Universitat Politècnica de Catalunya

Departament d'Enginyeria Telemàtica

Contributions to TOA-based location with WLAN

PhD. Thesis

Marc Ciurana Adell

Advisor: Francisco Barceló Arroyo

April 2010

Agreements

Quan el mes de març del 2005 començava la meva aventura en el món de la recerca al departament de Telemàtica de la UPC no podia imaginar que l'experiència em resultaria tan enriquidora i apassionant. No va ser fàcil emprendre aquell camí, però ara si miro enrera tinc del tot clar que aquella va ser una de les millors decisions que mai he pres. Durant aquests gairebé cinc anys de doctorat he tingut la sort de conèixer persones de les quals he pogut aprendre molt, he pogut viure molts moments que m'han fet sentir increïblement afortunat, he pogut créixer com a persona alhora que explorava nous horitzons... La veritat, però, és que tot això no hagués estat possible sense l'ajuda i el suport de les persones que han cregut en mi al llarg d'aquests anys. Des d'aquestes línies vull donar-los sincerament les gràcies. D'entre aquestes persones voldria destacar especialment el doctor Francisco Barceló, la persona que em va donar l'oportunitat d'enfilar el camí i que m'ha esperonat respectant sempre les meves decisions i punts de vista.

M'agradaria dedicar aquesta tesi doctoral a les persones que han estat al meu costat en tot moment, que senten les meves alegries com si fossin ben seves..., en definitiva a aquelles persones que m'estimen. Per últim, però per a mi molt important, el record més tendre i intens per a les persones que sé que haguessin volgut viure aquest moment amb mi i que desgraciadament no ho han pogut fer perquè ens van deixar; us porto sempre amb mi.

Abstract

Location techniques that satisfy the requirements of advanced Location-Based Services (LBS) in environments where GPS fails are needed, therefore accurate indoor positioning is becoming increasingly important. This PhD Thesis is devoted to the research on location of mobile devices employing WLAN (IEEE 802.11). The use of this kind of wireless networks infrastructures for positioning enables a powerful synergy between communications and location and allows solutions with good performances at moderated costs. However the adopted WLAN location methods suffer from important limitations that prevents from applying them to some fields that need more flexible and robust solutions. The main objective of this PhD is exploring precise WLAN location methods that allow overcoming these limitations.

The researched methods here are based on measuring the Time Of Arrival (TOA), which is the time that takes the signal propagating from the transmitter to the receiver. TOA-based location works in two stages: ranging and positioning. The ranging consists of estimating the distances between the targeted terminal and several WLAN access points, each distance obtained measuring the TOA and then multiplying it by the speed of the WLAN signal. After that, the positioning takes as inputs the estimated distances and the known coordinates of the involved access points and calculates the position of the terminal by means of a trilateration or tracking algorithm. The key problem is that the characteristics of the IEEE 802.11 protocols difficult to perform accurate TOA measurements. The main challenge that faces the research work reported here is demonstrating the feasibility of achieving this while keeping the modifications over standard WLAN consumer equipment at minimum. The objective of this work can be understood as exploring the current limits of TOA-based methods over WLAN, making contributions that form a complete TOA-based location method that goes a step forward with respect to the other existing proposals.

First, research on TOA-based ranging -the key component of TOA-based location methods- is reported. The general adopted approach consists of performing Round Trip Time (RTT) measurements employing IEEE 802.11 MAC frames, taking the maximum advantage of the combination of IEEE 802.11 protocol and WLAN consumer devices mechanisms. After that, the performed research on trilateration/tracking -the second stage of TOA-based location methods- is explained. Finally some performed studies about the achieved location method are presented.

Contents

List of Figures

List of Tables

Part I

Getting started

Chapter 1

Introduction, motivation and goals

1.1 Location based services

Since the early nineties location has been used in many civilian situations mainly taking profit of the Global Positioning System (GPS). Some representative examples are the following:

- Management of a company's vehicle fleet (e.g. motorbikes, cars, vans or trucks), including a range of functions such as: vehicle financing, vehicle maintenance, vehicle telematics (e.g. tracking and diagnostics), driver management, fuel management and health and safety management.
- Navigation systems for people in cars or boats.
- Navigation systems for civilian aircrafts and ships.
- Use of GPS receivers by hikers to avoid getting lost.
- Tracking of family pets so that the owner can know where the pet is.
- Mapping activities, for applications such as Geographical Information Systems (GIS).

In these situations a different service that need and use the knowledge of a target's position is working. They are known as LBS [1] and in general they require the location determination and tracking of people or objects in outdoor environments. However in the last years the growing importance of ubiquitous and context-aware computing, the more mature wireless technologies and the new portable devices have provoked an important evolution of the LBS market and a growing business interest on it. The result is that nowadays many new applications and services for which the user's positioning information is key can be found, most of them requiring not only location data in outdoor environments but also inside buildings and in combined indoor/outdoor urban environments. Here are some relevant fields on which these applications can play a key role:

Health services: Location of doctors, nurses, patients and sanitary equipment in real-time in order to improve the response time to attend urgent requests, the efficiency and the quality of the service.

Emergencies management: Remote monitoring and guidance of the firemen or policemen involved in an emergency situation inside a building, in order to enhance the operation efficiency and at the same time guarantee their own security.

Transport: The pallets or product units can be tracked from the origin at the warehouse to the final destination. In that case the location of the targets must be provided in a combined indoor/outdoor environment.

Logistics: In warehouses and logistic centres in which lots of products are often relocated, knowing the location of the pallets, the transport equipment and the staff when required can become crucial to be competitive and efficient. In addition, it allows lessening the needed time to make an inventory, enhancing the productivity and the security of the staff.

Industry: The location of the product units during production processes can allow their automation and better quality control, so that the productivity, quality and cost-efficiency can be noticeably enhanced.

Tourism: Provision of information and alerts to the users equipped with portable devices in airports, railway stations and indoors events depending on their location.

Advertising: Provision of advertisements to the users equipped with portable devices in commercial fairs and supermarkets depending on their relative location to the advertised product.

Penitentiary services: Tracking of dangerous inmates in order to enhance the security.

Children: Tracking of children in crowded events or children play centres in order to guarantee their security as much as possible.

Disabled people: Tracking of disabled people so that they can be easily located in case they raise an alarm.

Robotics: Enable self-positioning capabilities to robots in order to provide autonomous navigation.

Domotic systems: Smart home intelligent systems that adapt themselves to the location where the user is, performing automated actions depending on that location or providing the corresponding interfaces for the domotic services available at that location.

Car parking: location of vehicles in indoors car parks in order to enhance the security, to provide automated navigation to a free parking place or to achieve automated car parking.

Sport events: tracking of sportmen/women in indoors sport events in order to collect statistical data, achieve accurate measurements or enhance the event visualization for the audience.

Network security: Location of users of wireless networks in order to lessen the vulnerability to illicit network connections by unauthorized users.

1.2 Indoor positioning

The current situation is that most LBS need to locate or track physical assets inside buildings accurately, and therefore the availability of advanced indoor positioning has become a key requirement for the LBS market. Unfortunately this requirement cannot be met by the GPS: although it can be very accurate and efficient for an unlimited number of users in outdoor scenarios, it is unable to provide valid location information in most existing indoor environments, especially deep indoors, because most of the times the signals transmitted from the GPS satellites are blocked by walls. In addition, GPS often fails in urban canyons due to buildings obstructing the path between the receiver and the satellites. Possible alternatives include wide area cellular-based positioning systems such as Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS) and Universal Mobile Telecommunications System (UMTS), but they are not accurate enough for most of LBS. Hence, localization techniques specifically designed for being used in indoors are currently being researched and developed in order to complement GPS so that the continuous tracking of mobile targets, regardless of their environments, becomes feasible.

