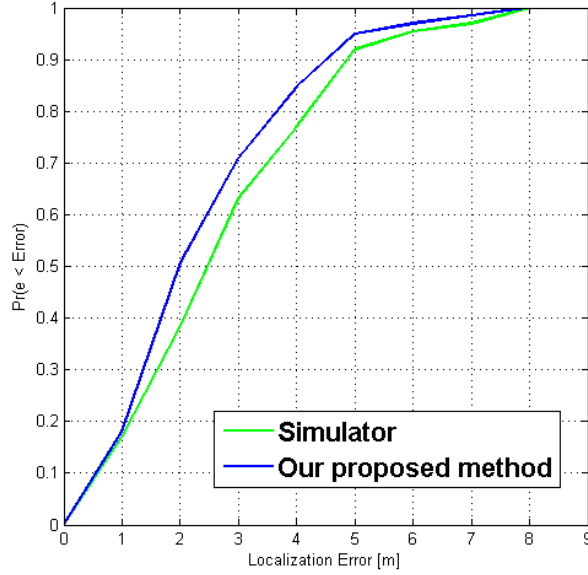


Figure 5.7: Cumulative density function of the localization error in a supermarket area of approximately $100m^2$ with 7 AP.

have a much denser distribution of anchor nodes, since *Zigbee* has a much more limited range than *WiFi*. Despite we aren't allow to use inter-AP measures, our results are comparable in localization performances and maintains a good accuracy as shown in figure 5.8.

5.4 Dynamic localization

In this section we are going to perform dynamic localization (tracking) of moving persons. Our test were performed in the same place and conditions as in static localization, and, at first, our algorithm remained exactly the same.

Now the situation is much more critical than before since a moving target experiences extremely fast-changing signal power readings, and the mobile terminal is able to send to the localization infrastructure no more than one power reading at each step if moving at the average speed of $1[m/s]$.

Obviously in such condition the localization error is greater than in static case

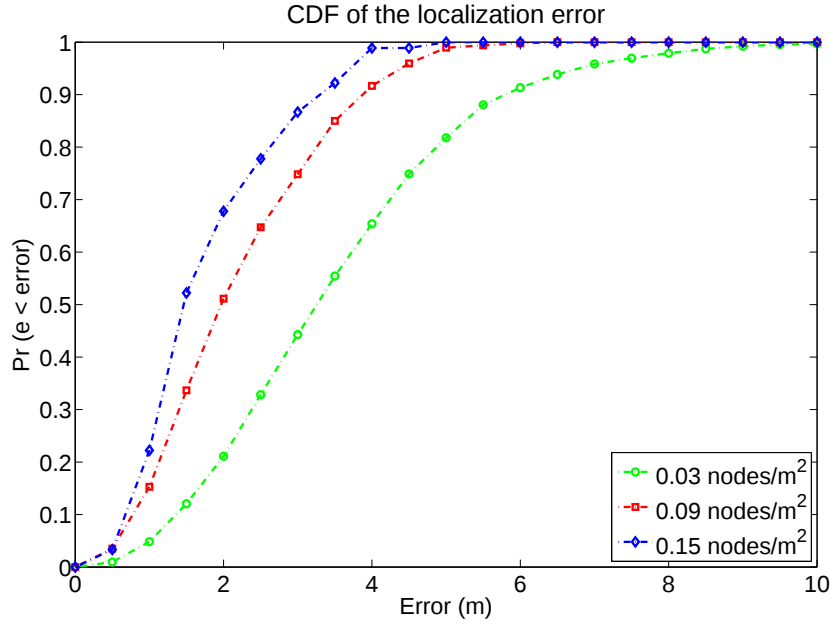


Figure 5.8: LAURA: Cumulative density function of the localization error in a test area of approximately $100m^2$.

analysis, as shown in figure 5.9.

Autoregressive model

At this point we tried some experiments in order to improve accuracy in the tracking case. Our first idea was to employ a simple *autoregressive model* described in formula 5.2. An autoregressive model is a type of random process which is often used to model and predict various types of natural phenomena. The autoregressive model is one of a group of linear prediction formulas that attempt to predict an output of a system based on the previous outputs. It's main fields of application are statistics and signal processing. In our case, $x(t+1)$ stands for the next position of the moving target to be predicted, $x(t)$ is the current estimated position of the moving target and α is a tunable parameter between 0 and 1.

We tried various values of α , and we obtained the best results with values comprised between 0.8 and 0.9. Unfortunately, as shown in figure 5.9, this approach was not very successful, since, although improving the simulator alone

performances, did not prove to outclass our used approach used up to now.

$$x(t+1) = \alpha * x(t+1) + (1 - \alpha) * x(t) \quad (5.2)$$

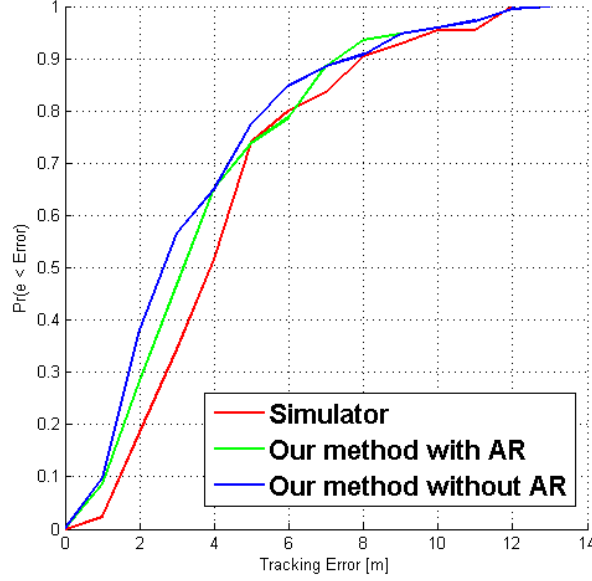


Figure 5.9: Cumulative density function of the tracking error in a supermarket area of approximately $100m^2$ with 7 AP.

5.5 Improving localization with Particle Filtering

As we sad before, in static conditions and with fixed client nodes, the our localization algorithm performs fairly well, as robust RSSI measurements can be obtained by means of time averaging. Conversely, in case of moving client nodes, severe short-time RSSI fluctuations are observed, especially when the target held the mobile terminal in its hand. As such, the accuracy of the estimated location decreases, and artificial motion discontinuities are detected. Incorporating a-priori knowledge about the moving node and the geometry of the environment might help improving the localization accuracy. In our system, we employ a *Particle Filtering* (PF) to track the position of the client node over time.

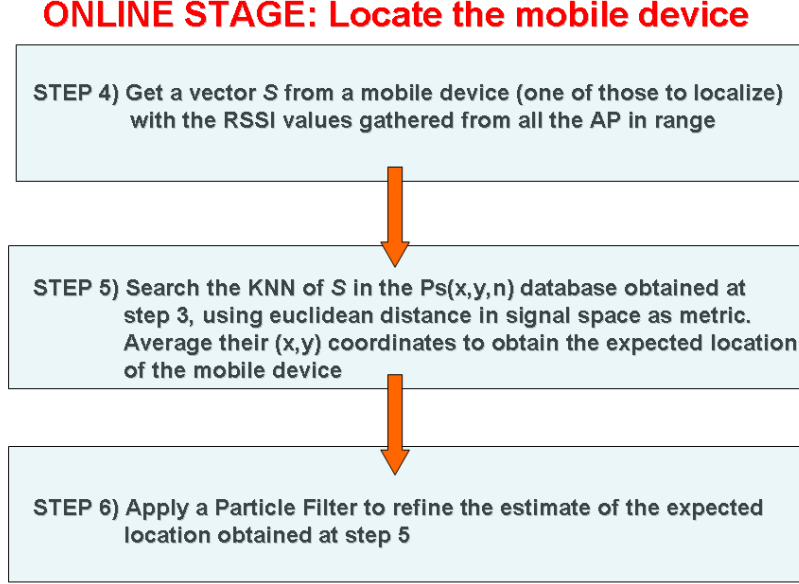


Figure 5.10: Schema of our dynamic localization algorithm: Online Stage

At each time instant t , the state of the node is represented by the vector $z(t) = [x(t)^T, v(t)^T]^T$, where $x(t) \in R^2$ indicates the position of the node and $v(t) \in R^2$ its velocity. PF estimates the a-posteriori probability density function of the state z at time t , given the sequence of previous states and all available observations. Such density function is represented in non-parametric form by means of a set of particles $p = 1, \dots, M$, each associated to a state vector $z_p(t) = [x_p(t)^T, v_p(t)^T]^T$ and a weight $\omega_p(t)$. Thus, we compute a point-wise estimate of the position and velocity of the client node as follows:

$$\begin{cases} \hat{x}(t) = (1 - \alpha_x)(\hat{x}(t-1) + \hat{v}(t-1)\Delta_T) + \alpha_x \sum \omega_p(t) \hat{x}_p(t) \\ \hat{v}(t) = (1 - \alpha_v)\hat{v}(t-1) + \alpha_v(\hat{x}(t) - \hat{x}(t-1)) \end{cases}$$

where Δ_T is the sampling period and α_x, α_v are the adaptation rates, which are tuned experimentally based on the node dynamics. For the problem at hand, PF alternates the execution of two steps:

1) *Prediction*: At each step, we compute the new state vector for the p -th particle. To this end, we employ the following dynamic model constructed upon kinematic equations

$$\begin{cases} x_p(t) &= x_p(t-1) + v_p(t-1)\Delta T + \xi_x \\ v_p(t) &= v_p(t-1) + \xi_v \end{cases}, \quad (5.4)$$

where ξ_x and ξ_v are samples drawn from zero-mean Gaussian random variables with variance, respectively, $\sigma_{\xi_x}^2$ and $\sigma_{\xi_v}^2$, which provide the system with a diversity of hypotheses. If the geometry of the environment is known, it can be incorporated in the dynamic model, e.g. by assigning $\omega_p = 0$ to those particles that crossed a wall.