The basic design concept behind indoor positioning systems is very different with respect to classical global systems such as GPS, Galileo or the wide area cellular-based ones. They provide localization in a limited area, acting as local systems, so that they employ infrastructures deployed inside the building instead of satellites or wide area cellular networks. In addition they usually face worse scenario conditions than in the case of outdoors, such as coping with the harsh indoor environment caused by radio signal propagation (e.g., multi-path and fading) and changing environmental dynamics (e.g., relative humidity level, human presence, and furniture variations). Thus, research on indoor positioning technologies has produced a vast literature since the mid-nineties. During the early years, research focused on the use of new infrastructures devised for the specific purpose of geo-location, entailing the development of a new network of reference sensors and a signaling system. These approaches were intended to work in very small areas, and most of the time they were extremely accurate. The main problems were high costs, complex deployment, and difficulties when scaling to large indoor areas; consequently most of applications could not benefit from these solutions in practice. In addition users have to be equipped with custom terminals or updated with custom location hardware. Some important examples of proposed solutions that follow this approach are Cricket, Active Bats, and the Ad Hoc Location System (AHLoS) ([2]). There were several technologies available -for example, infrared, ultrasound, optical, and radio frequency- but none was presented as a total solution.

Years later, advances in wireless communications technologies enabled the use of communications protocols to design new indoor positioning systems, so that existing wireless communications networks could be employed as infrastructures to build location systems. This way, more cost-efficient solutions can be achieved, since the cost of the deployment is substantially reduced and any device compliant with the selected communications standard can be used (either belonging to the network infrastructure or the user terminals). In addition, modularity and flexibility are high because the network infrastructure can also support communication services such as data transfer, which can be combined with location capabilities. However, it is important to notice that most of these technologies were not designed for positioning, so that the performance of the location solution is most of times limited by the performance of the employed network and therefore additional challenges emerge when trying to achieve high positioning performance.

The scope of this PhD Thesis is the indoor positioning using wireless communications networks, more precisely Wireless Local Area Networks (WLAN). The rest of this introductory chapter shows the available wireless communications technologies and explains in detail the different existing positioning techniques when WLAN is used. Once this necessary background is provided, the second chapter is about the motivation and the specific objectives of my research work. The subsequent chapters are grouped in three different parts (II, III and IV) that reports the research work that I have carried out to achieve these objectives. Finally in part VI some appendixes, including the list of publications during the four years of PhD thesis, are presented.

1.3 Indoor positioning using WLAN

Several wireless communications technologies can be considered to be employed for indoors positioning: IEEE 802.11 (WLAN), IEEE 802.15.4a Ultra Wide Band (UWB), Bluetooth, and Zigbee (IEEE 802.15.4). The latter two correspond to Wireless Personal Area Networks (WPAN) technologies and therefore are not suitable for covering whole buildings due to their short signal ranges. Regarding the other options, positioning with WLAN has become very popular due to several reasons: WLAN it is a mature technology that follows at the same time a continuous evolution to enhance its capabilities through new standard versions, IEEE 802.11 networks are widely deployed in many buildings for communications purposes, they can be deployed with minimal cost and effort because of the low cost, wide availability of the hardware and long signal range (large indoor areas can be covered with just few fixed network elements). In addition they can provide good combined capability of positioning plus data transfers at high rate. On the other hand the emerging UWB technology is just starting its expansion after the standard ratification process, and although theoretically better positioning accuracy than with WLAN can be expected, the high cost of UWB equipment causes the current limited success of UWB in consumer products and at the same time makes it difficult achieving cost-efficient positioning solutions. This explains why WLAN has been preferred for this indoor positioning PhD thesis.

Since IEEE 802.11 does not include specific characteristics to facilitate the position calculation of WLAN devices, building an accurate WLAN-based localization system presents some difficulties. In order to overcome them, the scientific community has explored several location techniques over WLAN. A location technique is a procedure that allows calculating the position of a target. They can be classified by different criteria, for instance depending on the employed location metric. Metrics are observed and measured by the wireless nodes and employed as input for the position computation. The most commonly used positioning metrics with WLAN are the cell of origin, the Received Signal Strength Indicator (RSSI), the propagation time of the signal, and the signal's Angle Of Arrival (AOA). The employed metric is an important characteristic of a location technique because it affects the positioning performance and determines often the simplicity of the system; however a more practical classification takes into account how these metrics are combined to estimate the position (i.e. taking only into account the employed location technique): trilateration, triangulation, fingerprinting, or cell identity. On the other hand, given a specific location technique it can be implemented in different ways regarding the degree of centralization of the position calculation. Basically two main options exist:

Centralized approach (also called network-based): the position calculation takes place in one element of the infrastructure, typically in a location server which has fast wired connection with the nodes of the wireless network infrastructure. The needed measurements can be done by the nodes of the infrastructure or by the terminal, and once they are collected they are sent to the location server. In some cases the centralized position calculation can cause a data traffic or processing bottleneck in situations of large number of users so that the scalability can be affected; this has to be carefully analyzed before adopting this approach. The main advantage is that the user terminals are not loaded with the position calculation process and therefore high processing capability for location is not required on them.

Decentralized approach (also called terminal-based): the position calculation and the measurements for collecting the location metrics are performed in the user terminal. This approach allows two main advantages: better scalability to large number of users (the mentioned bottleneck is avoided) and better user privacy, since the calculated positions can remain in the terminal without being transferred to the common infrastructure. The main drawback is the additional requirement for processing capability in the user terminals.

Next paragraphs explain the basis of each WLAN location technique, the positioning performance that can be expected and their complexity in practice; in addition the most relevant proposals are shown. The performance of each technique is analyzed according to several quality of service parameters referred to the achieved positioning performance: accuracy, latency (required time to compute a position of a node, also called response time), scalability and availability. After that, some representative commercial location solutions based on WLAN are commented.

1.3.1 Cell identity

Cell identity (cell-id) was the first approach proposed for positioning terminals in wireless networks. It is based on the fact that wireless networks are deployed in a cellular fashion: they are divided into cells, each consisting of one base station covering a portion of the whole network coverage and hence handling only a reduced amount of users. The location of the base stations is known at network-design stage. Accordingly, knowing the base station to which the user is linked, the user's position can be estimated.

The main advantages of this technique are availability (i.e., full availability for connected terminals), response time, and scalability. Because the network has the necessary information, terminals do not have to measure or deliver any metric. This feature allows the localization of legacy terminals without fundamental changes, which minimizes the deployment cost. However, techniques based on cell identity present drawbacks that constrain its use in location systems. The accuracy of cell-id obviously depends on the cell size; because the size is often large, the location accuracy is diminished to a level not acceptable for most LBS. Furthermore, the consistency of this location technique is poor because the cell size varies depending on the context (e.g., urban cells tend to be smaller than rural ones, and cells with light traffic tend to cover neighboring cells with heavy loads).

The use of cell-id in modern mobile telephone networks, such as those using GSM ([3]) or UMTS ([4]) technology is regulated by the 3rd Generation Partnership Project (3GPP). Nowadays, almost all public network operators implement the cell-id technique, and many mass market services employ it for entry-level services in which accuracy is not a key factor.

In WLAN networks, two main options exist to implement this method: using Remote Authentication Dial-In User Server/Service (RADIUS) based authentications ([5]) or asking the Access Points (APs) about their clients via Simple Network Management Protocol (SNMP) ([6], [7]). The former usually provides slightly longer (i.e., worse) positioning latency, but the generated network traffic and the number of loaded APs is noticeably smaller than with SNMP. However, not all Wireless Fidelity (WiFi) APs support RADIUS or even SNMP, and accuracy is limited to the size of a wireless network cell. According to some manufacturers, the maximum operating range of an IEEE 802.11 AP can vary between 100 and 300 meters outdoors and from 30 to 100 m. indoors. This accuracy meets the requirements for a limited number of LBS, but for most of them greater accuracy is necessary.

1.3.2 Fingerprinting

Most currently available WLAN location solutions are based on this family of methods, also called the radio-map-based technique. The idea behind this method is using the RSSI received from specific APs as a location-dependent parameter. The calculation of the position consists of measuring the RSSI from several APs and then attempting to match these measurements with the RSSI values of previously calibrated location points stored in a database. This database, or radio map, has to be built before the system is operational. Hence, the method works in two phases: an offline training phase and an online positioning phase. In the first phase, RSSI measurements must be obtained by placing the mobile device at each reference point and measuring the RSSI from all applicable APs. This way, the fingerprint of each point is stored as a set of RSSI figures in the database along with the known point's coordinates. In the second phase the target's localization can be estimated: the device measures the RSSI from the APs and compares these measurements with the data recorded in the database by means of a matching algorithm. The output of this process yields the likeliest location of the device. Figure 1.1 illustrates this second phase.