2) *Update*: After the prediction step, the weight of the p -th particle is updated based on the RSSI measurements collected at time t . We define $\hat{d}_i$ as the distance between the i -th anchor and the position of the moving target estimated with our WhAC localization method. We assign a weight computed as in formula 5.5, considering the difference between the estimated distance $\hat{d}_i$, and the current distance of the particle from the i -th anchor, $\|x_p(t) - x_i\|$.

$$\omega_p(t) = \omega_p(t-1) \exp\left(-\frac{1}{2} \sum_{i=1}^N (\|x_p(t) - x_i\| - \hat{d}_i)^2\right) \quad (5.5)$$

Then, we check the weight distribution of the particles in order to avoid critical situations, which arise, for example, when all particles are trapped within a room while the client node had moved outside. This is done by verifying:

$$\sum_{p=1}^M \omega_p(t) > \tau_1 \quad (5.6)$$

where τ_1 is a threshold value that we set equal to 10^{-5} in our experiments. If the test fails, particles are not correctly tracking the client node. Thus, the PF is re-initialized as described below. Otherwise, the particle weights are normalized such that $\sum_{p=1}^M \omega_p(t) = 1$ and the position of the client node is determined based on formula 5.3. As customary when working with PFs, a resampling step

is performed if the weight distribution is severely skewed, such that there are just a few particles with non-negligible weight. Here, we adopted the SIR (sequential importance resampling) algorithm. Resampling is performed if the following condition is verified:

$$\frac{1}{\sum_{p=1}^M \omega_p(t)^2} < \tau_2 \quad (5.7)$$

where τ_2 is equal to $M/2$ in our experiments.

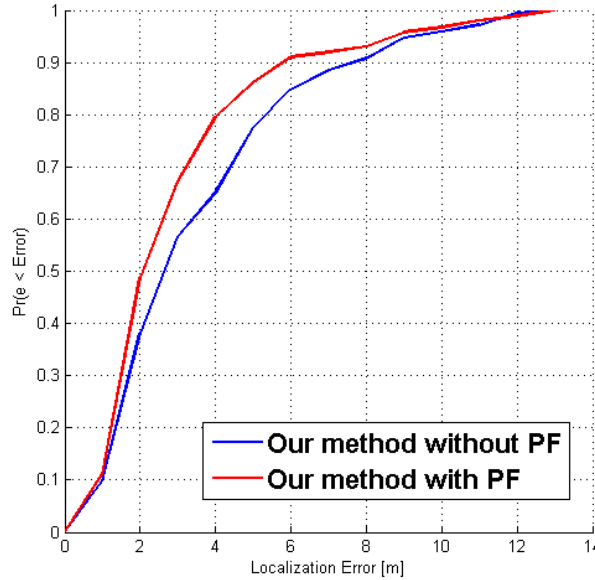


Figure 5.11: Cumulative density function of the tracking error in a supermarket area of approximately $100m^2$ with 7 AP.

3) *Initialization*: At system startup, when the first set of RSSI measurements are collected for a client node, its location $\hat{x}(0)$ is computed based on our method described in previous section.

Since the initialization of the particle filter might potentially locate the target outside the building, we leverage the knowledge of the geometry of the environment to constrain the initialization point to lie inside the building.

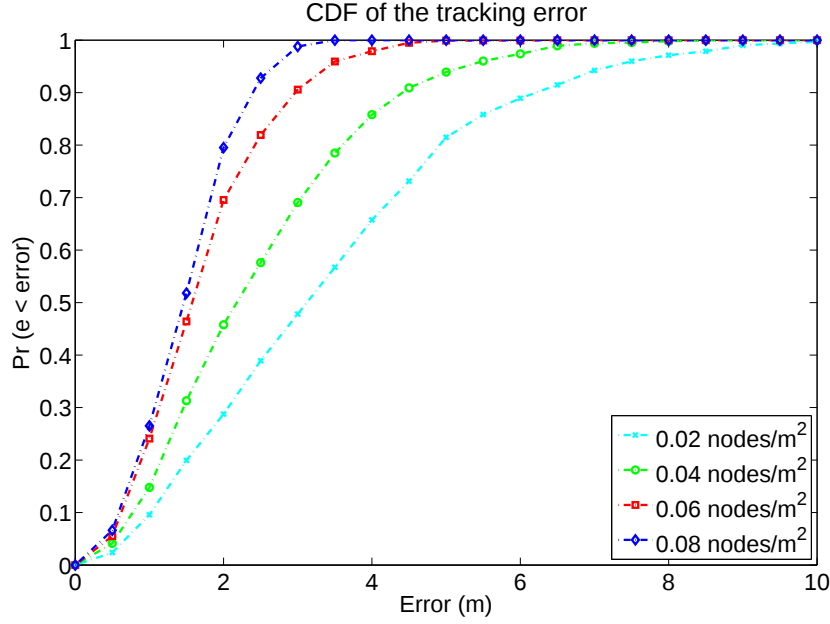


Figure 5.12: Cumulative density function of the tracking error for different values of the anchor node density in a test area of about $250m^2$

Our obtained results are shown in figure 5.11. The adoption of *Particle Filtering* has led to a good improvement in accuracy, yielding a tracking error within 2 [m] in half of the cases, and, overall, 80% of the measures were within 4[m]. Again, we can perform a comparison with 5.12: our results are just slightly worse than those of the LAURA system despite the differences explained above.

Chapter summary

In this chapter we have describe our proposed algorithmic solution for indoor WiFi localization, designed to suit fast-changing signal propagation conditions focused on dynamic tracking of customers in supermarkets. It combines the benefits of simulating empirical propagation models in terms of lowering the offline stage effort with the unmatched precision of fingerprinting techniques, merging the two approaches together.

In particular, through this chapter, we have seen how the best approach in literature Fingerprinting alone is not applicable for our scenario and how it could be effectively replaced by less demanding techniques such as Path-Loss simulators if properly tuned and used in conjunction with other techniques.

Moreover, in dynamic localization, we'll show how it is possible to improve it further with the adoption of a *Particle Filter* in order to incorporate the knowledge of cinematic models of the moving target.

We will also provide a comparative analysis of our method with respect to similar approaches, presenting the obtained results both in static and dynamic localization efforts.

In the next chapter will be described our WhAC system architecture to implement dynamic localization in supermarkets while providing at the same time value-added services to customers.

Chapter 6

WhAC: the software architecture

In this chapter we present the software architecture of our WhAC (*Where Are my Customers*) system. Our application part is closely linked to the localization techniques described in the previous chapter as it uses the information about the location of customers into the store to provide services. On the other hand, the application part is independent from the localization algorithm. This means that even changing the algorithms used for localization, the software infrastructure will still work. After giving a description of the problem, we will identify the actors of the system and all possible scenarios through the use of UML diagrams. Then we will present the design of our database used to support the application layer, and we will describe the concept of the *map of the environment* that will reside on the server. Finally, we are going to describe the client-side application and how it interacts with a service that resides server-side and uses the location informations.

6.1 Problem Description

6.1.1 General Description

As said in Chapter 4, nowadays *department stores* have complex profiling and loyalty programs and many players in this sector are configuring themselves as mobile phone virtual operators. The next frontier in this field is to extend the profiling techniques to the point of tracking the movement of individual customers

within the exhibition spaces, and to perform customized offers tailored to the single customer in real-time. Our idea is based on tracking of clients over the network WiFi in the exhibition area, on the provision of additional services to the client, and on pushing messages and advertising on the user's terminal.

6.1.2 Detailed Description

Our system is composed of three parts:

The first is the **application server** that manages the map of the supermarket. It is accessible by an *administrator* that can do operations such as: change the position of element that compose the supermarket, add new products, manage the profile of the customer and many other operations that will be explained in the next sections.

The second part is the **client application** developed for a mobile terminal. This application interacts with a service (that resides on server-side) and allows the client to use all services offered by the supermarket: search for a product, synchronized shopping list, load the supermarket map etc. Moreover the server service will hold tracking informations of movement of the users for statistical analysis purpose.

The last part is the **web site**, that allow the clients to do operations such as create a new personal profile, see the supermarket offers and pre-compile a shopping list that will be synchronized to his mobile device once that the client is in the supermarket.

6.1.3 Glossary

In the last part of this section we present a glossary that will be useful to explain the architecture of our system:

- **POS:** the "Point of sale" is a device that allows to accept payments via credit card or debit card.
- **Large-Scale-Distribution:** the "Large-Scale Distribution" is the evolution of business from retail and wholesale level. It consists of large structures or

large groups with many facilities covering national and international scale. In Italy, some examples are: Carrefour, Esselunga, Coop, etc.

- **Loyalty card:** is a personal card that links the customer at the supermarket are designed to offer its services as collection points and collection of prizes.
- **Mobile device:** is the device that may be available to the user and allows him to be tracked inside the store and that allows him to use the services provided.
- **Map:** is the logical representation of the supermarket. With the map we can identify where users are located and their movements.
- **Cell:** is the minimum boundary by which the user position is determined. The map is represented by a grid that contains a set of cells.

6.2 Actors of the system

In this section we talk about the users of the system and their role, the actions that they can perform and their privileges:

- **Administrator:** The system administrator is a super-user who can manage and update the supermarket map and the database of the products. He logs in to the system with an username and password and he can perform these actions:
 - Update the supermarket map: he can insert or delete new elements such as access point, shelves, change their position, associate one or more products to the shelves and more operations that will be explained with the help of *use cases diagrams*.
 - Update the products database: he can insert or delete new products, insert new supermarket offers.
 - Manage the client profiles: he can insert, change or delete profiles of the clients of the supermarket that are the final users of the system.

- **Client:** He's the final user of the system. The client is registered in the system. He has an account that allows him to use all the functionalities provided by the supermarket system. The information that the system store for each client are:
 - Name
 - Surname
 - Category
 - Username
 - Password

The client log in to the system with a mobile device inserting his username and password and he can perform these operations:

- Consult his historical profile
- Search for a product
- Receive advertising and / or personalized alerts
- Communicate his position to the server (this is not done by the client, but automatically by the mobile device)
- Request the current position inside the supermarket and show the supermarket map on his device.

In addition, an user can log in to the web site of the supermarket, and, after a registration, he can pre-compile a shopping list that will be synchronized on device.

All these operations will be explained in the next section, where we will describe all the possible scenarios.

6.3 Use case and scenarios

Now we identify the use cases and all possible types of scenario for each actor. A scenario represents an action that an actor perform while he's using the system. A

typical scenario is represented as:

- Name of scenario
- Actors that are involved in that scenario
- Pre-conditions that have been satisfied
- Operations that the scenario includes
- Post-conditions verified after the scenario is finished

Each scenario is represented graphically via UML: one sequence diagram and one activity diagram. The identification of the scenario is very important because it defines the actions that our actors can perform during the usage of the software. The scenarios will be used in the final part of the development of the system where we will talk about the test of software. We will test each scenario and we'll see if our software correspond to what we have suppose in these cases.

6.3.1 System Administrator

We now present the possible scenarios where the *System Administrator* can act. We assume that there is already a list of administrators registered to the system, because an administrator is a super user (so he as all privileges).

1. **Log In:** an Administrator Logs In to the system with username and password. From this moment on he can access the supermarket map, user profile, products, offer categories etc.
2. **Manage Supermarket Map:** an Administrator can update the supermarket map (that we suppose already created) by performing these actions:
 - Create, delete or update the cells that compose the supermarket map. It corresponds to changing the structure of the supermarket.
 - Create, delete or change the position of Access Point in the supermarket map.

- Create, delete or change position of the shelves in the supermarket map.
 - Associate one or more products to a shelf. Administrator can move the products from one shelf to another, or put on products in the shelf if the shelf is empty.
3. **Manage User Profile:** Administrator can add, delete or update informations about a client of supermarket.
 4. **Manage Products:** Administrator can add new products to the list of the supermarket products and create new offers associated to these products.

6.3.2 Client

Client Via Web Site

We present the possible scenarios of what a client can perform when he connects to the web site of supermarket:

1. **Create a new account:** a client that is not registered to the supermarket system can create a new personal profile.
2. **Log In:** a client logs in to the system with his username and password. He can access to all service provided by the supermarket.
3. **Manage his personal profile:** an user can change and update his personal informations.
4. **Compile the shopping list:** a client can see the supermarket products with their price and select some products, compiling a shopping list that will be loaded on the mobile device once he will enter in the supermarket.
5. **View the weekly offers:** supermarket will show the weekly offers that can help the client to compile his shopping list.

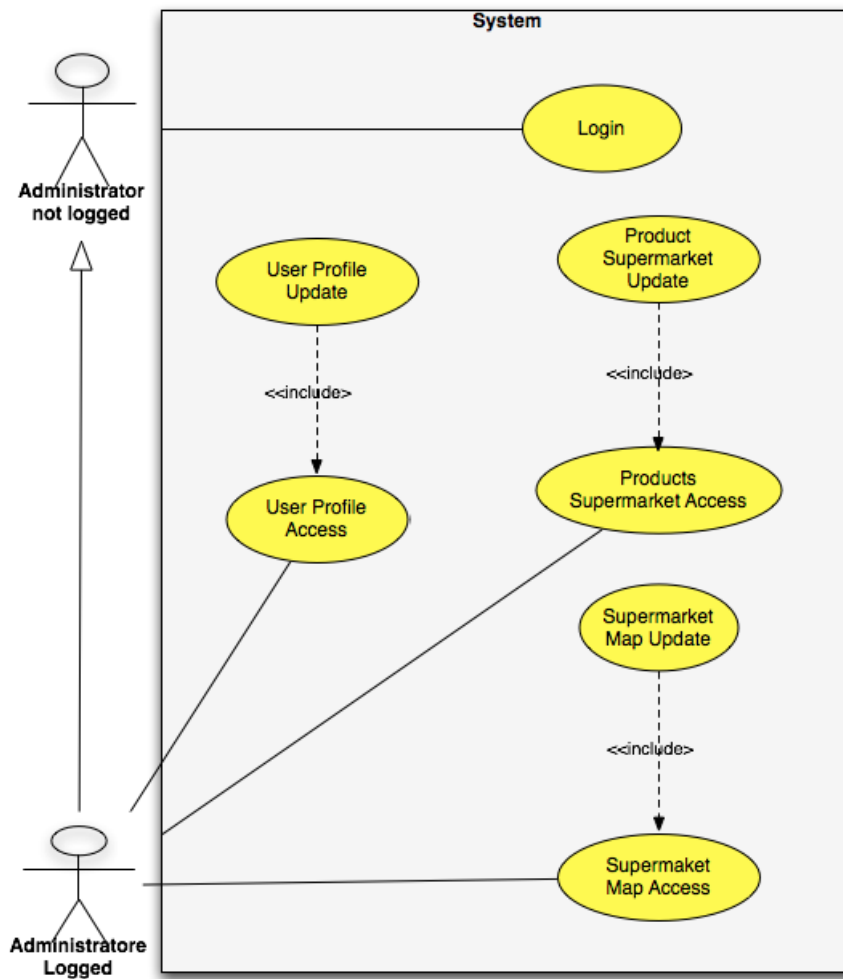


Figure 6.1: Use Case: Administrator

Client *in loco*

We present the possible scenarios about the Client when he's inside the supermarket and he uses his mobile device to access to supermarket service.

1. **Log In with a mobile device:** the client turns on his mobile device and logs in to the system inserting his username and password. Once logged, he can access to the supermarket application.
2. **Synchronize shopping list:** the client (after the login) synchronizes the

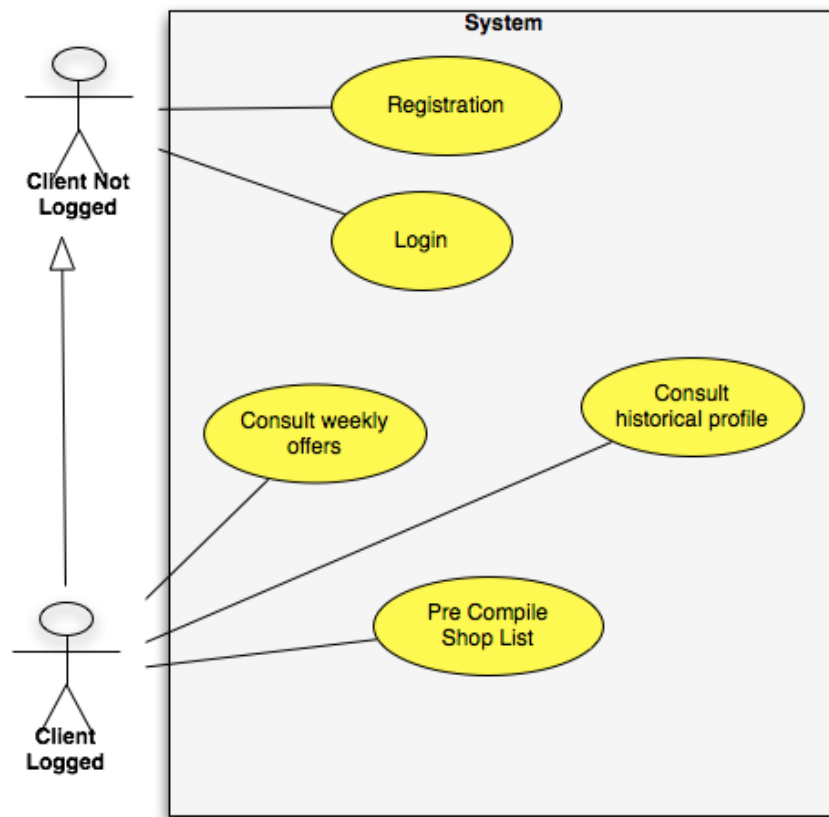


Figure 6.2: Use Case: Client Web Site

shopping list that he has previously compiled from the web site. The mobile terminal displays the shopping list.