Matching algorithms

The key component of fingerprinting is the matching algorithm, because it determines both accuracy and latency. There are two main types of algorithms, deterministic and probabilistic. In deterministic algorithms, the RSSI at a specific physical location is characterized by a scalar value (e.g., the average of the RSSI recorded samples) and non-probabilistic approaches to estimate the user location are employed. One widely employed deterministic algorithm is the Nearest Neighbor, which computes the distance in signal space between the observed set of RSSI measurements and each RSSI set recorded in the database and then selects the location that minimizes the distance.

In probabilistic algorithms, all possible information is considered when characterizing the RSSI. Thus, probabilistic approaches incorporate additional data such as movement history or map information. The RSSI characterization point is important for accuracy because the signal strength at a physical point is not constant; rather, it varies over time due to factors such as temperature changes, human movement, and the effects of the indoor radio propagation channel. Therefore, taking only one RSSI scalar value discards some important information. Most probabilistic algorithms employ

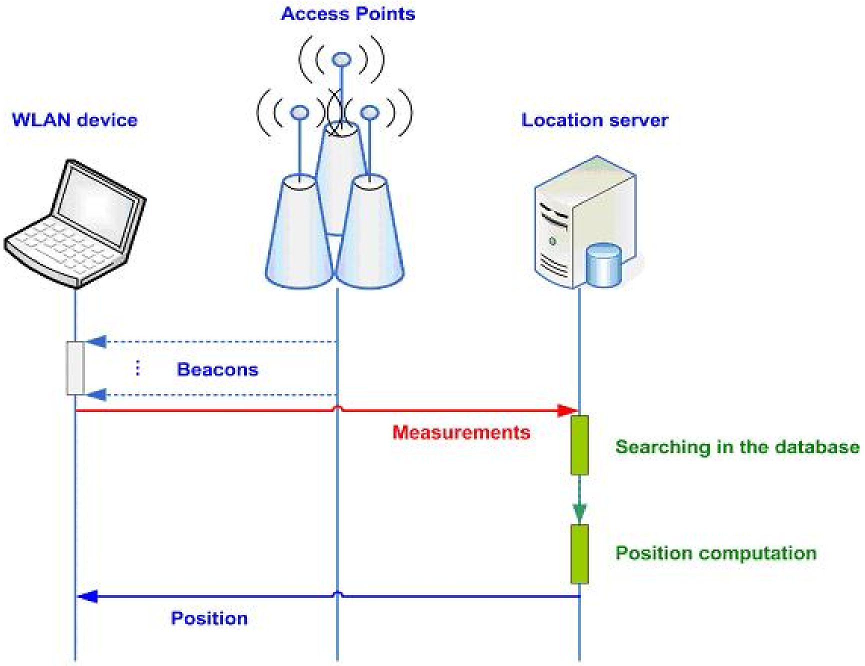

Figure 1.1: Online positioning phase in fingerprinting

Bayesian networks for inferring the user's location. These algorithms have been employed successfully in the field of robot localization, and they were proposed for fingerprinting with the intention of achieving higher accuracy by integrating several sources of information. They are based on the simple principle of the Bayesian rule: the probability of being at a certain location, given a certain observation, is equal to the probability of observing the mentioned observation at the mentioned location and being at that location in the first place. During the localization process, the conditional probability of being at that location is calculated using the fingerprints in the database, and the most likely location becomes the user position estimate.

Relevant approaches

The first deterministic fingerprinting proposal is the RADAR system ([8]). The matching algorithm used in this system is the Nearest Neighbor. Some interesting issues are addressed in this proposal, such as the significant variation that the signal strength suffers depending on the user's orientation (due to the obstruction caused by the user's body), the number of physical locations for which data needs to be collected, and the number of RSSI samples collected for each physical location. Experiments show that accuracy is around three meters for 50% of the cases.

Another significant contribution belonging to this group is ([9]), in which the performance of three different algorithms is assessed through experiments: the Nearest Neighbor, the back propagation neural network, and a third algorithm that introduces a probabilistic approach using histogram matching. Experiments conducted using three APs demonstrate that the neural network algorithm outperforms the other two in terms of accuracy.

The pioneer contribution proposing the use of a probabilistic algorithm (Bayesian) was the Nibble system ([10]). Nibble inspired later works, ([11]) being one of the most relevant, in which a post-processing technique called sensor fusion is used to refine the output from the Bayesian inference. Results show accuracy within 1.5 m. of error for 83% of the cases. In addition, in comparison to other proposals, the error due to the user's orientation is reduced. The HORUS system ([12]) appeared in 2005 with innovative features and performance. This design pursues two main goals, high accuracy and low computational requirements, so that it becomes feasible to implement it in energy-constrained devices. For accuracy, various causes of channel variations are identified and mitigated through techniques such as correlation, continuous space estimation, or small-space compensation. The accuracy enhancement is noticeable: close to one meter of error for 80% of cases. The low computational requirements are accomplished by using location-clustering techniques, which allow a client-based approach for system implementation and thus achieves better scalability than employing a network-based architecture.

Performance characteristics

The accuracy provided by this technique can be considered as room-level, because in most proposals and cases the positioning error remains below three or four meters; the yield and consistency can be considered good too. Most of the time latency can be kept within a range suitable for all the applications. The scalability to large numbers of users inside a limited area is good, but it is rather costly to scale these systems to large areas. The main advantage of this method is that RSSI can be obtained in every IEEE 802.11 device through low level Application Programming Interfaces (APIs) without the need of hardware or firmware modification. RSSI is much easier to achieve than signal propagation times or incidence angles. However, this technique presents two important drawbacks that limit its applicability for certain LBS that require flexibility and fast deployments. First, it requires extensive manual calibration efforts to build the database (i.e., the off-line training phase is costly and time-consuming.) Second, environmental (e.g., furniture) changes have a negative impact on the positioning accuracy. In some cases increasing the amount of APs allows better accuracy, but it also has negative effects such as collisions between signals from overlapping channels and the consequent costs.

Current trends

Recently some research has been carried out with the purpose of reducing the manual effort needed to construct the database. One example is ([13]), in which the total amount of manually collected RSSI samples is reduced by minimizing the number of sampled reference locations and the number of RSSI samples in each location. The main idea of this approach is applying an interpolation method to estimate the RSSI values in the missing points. Results show that positioning accuracy only decreases between 6% and 16% when reducing the number of collected samples to one-third. However, the more desirable solution is a totally automatic database building process ([14]), for which automated sensor-assisted online calibration employing Radio Frequency Identification (RFID) sensors is proposed. This approach also tries to avoid accuracy degradation due to environmental changes by labeling a subset of RSSI samples obtained from the online phase with the RFID sensors and using these samples to train different context-aware radio maps. Then, the radio map that best matches the current environmental situation is employed for the positioning process. Results demonstrate an error reduction of 2.6 m. with respect to traditional fingerprinting systems that do not adapt to environmental conditions.

Existing techniques such as tracking filters can be applied to fingerprinting as an upper layer over the matching algorithm. An interesting example is ([15]), in which the use of particle filters is proposed. Accuracy is not improved with respect to existing fingerprinting solutions such as the HORUS system, but a smoother target's trajectory is obtained. In addition, the technique constrains the obtained positions on a Voronoi diagram of the building in order to avoid incoherent trajectories (e.g., crossing walls) and provide more consistency with sudden velocity variations.

1.3.3 RSSI-based ranging and trilateration

This technique is based on estimating the distance between WLAN nodes by employing RSSI measurements as a metric. This metric is converted into distance by employing a proper propagation model and estimating the distance from the power attenuation introduced by the radio-path. Once this distance estimation, known as ranging, is performed between the target and several APs, the target's position can be estimated by means of trilateration (as shown in 1.2 or tracking algorithms (assuming that the coordinates of the APs are known). The difference between trilateration and tracking is that the latter employs past position estimates as additional information for computing the position. Tracking usually leads to better accuracy and a smoother estimated trajectory than trilateration and is often employed when the time between position requests is small. The trilateration and tracking algorithms usually correspond to well-known algorithms for outdoor positioning with non-complex tailoring. Three reference points are needed to estimate a 2D (2-dimensional) position.

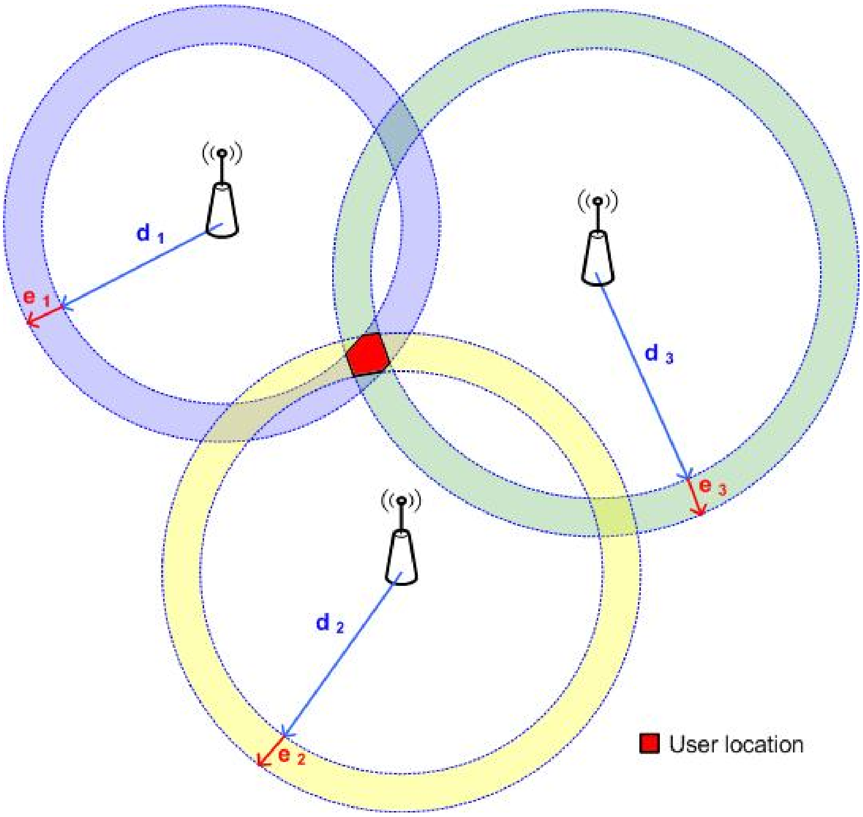

Figure 1.2: Geometrical scheme of trilateration

RSSI-based ranging

The main challenge is achieving accurate distance estimates, which requires very accurate propagation models to estimate the channel's radio losses with precision. This is a hard task because radio signals are affected by random occurrences that make that the signals propagate in unpredictable ways: reflection, diffraction, and absorption occur when the waves encounter obstacles. The signal reaches the receiver following more than one single path, a phenomenon known as multi-path, and consequently the received RSSI suffers random variations. In addition, environmental features, such as atmospheric conditions or the presence of people and other obstacles (2.4 GHz is a resonant frequency of water), also affect power reception. In practice, the consequence of all these factors is that the instantaneous RSSI fluctuates along time. Numerous studies have been conducted to determine accurate propagation models. One of the first examples is within the scope of the RADAR system research ([8]): several models were tested experimentally; in all cases poor results were obtained with respect to the RADAR fingerprinting approach. Adapting the radio propagation model for free space to indoor environments, including the number of floors in the path or the number of walls ([16]), is not a satisfactory approach since the number of obstacles is not known a priori. Others approaches try to improve models of the radio signal propagation indoors ([17, 18]), but currently single, consistent models yielding accurate distance estimates are not yet available.

Recently alternative and more advanced methods have been explored. A conceptually simple contribution ([19]) proposes to refine the obtained sets of RSSI measurements by processing them to mitigate noise and detect uncertainty before employing them for distance estimation. In addition, the proposed system is completed with a tracking algorithm using the Extended Kalman Filter (EKF) to calculate the position estimate from distance estimations, minimizing the variance of the estimation error. An accuracy of less than three meters of error is reported; however, authors recognize that the propagation model is specifically tuned for the tested environment. In [20], the RSSI is measured in all the IEEE 802.11 channels and the resulting figures are averaged in order to take advantage of the frequency diversity. Simulations of a Line Of Sight (LOS) scenario with trilateration show positioning accuracies close to three meters. In [21], it is proposed to perform on-line RSSI measurements periodically between the APs of the positioning system and then build a RSSI-distance model in order to mitigate the undesired effects of multi-path fading, various atmospheric conditions, and physical changes in the environment. This method produces a dynamic and adaptative propagation model. Experiments indicate a good response to environmental fluctuations, keeping the positioning error close to three meters.

Performance characteristics

Metrics needed by RSSI-based ranging can be easily accessed at the device. Consequently, this technique can be implemented with software-only solutions in legacy WLAN terminals. The main drawback is its poor and unstable accuracy due to the difficulty of achieving accurate and consistent RSSI-based ranges. The latency can be kept low as in fingerprinting. The scalability to a large number of users is similar to fingerprinting, whereas scalability to large areas is better because the database is much smaller, storing data such as model parameters. In contrast to RSSI fingerprinting, this technique is not considered advanced enough. One indicator is that although there are some proposals for using RSSI (known as network measurement report in the Public Land Mobile Network (PLMN) terminology), its limitations lead 3GPP to exclude RSSI techniques from wide-deployed network technologies such as GSM or UMTS.

1.3.4 TOA-based ranging/trilateration

TOA based techniques compute the target location using a trilateration or tracking algorithm, taking as inputs the measured distances to reference points and the coordinates of these references, as in the case of the RSSI-based ranging/trilateration technique. However, the difference is that in TOA-based methods the distance between WLAN nodes is estimated by measuring the TOA (i.e., the time that the signal spends traveling between them) and then multiplying by the speed of the radio signal, which is very stable. Since the current PhD thesis explores this technique, deeper details about how this technique works, relevant proposals and current trends are explained in other chapters (mainly 2 and 3) of this document.

Performance characteristics

This technology has interesting properties that make it useful for WLAN. Since TOA is more stable and less environmentally-sensitive than RSSI, TOA-based ranging is more accurate than RSSI-based ranging, resulting in a positioning accuracy similar to or better than RSSI fingerprinting. Theoretically, TOA-based location techniques overcome the limitations of RSSI-fingerprinting by accommodating environmental changes and enabling flexible and easy deployment. The penalty is worse scalability to large number of users due to the need for network traffic in order to estimate the distances.

On the other hand, the scalability to large areas is good because the process at each terminal is always the same, and there is no database that grows along with the covered area. A key issue makes this technology more difficult than RSSI-based techniques for WLAN implementation: the IEEE 802.11 standard does not provide by itself any mechanism to accurately measure propagation times. This means that a priori it is really difficult to obtain accurate metrics by means of software-only solutions over standard consumer WLAN devices, and for that reason in practice most of proposals consider hardware modifications over the nodes; thus, increases in cost and complexity are incurred.

1.3.5 Time Difference Of Arrival (TDOA)

This technology calculates the time difference between the TOA from the transmitter to two reference points at different known positions. These time differences are converted to distance differences by multiplying them by the constant speed of the radio signal. As in one-way TOA, there is no need for synchronization between transmitter and receiver, but all APs must be synchronized with the same clock reference. Geometrically each estimated range difference gives a hyperbola with foci at the reference point receivers where the target can be located. A trilateration algorithm is then employed to estimate the position where at least two hyperbolae intersect (Figure 1.3).