3. **Add a new product to the shopping list:** the client add to the cart a new product.
4. **Search for a product:** the client search the position of a product inside the supermarket. The client inserts the name of the product to search and the system return its position (map of supermarket or number of shelf where the desired product is).
5. **Get Current Position:** the client requires his current position inside the supermarket. On the display of the mobile terminal is showed the supermarket

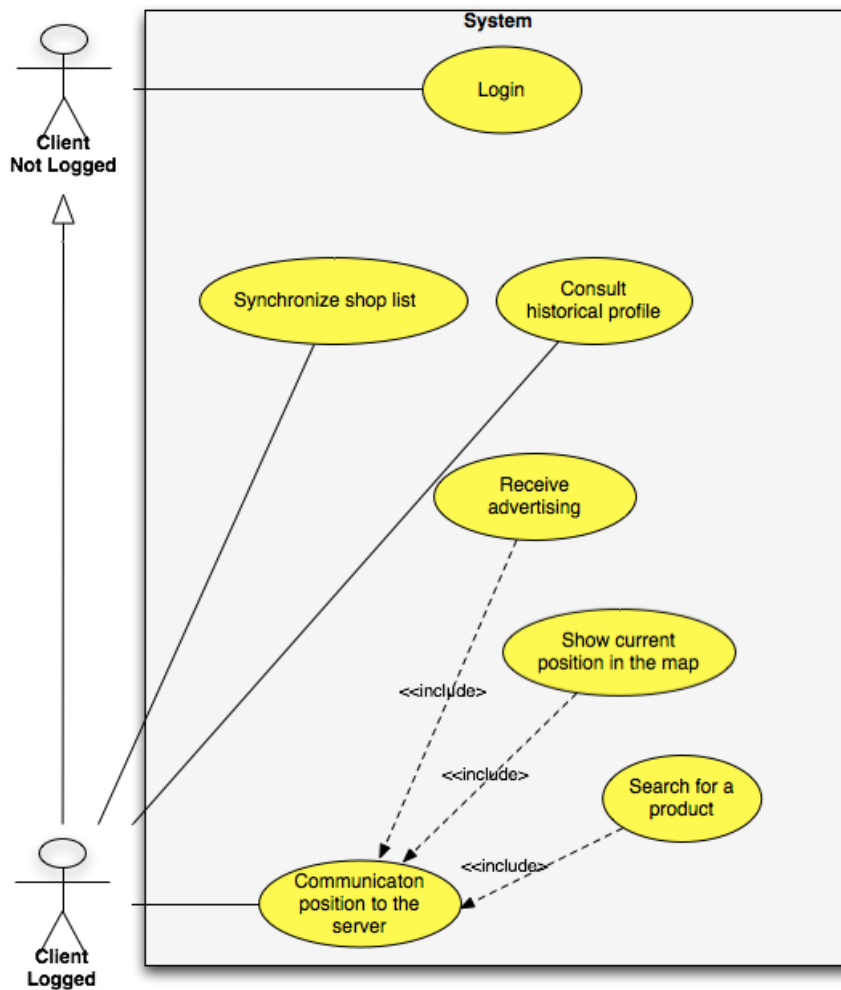


Figure 6.3: Use Case: Client into the supermarket

map with the current position of the client.

6. **Receive advertising and / or personalized alerts:** the system sends advertising and personalized alerts about products and offers based on both the position of the client inside the supermarket and on his profile.
7. **Log out:** after the client has finished shopping, he logs out of the system and turns off his mobile device.

We present an example of a scenario described in this section:

Search for a product:

- Use case name: Search for a product
- Actor: Client
- Pre-condition: Client is doing shopping and he wants to know the position of a product.
- Operations:
 1. the Client opens the application and performs the login.
 2. the Client inserts the name of product to search.
 3. the Application Server queries the map database and searches the position of the product.
 4. the Application Server sends the position of the product to the mobile device.
 5. On the display of the mobile device is showed the position of the product (number of shelf or map)
- Post-condition: the client knows the position of the product.

In this section we have provided a general description of the system, we have identified the actors of the system and all the possible actions that they can perform. All the scenarios that we have identified will be tested in the test phase.

In the next section we will describe the design of the application. We will explain how we have designed the supermarket map and we'll illustrate the design of the database on which our application is based.

6.4 System Architecture

The supermarket system has been developed as application based on the J2EE platform [30], the architecture will be multi-tiered, spawning four levels:

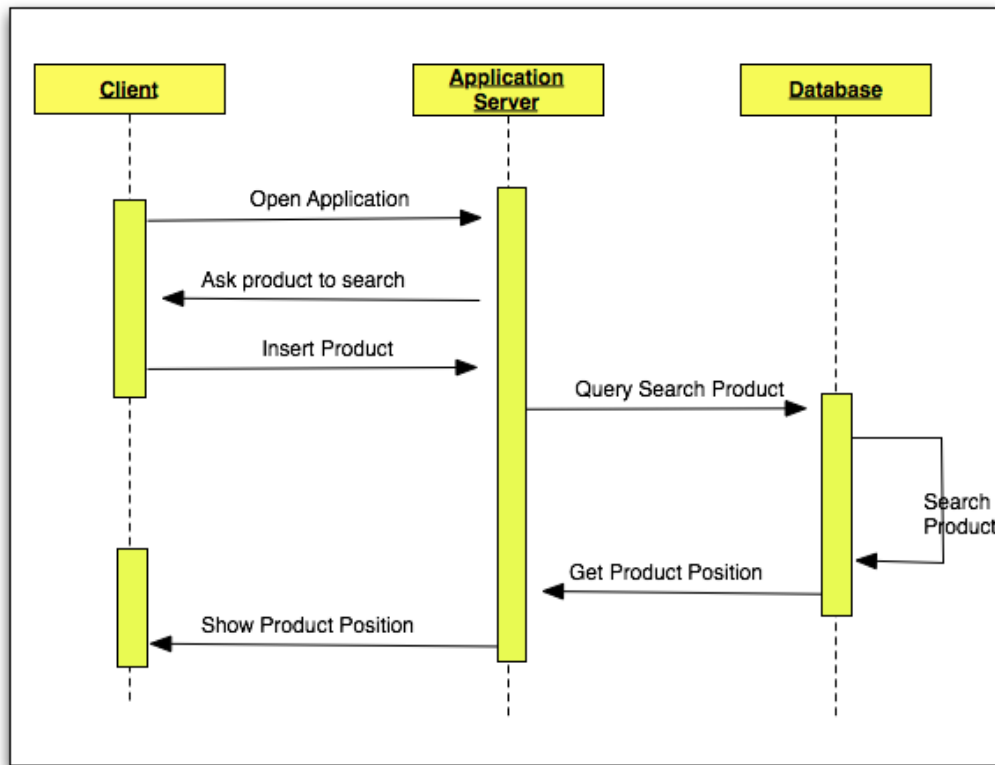
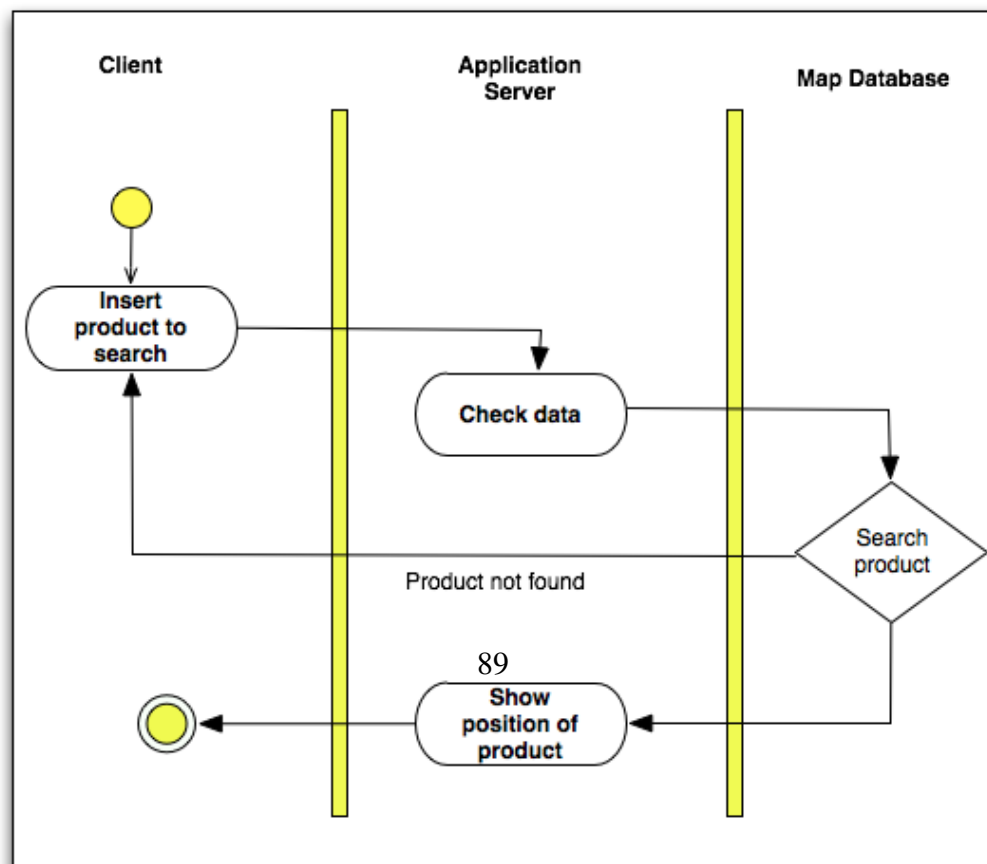