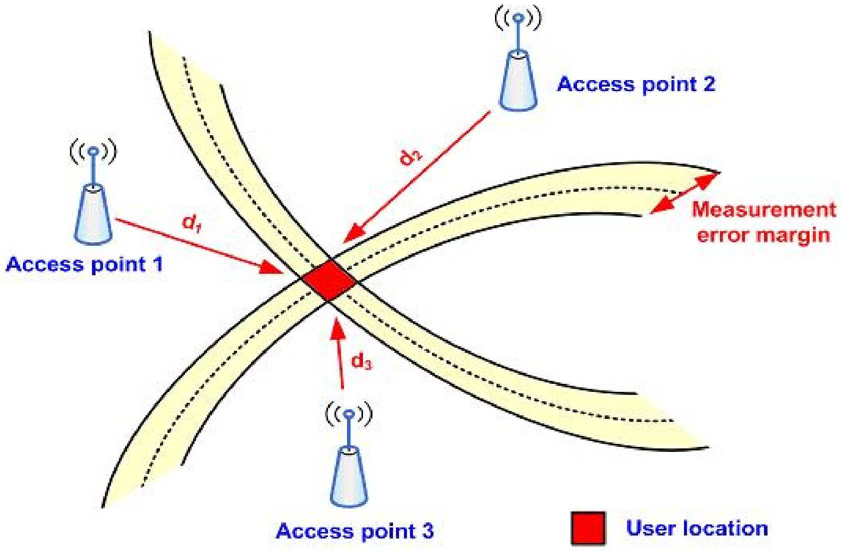

Figure 1.3: TDOA trilateration

Relevant proposals

Due to the complexity of the TDOA-based systems, existing proposals are not as numerous as in the case of RSSI or TOA-based localization. The main difference between TDOA approaches is the method to synchronize the APs. A frequent approach adds a location server that computes the clock offset of the APs using synchronization packets and takes into account the estimated deviations accordingly when calculating the position. [22] is a representative example in which the time difference measurements are computed for the APs by means of a cross-correlation technique, modifying the APs by adding a dedicated location module. Another variation of TDOA created with the objective of avoiding the synchronization mechanism is Differential Time Difference of Arrival (DTDOA) ([23]). Accuracy based on simulations is around 0.5 m., which is a substantial improvement with respect to the conventional TDOA technique.

Performance characteristics

Accuracy using this technique is similar to TOA-based or RSSI fingerprinting methods. The time subtraction in TDOA calculation eliminates some of the measurement error associated with TOA-based ranging. Necessary synchronization between APs increases the deployment complexity and cost and decreases the flexibility and scalability of TDOA-based systems to large areas. In addition, given the constraints of the WLAN standards, this technology poses the same challenges as TOA in achieving accurate time measurements; thus, hardware modifications for current WLAN nodes are required.

1.3.6 Angle of arrival or direction of arrival

This technique uses an AP to determine the direction of the arriving signal from the mobile device to be located. The 2D location of the mobile device can then be determined by triangulating with AOA information from at least two known reference points. Figure 1.4 illustrates the procedure; in the figure, a1 and a2 are the angular errors achieved in the position estimation.

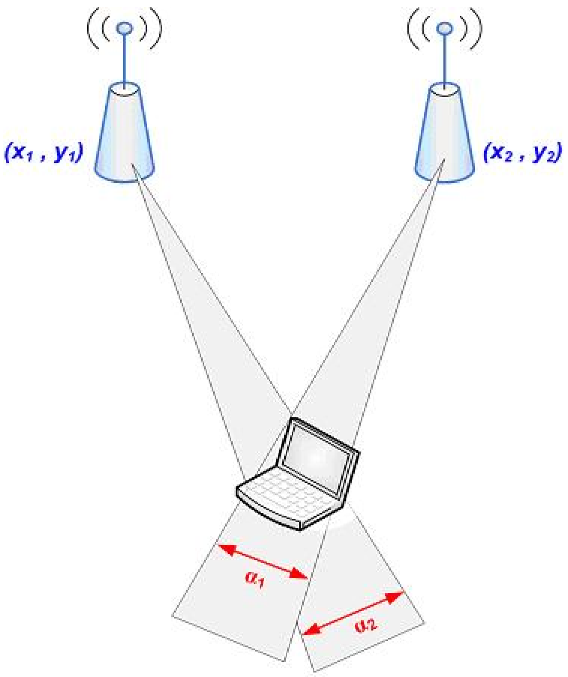

Figure 1.4: Positioning with AOA

Relevant proposals

This technique has not raised great interest regarding WLAN application. In fact, existing proposals for WLAN focus on combining AOA features with other localization techniques. AOA can be combined with RSSI-based ranging; the additional AOA information helps to mitigate the negative impact of indoor environments on RSSI-based range measurements. In [24] authors propose the use of special VHF Omnidirectional Ranging (VOR) IEEE 802.11 base stations to provide AOA and RSSI-based range measurements. The base station includes specific hardware with a continuously rotating directional antenna and software-based ranging estimation. An algorithm that combines trilateration from calculated ranges and triangulation from calculated angles is proposed to calculate the final position of the target. Results show positioning accuracy close to two meters. The idea after combining the two techniques is to improve the performance level of RSSI fingerprinting in terms of accuracy, needed infrastructure, and robustness in coping with environmental changes. Representative proposals of this kind include [25] and [26]. In the latter approach, hardware similar to [24] is employed. Its main advance is decreasing the number of base stations needed by half without degrading the positioning accuracy, and the amount of training data required is significantly less than classical fingerprinting solutions.

AOA measurements can be used for mitigating the Non Line Of Sight (NLOS) error of TOA-based positioning. The main idea assumes that the signal from the mobile target reaches each base station via one dominant scatterer (each base station with its own dominant scatterer). The scatterers' coordinates are then included as unknowns in a TOA/AOA-based cost function for calculating the position. Results show that, compared with solely TOA-based approaches, the performance of this algorithm is especially good when the target is in a NLOS situation with all the APs, a common occurrence in certain indoor environments. Both [27] and [28] also follow this idea.

Performance characteristics

Situations of NLOS between transmitter and receiver impair the accuracy of this technique in indoor environments. Long distances between APs and the terminal also decrease accuracy because the angular error increases with distance. Highly directional antennae are necessary, which means specific, complex hardware must be implemented to locate WLAN terminals. Accordingly, when applied to WLAN, the consistency and practical viability offered by this technique alone are poor. However, this method might become more attractive as IEEE 802.11 moves to Multiple-Input Multiple-Output (MIMO) capabilities. In this case, the direct path could be emphasized and, by hybridizing with other WLAN localization techniques, the number of reference points required to compute the position could be reduced. Strengths of AOA are that it does not require pre-calibrations, it is unaffected by environmental changes, and it scales well to large areas.

1.4 Some commercial location solutions based on WLAN

Several IEEE 802.11-based location and tracking products are commercially available nowadays. Their cost-effectiveness and accuracy are appreciated by users across a variety of industries, including health care, government, mining, oil and gas companies, manufacturing, and logistics. Here, a brief overview of the most relevant solutions is provided.

1.4.1 Ekahau real time location system

The Ekahau Company was founded in 2000, and their location system was launched in 2002 as the industry's first WLAN-based location system. The Ekahau Real Time Location System (RTLS) ([29]) is a software-only real-time tracking solution over existing IEEE 802.11 networks. The technology is based on RSSI and fingerprinting with probabilistic algorithms. In addition, this system employs innovative algorithms and techniques patented by Ekahau, such as the probabilistic signal strength modeling and the predictive algorithm to compute location estimates. Theoretically people, furniture, doors, and minor environmental changes do not require re-calibration of the positioning model. Location information can include x and y coordinates, building, floor, room, or any user-defined zone. The targets can be Ekahau WiFi location tags as well as standard WiFi-enabled devices (e.g., Personal Digital Assistants (PDAs) or laptops). Positioning accuracy ranges from one to three meters of error on average, depending on the layout deployed by the network. According to the product specifications, one meter of error can be reached if there are three or more overlapping AP signals. The positioning application can be integrated with existing customer middleware, Enterprise Resource Planning (ERP), a database, a workflow and other enterprise systems through a Hyperlink Transport Protocol (HTTP) / eXtensible Markup Language (XML)-based programming API and a Software

Development Kit (SDK). Ekahau investors include Nexit Ventures, 3M Company, Finnish Industry Investment Ltd., Sampo Group, the Finnish Funding Agency for Technology and Innovation, ETV Capital London, and a group of individual investors.

1.4.2 AeroScout visibility system

AeroScout ([30]) invented the industry's very first Wi-Fi-based Active RFID tag back in 2003. This system is a real-time tracking, active radio-frequency identification (i.e., Active RFID) and telemetry solution over existing IEEE 802.11 networks. It can require specific hardware equipment or modifications to the firmware of the existing APs depending on the chosen location technique and performance requirements. Three techniques can be employed to calculate the positions of the targets depending on environmental characteristics and user requirements:

- TDOA: The system employs this technique for outdoor and open indoor environments. Specific fixed hardware equipment (i.e., AeroScout location receivers) is required. These receivers read the beacons sent by the targets and perform the TDOA measurements. They send the measurements to the location server applications, which perform the position calculation. The signals employed for the TDOA measurements are standard IEEE 802.11 beacons. AeroScout tags use a unique "beaconing" method that communicates with minimal disruption to the network and allows scalability, unlike the competing "association" method. A patented clear channel assessment mechanism is employed to ensure that traffic does not interfere with other WiFi traffic.

- RSSI-based technique: In that case, IEEE 802.11 APs measure the RSSI with modified AeroScout firmware.

- Active RFID: specific fixed hardware equipment (i.e., AeroScout Exciters) is needed. Using AeroScout Exciters, a tag's passage through a defined area such as a gate or doorway can be detected. Exciters trigger very precise and immediate notification that a tag passed a certain threshold or is located within a very small area. These data are then added to the real-time location data coming from the WiFi APs and can add both clarity and immediacy where needed. Both AeroScout's WiFi-based Active RFID tags and standard WiFi-enabled devices can serve as targets. The degree of positioning accuracy depends on the environment. The system platform can be integrated with existing customer protocols by means of a Simple Object Access Protocol (SOAP) API among other provided tools. The main added values of this product are its flexibility, specific functionality, and suitability for both indoors and outdoors. Some Aeroscout partners are Cisco Systems, Microsoft, Aruba Networks, 3COM, Intel, and Belden.

1.4.3 Skyhook wireless WiFi positioning system

Skyhook Wireless was founded in 2003. The main difference between the Skyhook Wireless WiFi Positioning System ([31]) and other products such as Ekahau or Aeroscout is that it is intended to provide global coverage both indoors and outdoors, but the reachable accuracy is worse. It is a software-only location platform based on existing WiFi networks. Skyhook uses a massive reference network comprised of the known locations of over 23 million WiFi APs (serving as reference data for the position calculation) and requires the installation of a thin software client in the WiFi-enabled device to be located. The technology used to obtain the position of the target is based on RSSI; the positioning algorithms are developed by Skyhook Wireless. The device to be located receives the IEEE 802.11 beacons from all the APs in the area. Beacons include the unique signature and precise location of each AP. Typically, the device will receive more than five signals from any given scan. The results of this scan are then compared to the local cache of reference data or the central reference database via the network connection. The location engine filters out signals from APs that are unknown or may have moved their location recently, instead focusing on high-confidence points. The resulting list of reference points is then fed into Skyhook's patented suite of positioning algorithms, which then determines the user's current location. Targets are WiFi-enabled devices. The system provides positioning accuracy up to 20 meters. The more APs populating the area the more accuracy can be reached. The company has invested resources to build a massive coverage area for the system in North America and is currently rolling out coverage in Europe and Asia. As it grows, it repeatedly re-calibrates its reference data in order to maintain the same level of performance over time. The system complies with all location standards, simplifying the process of integrating with applications via standard interfaces such as Nation Marine Electronics Association (NMEA) and integrating within carrier networks via industry standards like Secure User Plane Location (SUPL).

1.5 Motivation

The overall final objective to which the scientific community researching on indoor positioning dedicates biggest efforts nowadays is in achieving an indoor location technique able to provide the closest performance to the following desirable characteristics: good positioning accuracy, good positioning availability, performance robustness and responsiveness to environmental changes (e.g., furniture, people, cars), quick and flexible deployment, good scalability to large areas and large number of users, small positioning latency and moderate cost (in other terms, a cost-effective solution). Reaching this high performance level would satisfy the requirements of most of the existing and devised LBS. On the other hand, it is important to provide the integration of localization and communication capabilities in order to provide a platform of existing communications infrastructures for allowing a proper LBS deployment. Although presently achieving all these goals with a single location technique remains a challenge, the research work about indoor location that makes up this PhD thesis has been carried out with the firm intention of making noticeable progress towards this objective.

The first choice made for designing the pursued indoor location method was adopting WLAN as the under-laying technology. This fact is important and deserves an accurate analysis that has been started in Section 1.3. Being aware of the overall objective explained in the last paragraph, using WLAN allows achieving a cost-effective solution because the equipment, both the infrastructure and the client hardware, is relatively cheap and accessible for the consumer, and the installation is simple. In fact WLAN infrastructures are available almost everywhere for data communication and consequently almost all modern mobile devices (from smartphones to laptops) are equipped with built-in WLAN interfaces. In addition large areas can be covered with just few static nodes due to the long WLAN signal range. On the other hand, WLAN is the best candidate regarding the desired combined capability location-communication.

Despite these advantages and the good current state of the IEEE 802.11 technology, this analysis would make no sense if good location performance could not be achieved. After analyzing the basic principles and characteristics of each available WLAN location technique in chapter 1, achieving all of these goals seems difficult considering the intrinsic limitations of each technique. For example, fingerprinting presents good positioning accuracy (room-level), a software-based solution that guarantees cost-effectiveness, and good scalability; however, dependence on a radio-map makes it vulnerable to environmental changes, and the significant task of building a database can prevent quick and flexible system deployments. The other RSSI-based technique, RSSI ranging-trilateration, allows easier deployments and more resilience in response to environmental changes, but accuracy is very poor compared with the fingerprinting technique. On the other hand, the methods based on TDOA and AOA require additional mechanisms (e.g. the synchronization between APs for TDOA) that provoke an increment of the deployment complexity and cost while lessens its flexibility.

1.6 Goal and global approach

Given this situation, TOA-based methods emerged as a promising alternative to replace or enhance RSSI-based solutions and overcome their important limitations, because TOA enables flexible and easy deployments and it is robust to changes in the environment while at the same time good accuracy (maybe not sub-meter but yes room-level) can be expected. The problem is that a priori the characteristics of the IEEE 802.11 protocols make it difficult to implement such technique reaching good accuracy without deep modifications of the hardware of the WLAN-enabled devices. The main challenge that faces this research work is achieving a WLAN TOA-based location method with all the inherent advantages of the TOA technique, and at the same time mitigating its known drawbacks. The objective of this work can be understood as exploring the current limits of TOA-based methods over WLAN, proposing a complete TOA-based location method that goes a step forward with respect to the current state of the art, getting closer to the "ideal" indoor location system that can bring together all the explained desirable characteristics.

Here it is important to recall the two main parts of the process that form a TOA-based location method:

Ranging: the distances between the targeted terminal and several APs are estimated. Each distance is obtained measuring the TOA and then multiplying it by the speed of the WLAN signal.

Positioning: the estimated distances and the known coordinates of the involved APs are taken as input to calculate the position of the terminal, by means of a trilateration or tracking algorithm.

The vital component of the method is the ranging, because even using close to optimum algorithms to calculate the position from distances, a good positioning accuracy is not possible if distance estimations contain significant errors. Ranging with WLAN entails the biggest challenges to reach good performance, especially good accuracy, given the limited characteristics of the WLAN protocols. In addition, this part requires data communication between the WLAN nodes and a tight interaction between their software and hardware, so that it has big and direct impact on the scalability, availability, latency, simplicity, robustness and cost of the solution. In other words, the performance level reached by a TOA-based location system depends to a large extent on the performance of its ranging sub-system. For this reason the major efforts of the research in this PhD Thesis have been dedicated to the TOA-based distance estimation technique. The basic idea behind the explored ranging approaches presented here is taking the major advantage of the IEEE 802.11 standard and their capabilities in order to reach the best location performance keeping the simplicity and cost as low as possible. In the next part of the document -i.e. the Ranging part, which includes chapters 2, 3 and 4- the performed research on distance estimation is deeply explained.