Figure 6.4: Client Sequence Diagram



- Client Tier: this is the system that interacts with the actors. It consists of independent elements that provide the four services. In all cases it is implemented with a Java Swing interface accessible to the administrator.
- Web Tier: it consists of web components (servlets and JSP). It handles the HTTP requests from the browser, interact with the Business Tier to operate on the system datas, and generates response pages for the user.
- Business Tier: it consists of Enterprise Java Bean (EJB) and contains the data model and business logic. Enables the creation, querying, editing and deletion of data.
- Data Tier: it consists of a relational database, which ensures the persistence of the data model.

6.4.1 Architectural Details

Our solution offers the business and the web tier implemented on a Java EE 5 application server [30]. For this purpose we have chosen to use JBoss Application Server 5.0 [31], with the data tier implemented on the RDBMS MySQL 5.1 [32]. In details:

- The data are described by the Java Persistence API as specified in *EJB 3.0 class "entity" Java* and mapped to relational database by mean of the object-relational-mapping provided by JBoss Hibernate.
- The business logic is implemented as a *Stateless Session Bean*.
- The web tier uses the remote interfaces of the Session Bean and the exchange of datas is done through a regular JavaBean. Although the solution provides the *web-tier* and the *business tier* on the same server, it was decided to completely separate out the two parts in anticipation of further developments of the system.

6.5 Supermarket Map

In this section we present how we have designed the supermarket map and how we have implemented it in our *map database*.

The Map is a logical representation of our supermarket. It's divided into a grid of non overlapping square cells. The cell is the minimum entity based on which we know the customer's position, so it is the maximum level of detail that can be achieved. A cell has a own unique identification code and it's localized by mean of the coordinates of its 4 vertices compared to a reference system in two-dimensional terms in abscissa / ordinate on the map. In our tests, we set a cell side length of 4 meters, but it may be changed according to topology needs.

We have different types of cell, as in figure 6.6:

- **Entry Cell:** The cell located at the entrance of the supermarket (marked in violet) is one to which you link customers just entered the shop.
- **Exit Cells:** They are the cells that contain the cashes (marked in red) or exit without buying (marked in blue).
- **Classic Cells:** They are the normal Cell of the supermarket, they can contain shelves carrying the products.

Inside the cell there may be from 0 to N shelves. Each shelf has a unique identification code and contains products of the same type. The shelf is the minimum granularity with which is known the position of a product.

6.5.1 Database Design

The database should allow the company to analyze and to manage in real time the situation regarding the presence and movement of customers in the supermarket. In particular, the database should allow to link the provision of products with the location and profile of individual customers. To this end, the database must provide a complete picture of the supply of products on display: they must be known that information about trade name, category, position on the shelves and special

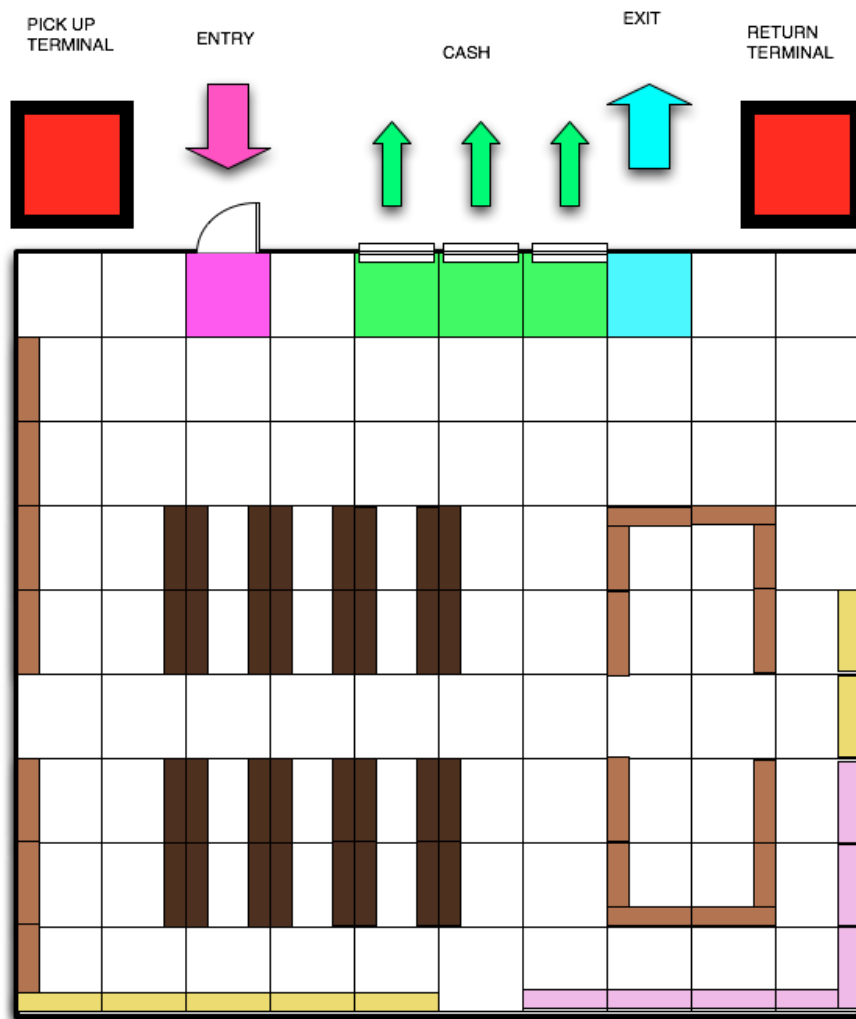


Figure 6.6: Supermarket Map

offers. Similarly, it must provide summary information relating to clients in the supermarket: we need to know the precise position in real-time of the customers, tracking their movement, a customer identifier, the presence of a "shopping list" precompiled by the client, together with the knowledge of what the customer purchases step by step while shopping, and finally a series of short-listed bids for the customer.