On the other hand, although the positioning part of a TOA-based method is of course also important, in practice it corresponds to a software module that entails less difficulties since it is not directly constrained by the limitation of the under-laying WLAN characteristics. In fact, most of trilateration and tracking algorithms employed for indoor location have been available for classical positioning (i.e. for outdoors) since many years ago. Despite these issues, important efforts have been dedicated to this part during this research in order to maximize the reachable location accuracy and availability. Chapters 5 and 6, which conform the positioning part of the document (Part III), explain the performed research on this topic.

Once the different components of the location technique are explored and thus the complete location method is proposed, I considered important studying the impact of the a priori major limitation of TOA-based location using wireless communications networks -its limited scalability to large number of users- on the proposed method. Part IV of the document includes this study, the proposal of a method to improve the scalability and finally an evaluation of the new expected capabilities for TOA location of the upcoming IEEE 802.11v WLAN standard.

Part II

TOA-based ranging over WLAN

Chapter 2

TOA-based ranging: hardware enhancement

This chapter presents the first research work on TOA-based ranging with WLAN performed during this PhD Thesis. The chapter is organized as follows. First the specific objectives of this work, which translate on requirements of the pursued ranging method, are explained. After that the most relevant related work is explored in order to give an idea of the state of the art on this topic. Then the studied ranging method intended to satisfy the previously established requirements is presented, from the basic ideas to the obtained results in real situations. Finally conclusions are given.

2.1 Objectives

The ranging part constitutes the essential step to achieve a TOA-based location method with WLAN. The objectives of this research on TOA ranging practically correspond to the ones listed for the pursued location technique. Thus the requirements of the aimed ranging technique are the following:

- Good ranging accuracy, enough to satisfy the requirements of most LBS.
- Low complexity of implementation. The hardware modifications over standard WLAN consumer equipment have to be kept at minimum levels, in order to maximize the feasibility of the achieved solution with a moderate cost.
- Easy deployment. Costly pre-calibrations and additional pre-installed mechanisms over a standard WLAN infrastructure should be avoided, in order to achieve high flexibility of the method.
- Immunity to the environmental changes. The ranging performance can not be affected by changes in the environment, such as furniture, people or cars.
- Low bandwidth consumption. This requirement has to do with two main aspects: easing the combined functionality of data transfers plus location for the network users and maximizing the scalability of the method to large number of users. The raw idea behind this latter aspect is that the less is the consumed bandwidth for location by each user, the less number of conflicts occur when accessing the shared wireless medium and therefore a larger number of users attached to the same network can simultaneously be located.
- Low computational cost. Since it is desirable that the mobile terminal is able to make the distance measurements by its own (in order to allow decentralized location approaches), and taking into account that the WLAN portable devices usually have constrained CPUs, high computational cost of the ranging method would affect the implementation in practice.

2.2 Related work

In TOA-based ranging with WLAN, the distance d between the mobile terminal and one WLAN AP is obtained by multiplying the TOA estimate by the speed of light (c):

$$d = c \cdot TOA, \tag{2.1}$$

so that the key step is obtaining the TOA. Two main approaches exist: measuring the one-way trip time and measuring the RTT. In the former approach, the receiver determines the TOA based on its local clock, which is synchronized with the clock of the transmitter. The latter, also known as 2-way TOA, measures the time spent traveling from a transmitter to a receiver and back to the transmitter again; this approach avoids the need for synchronization, which entails an increase in complexity and cost.

2.2.1 Estimating TOA at the physical layer

Measuring the TOA at the physical layer leads to accurate distance estimates, but specific hardware modules are needed, making the solution not implementable on standard WLAN devices. Most proposals are based on frequency-domain measurements of the channel response with super-resolution techniques, due to their suitability for improving the spectral efficiency of the measurement system. Some examples are the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT), Multiple Signal Classification (MUSIC) [32], and Matrix Pencil [33]. The more recent Prony Algorithm [34] may be considered a more advanced super-resolution technique because of its robustness, noise immunity, accuracy, and low bandwidth requirements. This algorithm determines TOA from estimation of the multi-path parameters of the transmission channel. Other methods are based on the correlation of the received IEEE 802.11 signal. A recent technique [35] consists of correlating the received signal with a long-training symbol stored in the receiver and afterwards obtaining the channel frequency response to refine the initial TOA estimation, which provides better accuracy than traditional correlation-based methods.

2.2.2 Estimating TOA at upper layers

When TOA is estimated at upper layers of the WLAN protocol stack, the adopted technique performs two-way ranging by employing frames of the IEEE 802.11 standard protocol (e.g., Ready To Send (RTS) – Clear To Send (CTS), Data-Acknowledgement (ACK), or probe request – probe response) as traveling signals. Efforts are concentrated on the mechanism for measuring the RTT in the WLAN enabled node, which typically means obtaining the time-stamps of transmission and reception of the selected frames for the RTT and then subtracting both time-stamps. Since the WLAN signal propagates approximately at the speed of light, a time resolution of a few nanoseconds is theoretically needed to achieve accurate RTT measurements. Currently, neither the IEEE 802.11 standard nor the WLAN chipsets provide timestamps with this resolution in the transmission and reception of the frames. Only in most of WLAN chipsets the reception of a frame is recorded with a resolution of 1 microsecond; since this resolution is enough for the operation of the WLAN channel access procedure, it is unlikely that chipset manufacturers will enhance the resolution. Given this situation, different alternative mechanisms have been explored by the scientific community researching on TOA-based WLAN location. Current existing mechanisms for measuring the RTT can be grouped in few basic approaches:

Mechanisms that entail major hardware modifications at the WLAN interface. Although TOA is estimated at upper layers, in that case modifications to the WLAN physical layer are needed in order to obtain time measurements at hardware level with very good accuracy. An example is [36], in which the internal delay calibration both at transmitter and receivers is employed, with the RTS/CTS frames exchange. In [37] the key for obtaining the timestamp on frame's transmission and reception is capturing a segment of the waveform and then performing a matched filter using the probe request - probe response exchange.

Pure software mechanisms using standard hardware time-stamps. Most of WLAN cards record the reception of frames time-stamping them with a resolution of 1 microsecond. Although the precise point of time at which this time-stamp is recorded is not documented, the good feature is that it is performed at the hardware level of the WLAN card. The theoretical drawback is its low resolution (1 microsecond error corresponds to 300 meters). A representative attempt to obtain a software-only solution taking advantage of this approach is [38], in which RTT measurements are collected using tcpdump and an additional monitoring node to handle the inexistence of time-stamps on the transmission of frames. The quantization error due to the low time-stamp resolution is mitigated collecting a large number of measurements and then performing an original statistical processing. The reported ranging accuracy is around eight meters of error, which is poor when compared to existing RSSI-based proposals. In 2008 the same authors propose a complete location method -called Goodtry [39] - that makes use of a similar ranging technique to [38] but adding some new features that allows improving the accuracy.

Mechanisms that entail minor hardware or firmware modifications at the WLAN interface. This approach tries to reduce at maximum the needed modifications so that they can be implemented as upgrades of the WLAN card firmware, while maintaining a good ranging accuracy as close as possible to the one provided by methods that entail major hardware modifications. The most relevant contributions belonging to this group are part of the research of my PhD thesis and their corresponding work is explained in the current chapter. After our results were published, some relevant research works that took our method as reference appeared ([40, 41, 42]).

Pure software mechanisms implemented at operating system level. This approach mainly consists of capturing the time-stamps for the RTT at the driver of the WLAN interface, when the hardware interrupt of the WLAN card notifies the operating system about the transmission or reception of a frame. The CPU clock can be used to time-stamp the RTT events, so that a good clock resolution is available. The theoretical problem is that the RTT measurements are affected by the interrupt latency, which depends on factors such as the load in the CPU or the state of the operating system. The most relevant contributions belonging to this group are part of the research of my PhD thesis and their corresponding work is explained in Chapter 3.