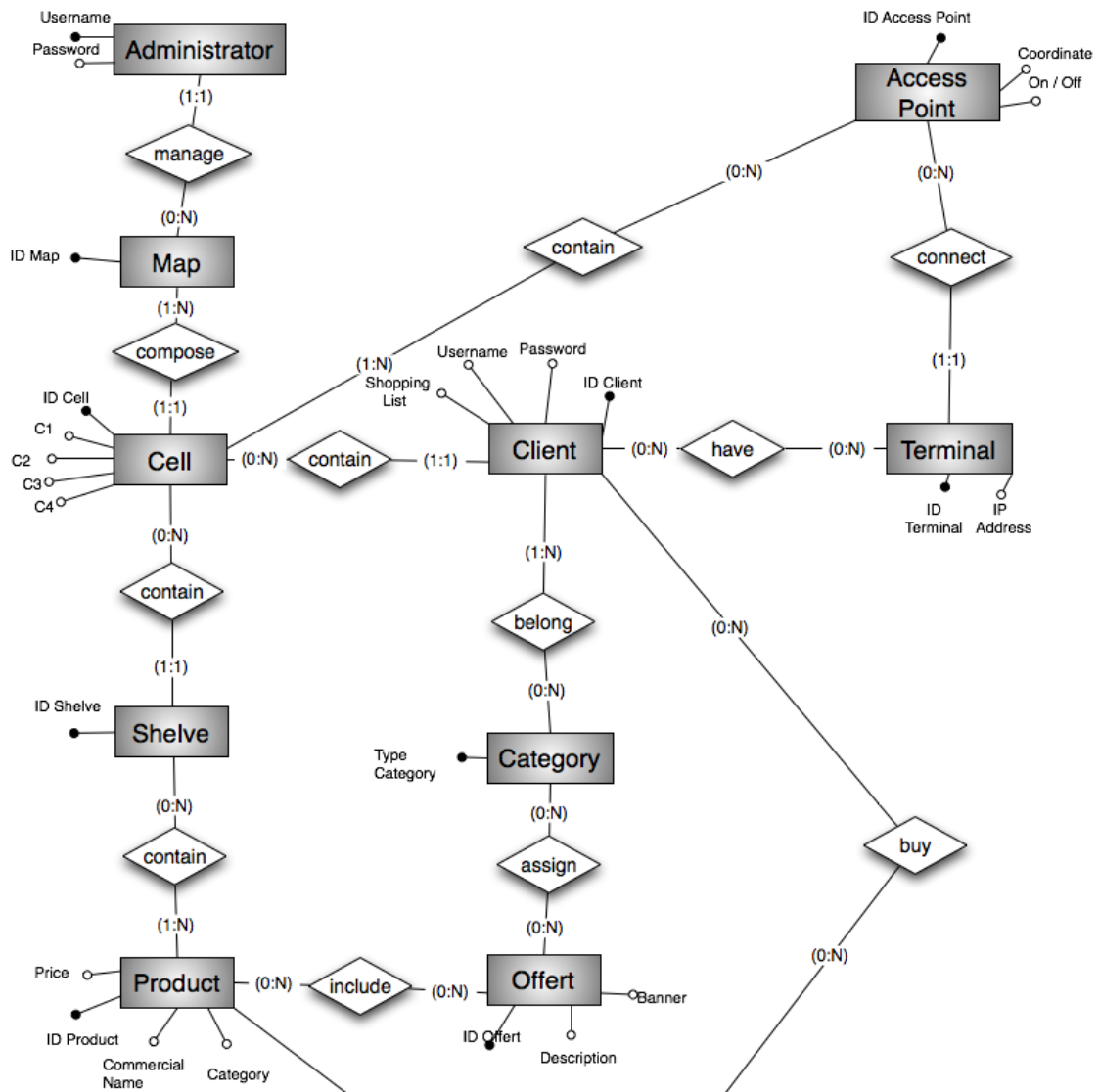


Figure 6.7: Design of Database Map

6.6 Client Application

6.6.1 Server Service

The Server Service is the part of Server that manage all the connections of the customers that use the application and offers services based on the position of the clients.

Main Process "Attesa"

It consist of a main process called *Attesa.java* that accepts all connections that arrives from the mobile device via Socket. *Attesa.java* establish a connection with the mobile device of the client. The client inserts his username and password, the service check the correctness of the data and, if they are correct, launches a thread called *ManageClientConnection.java* that manages the single connection with the client. Then the Process *Attesa.java* returns to listen the channel and accept other new connection requests that arrive from other customers.

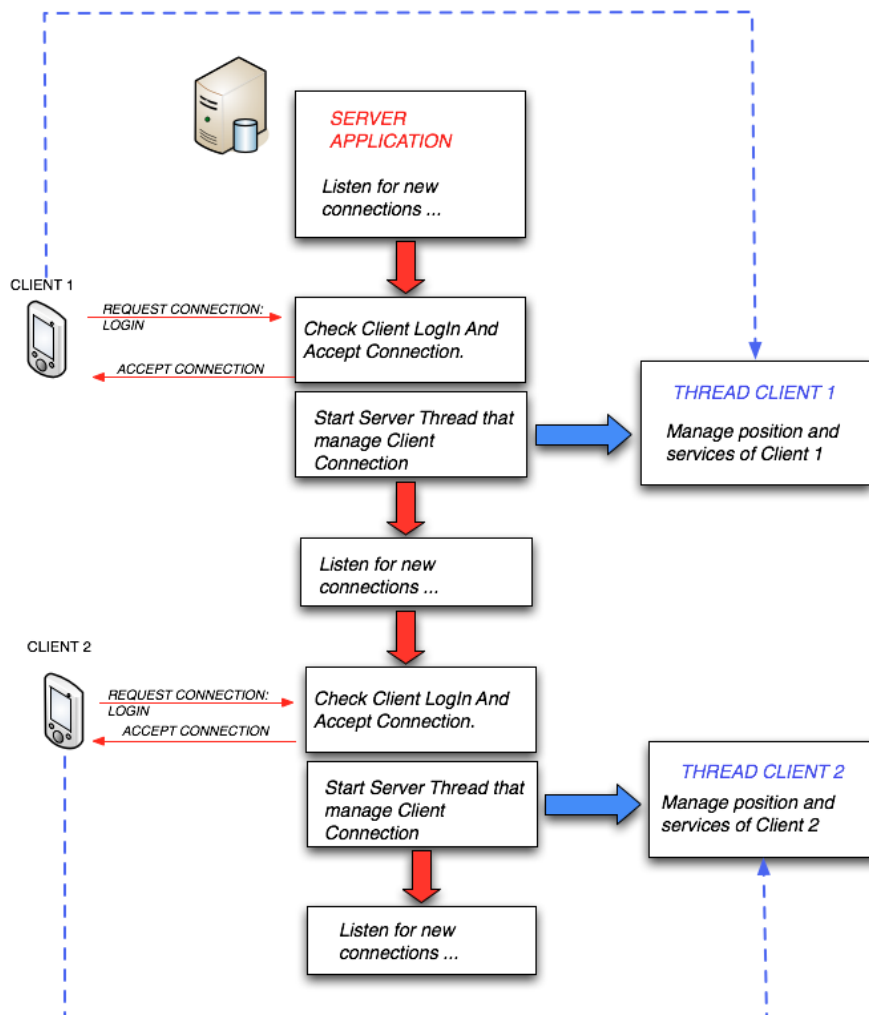


Figure 6.8: Design of Server Service

Manage Client Connection Once a client has performed the login, his connection is passed to a thread called *ManageClientConnection.java*. We can have more than one thread managing the client sessions: for example, if 10 customers are connected at the same time, we will have 10 threads, everyone managing a single client. *ManageClientConnectio.java* manages two different parts that are very related one to the other: the first one manages and updates the position of the client, the second one offers the personal services to the customer. It is obvious, that the second part need to know informations of the first one in order to work.

Update Customer Position Every few seconds a client device sends its RSSI vector to the server. This vector (as explained in the previous chapter) contains the RSSI reading received by each Access point. The Server receives these values and with our algorithm, described in the previous chapter, calculates the actual position of the client inside the supermarket. The Client position is updated inside the database writing the new position of the client (or the same if the client is determined to have remained in the same cell as before).

Moreover, the *ManageClientThread* is connected to a *JavaFrame* called *Frame-Position.java* that represents the Map of the supermarket. In this representation, the map is composed of a grid of empty cells, than is different from the representation that we have described in previous section. There are no shelves, access point or cashes represented. The goal of this map is to describe the movements of the clients. In the *FrameMap* is update in real time with the new position of the client and it colors the cell in which the client is located at the moment according to the RSSI readings. Also, these is a console that gets information about client position. For each new position is the cell in which the customer is, showing also the shelves and products in the same cell.

At the end, the server stores the *Log File* with the position of the client. For each Client, the server has a log file that indicates how many time that Client has spent in each Cell. Moreover, we have a log file for each cell that indicates what is the affluence in that cell, i.e. how many customers have been in that cell and how much time they have spent there. All these informations about the tracking of the users can be useful to provide statistical models, see what are the

most crowded cell, identify the locations of customers etc. They can be helpful to improve system localization.

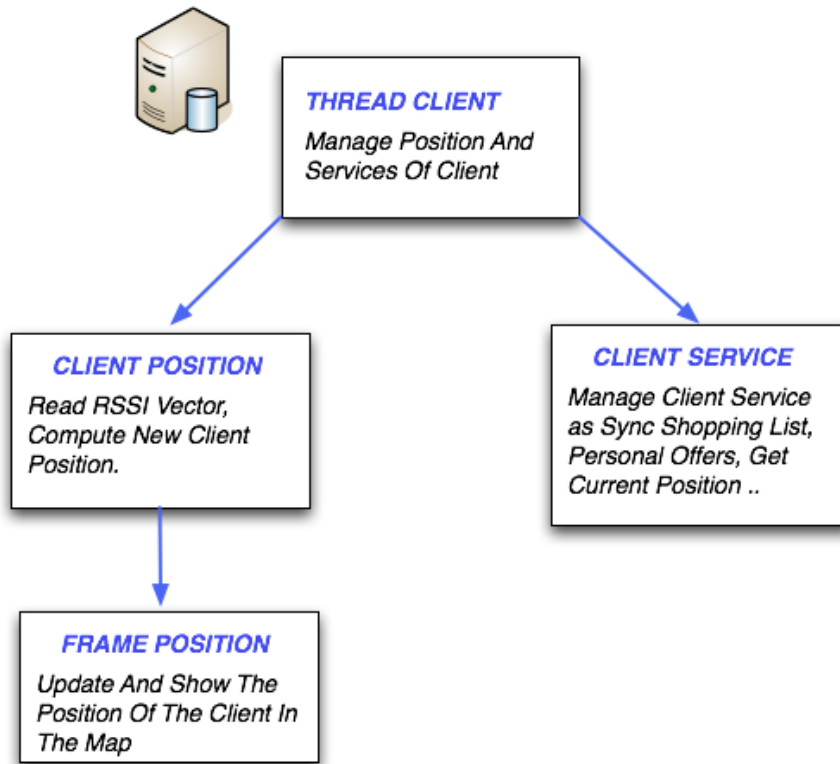


Figure 6.9: Functionalities of Server Services

GetService The second part of service Server is related to the first one and offers the services to client. In order to work, it needs informations about the client position. Based on the commands received via socket, the service can perform thiese task:

- Load the shopping list that the client has precompiled at home.
- Load the map on the mobile device with the current position of the client.
- Send real time advertisements based on the client position. For example, if the client is in front of the shelf containing pasta, the server may send a

message with an offer related to pasta. On the client terminal will appear a notification.

- Show (and add) products on shelves near to the client position. Knowing the current position of the client server service sends to the customer the products that are in front of him in that moment. So client can add one or more product to his shopping list.
- Finish the shopping: when a customer finishes his shop, the server service calculates an estimate of the time of queue in each cash, so the client can decide at which cash is convenient to go to pay. The computation of the wait time uses the localization informations and it is done in the following way: it is counted the numbers of customers that queueing (count the customer that are in the cell of the cash). For each customer we know how many products he has in the cart, and the expected waiting time is also directly proportional to the number of items in the trolley. We compute the time for each client and we sum the time of all customers in that cell.

6.6.2 Client App

We have decide to develop the application in Android environment [33]. For the building of the application we decided to use the Eclipse IDE [34], in which is possible to include the plugin for the development of Android applications. The Android version used is the 2.1. The final application has been tested on a Samsung Galaxy MINI smartphone [28].

The application is composed by two different and decoupled parts:

Thread sending power signal

The first part of the application is a thread that collect the RSSI readings. The mobile device reads the power trasmitted from each Access Point. The RSSI vector is sent via socket to the server service that computes the readings and calculates the position of the client inside the supermarket. The client doesn't interact with

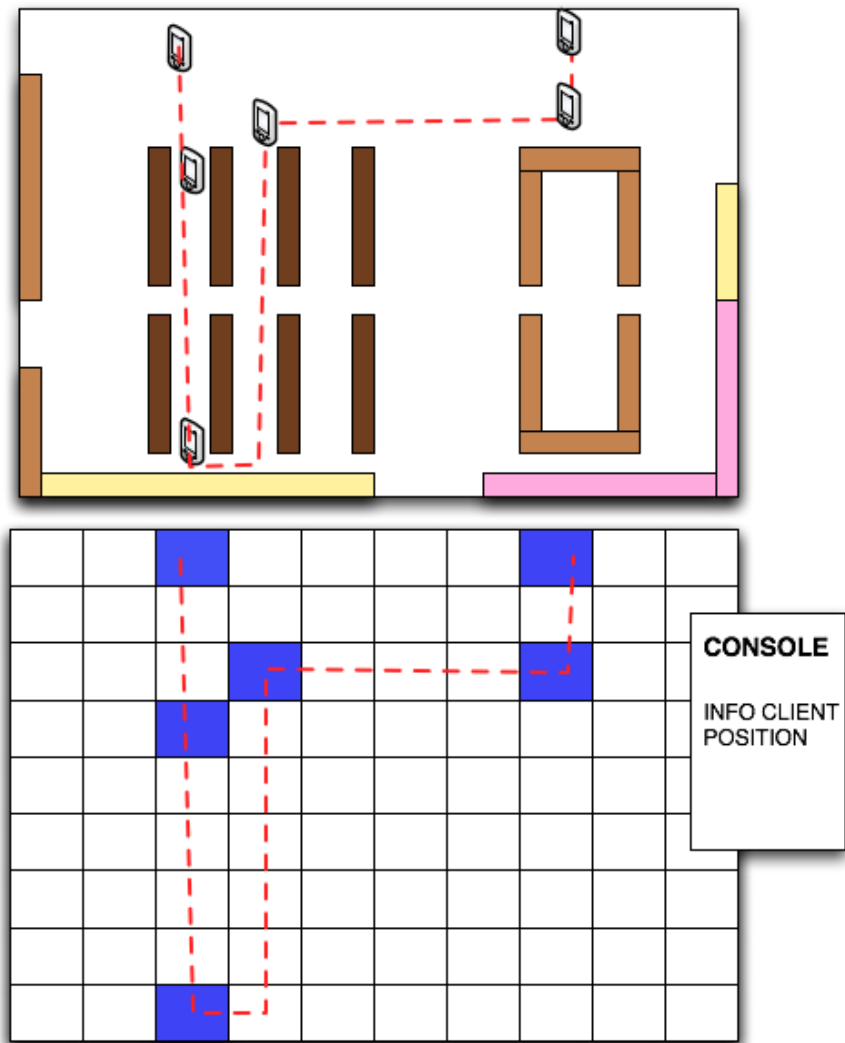


Figure 6.10: Example of the tracking of a customer

this part, that is implemented as a separated thread, independent of the rest of application.

6.6.3 Service Client

The second part of the application is the part that interacts directly with the clients. After the login, the client will be showed the shopping list that he has pre-compiled

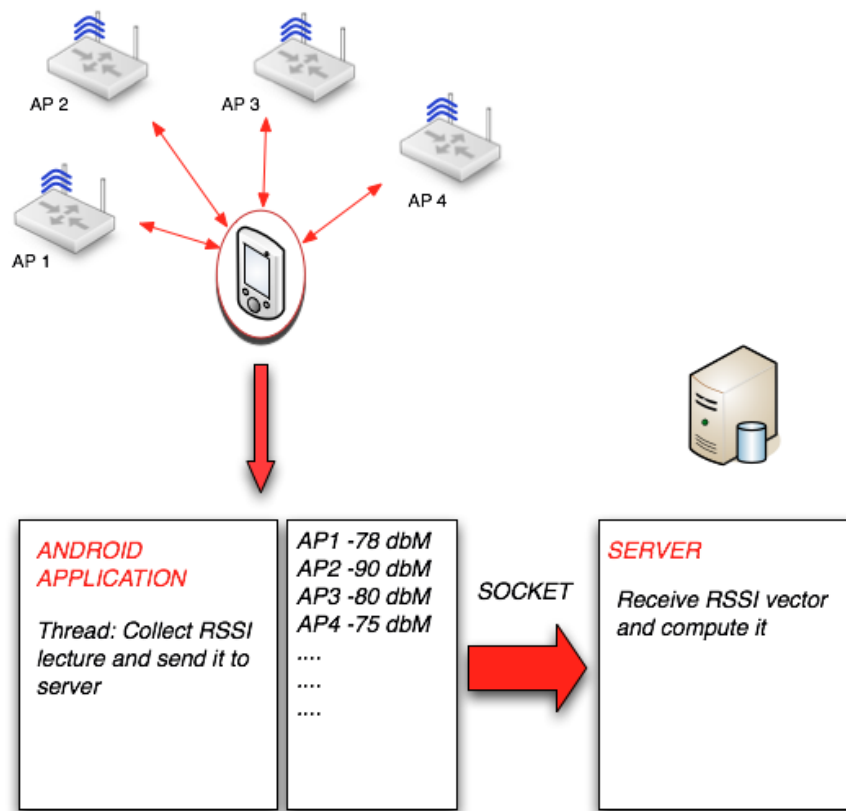


Figure 6.11: The Android device collects RSSI readings and sends them to the server

at home. Then client, during the shop can perform these actions:

- Ask his current position: on the Android device will be showed the map of supermarket divided in cells with the position of the client colored in red.
- Compile the shopping list: the client checks the products while he is doing his shopping. If there is one or more products that it isn't in the shopping list, client can add it at run-time new products. The Android device will indicate the products that are in front of him at every moment.
- Finish the shopping: once the client has finished the shopping, the application computes the total price of the goods in the cart, and indicate at the

client the wait time at each cash. The computation of the wait time is done by the server service as we have explained in the previous section.

- Receive personal advertisement in real time. When a client is in front of a one or more products that are associated at some offers, it is shown a notification on Android device that indicates the presence of a new advertisement directed to him. By opening the notification, it is shown the offer that is represented by a title, a description and an image of the product. The choice of the offers is done joining the client position and the information about client profile.



Figure 6.12: Synchronization of the shopping list on the Android device



Figure 6.13: The products located in the same cell as the customer are shown on the Android device

6.7 Test cases

The last part in the development of our software is the test case: in this part we have analyzed all the scenarios that have been identify when we have designed the system and we have verified if this scenarios are corrected under some hypothesis. An exhaustive documentation could not fit in this thesis but is provided in the design document. Each scenario has the follow structure:

- Name of scenario
- Input Data
- Expected Result
- Real Result
- Diagnosis

For each scenario we have been tested the general case, and the most critical case obtained inserting critical values in the application. To test directly the written code, we have performed some JUnit [35] testing.

Chapter summary

In this Chapter we have described and documented the WhAC application of our localization system. After identifying the actors and their role in the system with the help of *use case diagrams* and scenarios, we explained the design of the application, defining the concept of *map* and its implementation in our database. Also, we have developed a server service that interact via socket with client applications to provide services to the final customer.

In particular we have seen how the application can improve an everyday activity such as shopping. We have develop services on the application that are closely related to the localization part but the application can be improved inserting other functionalities. Moreover, it's possible to develop the mobile application for platform different from Android such as iOS or Windows CE.

Chapter 7

Conclusion and future works

This thesis has pointed out how, nowadays, indoor localization in noisy environments is still an open issue, involving a difficult compromise between accuracy and costs.

We focused in finding a solution to indoor WiFi localization for the purpose of tracking customers moving around in malls and supermarkets, in order to provide additional services, enhance profiling techniques and improve marketing effectiveness. Developing such a system has involved taking into account parameters as *user comfort* and *acceptance*, besides accuracy, scalability and installation efforts and costs. WiFi 802.11 turned out to be the most suitable option to our purpose, although many issues arisen in the project of our infrastructure. Among these, major issues are due to the strong fluctuations of the RSSI values typical of WiFi even in line of sight conditions, the impossibility of using inter-AccessPoint measures in order to estimate a propagation model, and the different gains of eterogeneous types of antennas.

Our proposed algorithm for indoor WiFi localization combines the benefits of simulating empirical propagation models in terms of lowering the offline stage effort with the unmatched precision of fingerprinting techniques, merging the two approaches together.

It is designed to suit fast-changing signal propagation conditions and to satisfy all requirements and constraints derived from noisy environments and from our

particular field through this thesis.

Moreover, in dynamic localization, we have shown how it is possible to improve it further with the adoption of a *Particle Filter* in order to incorporate the knowledge of cinematic models of the moving target in our localization algorithm.

Despite all the exposed constraints, our results yield a static localization error in the previously descipted noisy and fast changing environments of 3.5 [m] at the 80-*th* percentile.

As for dynamic localization (*tracking*) in such conditions, the adoption of *Particle Filtering* to incorporate the cinematic behaviour of the target, has led to a good improvement in accuracy, yielding a tracking error within 2 [m] in half of the cases, and, overall, 80% of the measures were within 4[m].

We designed and implemented the WhAC (*Where Are my Customers*) application infrastructure that exploits our localization algorithm in order to provide additional service based on the location of customers into the store and improve profiling and marketing techniques. However, the application part is independent from the chosen localization algorithm. This means that even changing the algorithms used for localization, the software infrastructure will still work.

Our system is composed of three parts:

The first is the *application server* that manages the map of the supermarket. This part is developed in J2EE technology. It is accessible by an *administrator* who can do operations such as: change the position of shelves that compose the supermarket, add or delete products on display, manage the profile of the customers, and many others.

The second part is the *client application*, that we have developed for the *Android* platform. This application interacts with a service (that resides server-side) and allows the clients to use all services offered by the supermarket, such as: receive custom offers, access search for a product, synchronized shopping list, load the supermarket map etc. Moreover, the server service will record tracking informations about the movement of the users for statistical and analysis purposes.

The last part is the *web site*, that allows the customers to perform operations such as create a new personal profile, see the supermarket offers and pre-compile

a shopping list that will be synchronized to his mobile device once that he arrives in the supermarket.

In the future, we plan to improve our algorithm by refining the adopted cinematic model. In particular, we can exploit datas gathered from built-in sensors such as accelerometer and compass, since these devices are commonly included in all Android mobile terminals. In particular, these datas will permit us to distinguish the RSSI fluctuations caused by environmental changes from fluctuations ascribable to actual target movements, thus increasing the overall accuracy and precision of the dynamic localization .

Another improvement of our localization architecture could be the extention of the simulator capabilities with the introduction of a customizable propagation attenuation parameter for each different product among those on display in the supermarket. In this way, we can greatly speed-up the building and calibration phase of the map of the environment used by the simulator, and, whatismore, refine the overall simulator predictions. Further amends in localization accuracy would come by combining the knowledge of the actual number of products of each type on display (thanks to the knowledge of what customers put in their trolleys) with their attenuation factor.

Bibliography

- [1] B. Schilit, N. Adams, and R. Want. "Context-aware computing applications". IEEE Workshop on Mobile Computing Systems and Applications (WM-CSA'94), Santa Cruz, CA, US. pp. 89101.
- [2] Jorgen Kemper, Holger Linde, *Challenges of Passive Infrared Indoor Localization*, Proceedings of the 5th Workshop on Positioning, Navigation and Communication (WPNC08), 2008
- [3] Michael Tuchler, Volker Schwarz, Alexander Huber, *Location accuracy of an UWB localization system in a multi-path environment* IEEE International Conference on Ultra-Wideband, Zurich, September 2005
- [4] Y. Huang, J. Benesty, *Audio Signal Processing for Next Generation Multimedia Communication Systems*, Kluwer Academic Publishers, 2004
- [5] Gadjah Mada, Resolving the Fingerprinting Problem: Comparison of Propagation Modelling and Machine Learning Approach, 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)
- [6] Ching-Hsien Hsu, Chia-Hao Yu, *An Accelerometer Based Approach for Indoor Localization*, Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 2009
- [7] Nathaniel Bowditch, *The american practical navigator*, National Imagery and Mapping agency, Maryland, 1995, chapter 7

BIBLIOGRAPHY

- [8] Martin Azizyan, Ionut Constandache, Romit Roy Choudhury, *Surround-Sense: Mobile Phone Localization via Ambience Fingerprinting*, MobiCom09, September 2009, Beijing, China.
- [9] Official IEEE 802.11 working group, <http://www.ieee802.org/11/>
- [10] Remigiusz Olejnik, *A Floor Description Language as a Tool in the Process of Wireless Network Design* Springer Berlin Heidelberg, 2009, isbn 978-3-642-02671-3
- [11] Hyuk Lim, Lu-Chuan Kung, Jennifer C. Hou, Haiyun Luo, *Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure* Published online: 4 October 2008, Springer Science+Business Media, LLC 2008
- [12] Chen Feng, Wain Sy Anthea Au, Shahrokh Valaee, and Zhenhui Tan, *Compressive Sensing Based Positioning Using RSS of WLAN Access Points*, IEEE INFOCOM 2010 proceedings
- [13] Youngjune Gwon, Ravi Jain, and Toshiro Kawahara, *Robust Indoor Location Estimation of Stationary and Mobile Users*, IEEE INFOCOM 2004
- [14] Przemyslaw Madej *3D Wireless Networks Simulator visualization of Radio Frequency propagation for WLANs* (2006) A dissertation submitted to the University of Dublin, Trinity College
- [15] Lilian Aveneau, Emilie Masson, Pierre Combeau, *RaPSor: a Radio Propagation Simulator*, <http://rapsor.sourceforge.net/>.
- [16] Sharp, Byron and Anne Sharp (1997), *Loyalty Programs and Their Impact on Repeat-Purchase Loyalty Patterns*, International Journal of Research in Marketing, 14 (5), 473-86.
- [17] Gajanayake, R., Gajanayake, S., Surangi, *The impact of selected visual merchandising techniques on patronage intentions in supermarkets*, International Conference on Business and Economic Research, H 2011 p1130-1165

- [18] Browne, Karen (April 2010). *Trolley Psychology: Choice unlocks the psychological secrets of the supermarket and shows you how to avoid spending more than you mean to*. Australian Consumer's Association Choice Magazine (4): p60.
- [19] Andreas Fink, Helmut Beikirch, *RSSI-based Indoor Localization using Antenna Diversity and Plausibility Filter*, Proceedings of the 6th Workshop on Positioning, Navigation and Communication 2009 (WPNC09)
- [20] You Zheng, Oumaya Baala, Alexandre Caminada, *Optimization Model for an Indoor WLAN-based Positioning System*, 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)
- [21] V.P.C. Dassanayaka, Ajith Ukwatte, Department of Mechanical Engineering, University of Moratuwa, *Development of an Automated Guided Vehicle for use in Industrial Environments*,
<http://192.248.11.79/projects2010/group19/>
- [22] Alessandro Redondi, Marco Tagliasacchi, Matteo Cesana, Luca Borsani, Paula Tarrío, Fabio Salice, *LAURA - LocAlization and Ubiquitous monitoRing of pAtients for health care support*,
<http://laura.como.polimi.it/index.php?Resources>
- [23] Youngjune Gwon, Ravi Jain, and Toshiro Kawahara, *Robust Indoor Location Estimation of Stationary and Mobile Users* IEEE INFOCOM 2004
- [24] Paramvir Bahl, Venkata N. Padmanabhan, *RADAR: An In-Building RF-based User Location and Tracking System*, IEEE INFOCOM 2000
- [25] P. Prasithsangaree, P. Krishnamurthy, P.K. Chrysanthis, *On Indoor Position Location With Wireless LANS*, IEEE PIMRC 2002
- [26] Winplanner optimization tool for wireless networks,
<http://www.aws.cit.ie/ubiaware/>

BIBLIOGRAPHY

- [27] OpenWRT: A GNU/Linux based firmware program for embedded devices such as residential gateways and routers, <https://openwrt.org/>
- [28] Samsung Galaxy Mini GT5800 Datasheet, <http://www.samsung.com/>
- [29] Samsung Galaxy TAB Datasheet, <http://www.samsung.com/>
- [30] Java Platform Enterprise Edition (Java EE) 5: the industry standard for enterprise Java computing.
<http://www.oracle.com/technetwork/java/javaee/overview/index.html>
- [31] JBoss Application Server, <http://www.jboss.org/>
- [32] MySQL open source database, <http://www.mysql.com/>
- [33] Android operating system, <http://www.android.com/>
- [34] Eclipse IDE for Java EE Developers, <http://www.eclipse.org/>
- [35] JUnit testing framework, <http://www.junit.org/>