As it can be noticed, the most representative proposals except [36] have been published while this PhD has been performed. Thus at the moment of starting this research -beginning of 2005- relevant contributions about the RTT measurement hardly existed in the literature. As shown later, the first approach explored during this PhD is a mechanism that entails minor hardware or firmware modifications at the WLAN interface (the approach explained in this chapter). Later in 2007 I started the study of a pure software method that is implemented at the driver of the WLAN interface (see chapter 4).

On the other hand, it has been shown that TOA estimate can be validated with RSSI measurements in order to enhance their robustness. The idea behind this cross-validation is assuming that both measurements are statistically independent; if some statistical dependency exists–mainly due to channel fading–the two methods would not yield the same value ([43]).

In indoor scenarios, the multi-path propagation poses a challenge to the accuracy of TOA estimation, especially in NLOS situations. The signal reaches the receiver through indirect paths because the direct path is partially or totally blocked, and therefore the measured TOA can contain large, positively biased errors. This issue is treated in detail in chapter 5.

2.3 Analysis

2.3.1 Basic design issues

The objectives and the requisites explained before are the main guidelines for facing the main issues of the design of the ranging approach. First of all, in order to reduce the complexity of implementation and ease the deployment, it is discarded to measure the TOA at physical layer. For the same reason, the need for time synchronization between the terminal and the APs is avoided obtaining TOA through RTT measurements. In addition, the RTT measurements are taken at the terminal in order to allow decentralized location approaches, with the corresponding further advantages of better scalability and better user privacy.

In order to avoid as much as possible delays due to extra processing between network layers when measuring the RTT, the measurements are taken using the WLAN signal at the lowest possible layer above the physical, which corresponds to the MAC layer. Then it is considered essential taking the maximum advantage of the under-laying wireless network infrastructure, specifically making use of the capabilities offered by the IEEE 802.11 standard protocol. The first option that is considered is employing the RTS-CTS (Request to send - Clear to send) MAC frames. The RTS-CTS exchange is a handshaking mechanism of the Distributed Coordination Function (DCF) of the IEEE 802.11 MAC protocol. However, it is discarded because this mechanism is not enabled in most of WLAN deployments nowadays and therefore it would limit the deployment of the technique. The selected option is taking advantage of the fact that the reception of each uni-cast data frame is immediately acknowledged by the receiver WLAN node, it is taking the data and ACK standard data frames of the IEEE 802.11 MAC protocol as the signals to measure the RTT.

The IEEE 802.11 standard specifies several measures to alleviate some of the inherent disadvantages of wireless network systems. These are enacted at the MAC layer through Frame Exchange Protocol (FEP) mechanisms to combat the problems of data transmission over a shared and unreliable medium. FEP is employed by both APs and mobile devices as a medium reliability countermeasure to speed the process of data exchange confirmation that would otherwise have to been done through higher (and slower) layer mechanisms. Each data frame that passes between IEEE 802.11 stations is automatically acknowledged at the MAC layer by the recipient's network adapter hardware with an ACK frame during a system time interval known as the Network Allocation Vector (NAV).

Obtaining TOA from RTT

As mentioned above, the RTT corresponds to the time elapsed between the transmission of a MAC data frame to the AP and the consequent reception of the ACK frame at the mobile device. Figure 2.1 illustrates the process.

The propagation time of both frames (i.e. data and ACK) is assumed to be the same (i.e. the TOA); t_{TX_DATA} and t_{TX_ACK} are respectively the transmission times of the ACK and data frames at the physical layer, and t_{PROC_AP} is the time elapsed between the AP receives the data frame and the ACK is transmitted. This processing time in the AP theoretically corresponds to the the Short Inter Frame Space (SIFS). Because collision detection (as implemented in wired networks) is impractical in a wireless environment, 802.11 depends on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) techniques, also known as "listen before talk", to accomplish efficient data traffic management. FEP utilizes five different Interframe Space time intervals as self-regulating mechanisms to reduce the contention period during which stations vie for access to the medium (see Figure 2.2. These are specific to each physical layer implementation of the 802.11 standard and are the: SIFS, Slot Time, Priority Interframe Space (PIFS), Distributed Interframe Space (DIFS), and Extended Interframe Space (EIFS). The SIFS interval is reserved for receivers to process and transmit MAC layer responses (ACK frames) to incoming data frames. It is designed to ensure that no other station within network reception range attempts to transmit during that time.

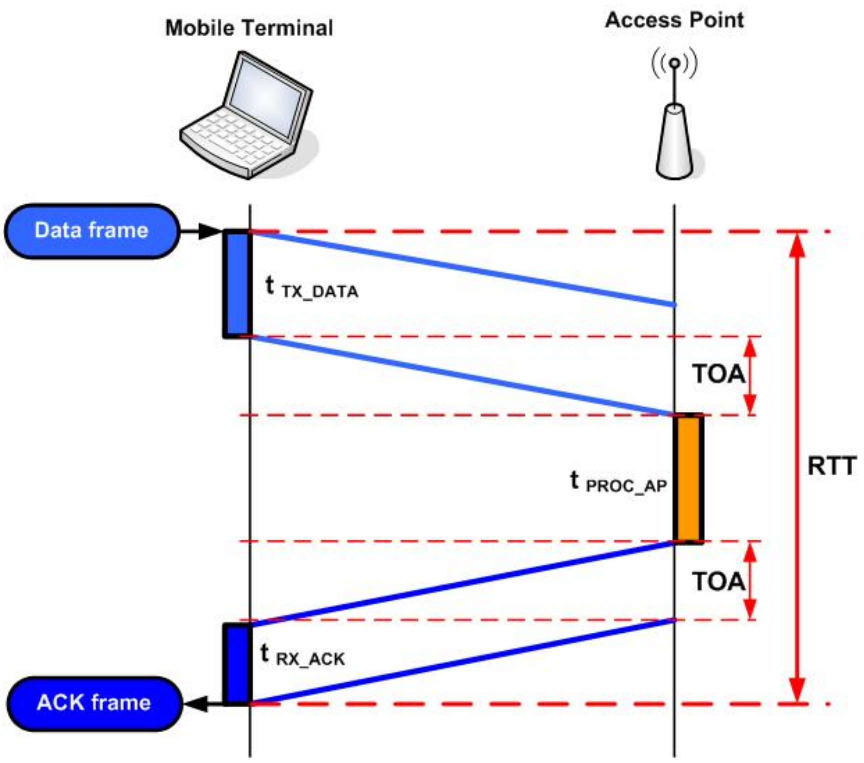

Figure 2.1: Basic scheme for the RTT with data and ACK frames

Then formally a RTT measurement can be expressed as follows:

$$RTT = t_{TX\_DATA} + 2 \cdot TOA + t_{PROC\_AP} + t_{TX\_ACK}. \tag{2.2}$$

Once the RTT is estimated, the TOA can be extracted from Equation **??** dist with the following expression:

$$TOA = \frac{RTT - t_{PROC\_AP} - t_{TX\_DATA} - t_{TX\_ACK}}{2}. \tag{2.3}$$

The RTT when the distance between Mobile Terminal (MT) and AP is zero is assumed not to contain propagation time, hence:

$$RTT_0 = t_{PROC\_AP} + t_{TX\_DATA} + t_{TX\_ACK}. \tag{2.4}$$

Then, combining Equations 2.3 and 2.4, the TOA for a distance i different to zero can be obtained as:

$$TOA_i = \frac{RTT_i - RTT_0}{2}, \tag{2.5}$$

where RTT_i is the RTT for a distance i different to zero. In practice, RTT_0 is obtained by placing the MT and the AP together and then estimating the RTT. RTT_0 is supposed to be constant over the time for a specific model of AP implementing a specific physical layer (e.g. the physical layer of IEEE 802.11b), regardless of the traffic load and other network and environmental parameters. This means that, theoretically, it is only needed to obtain this processing time (RTT_0) once, being valid from that moment for all the distance estimations. The upcoming IEEE 802.11v standard ([44]) is going to avoid this process because with this standard version the processing time in the AP could be calculated in real time and then sent back to the mobile device.

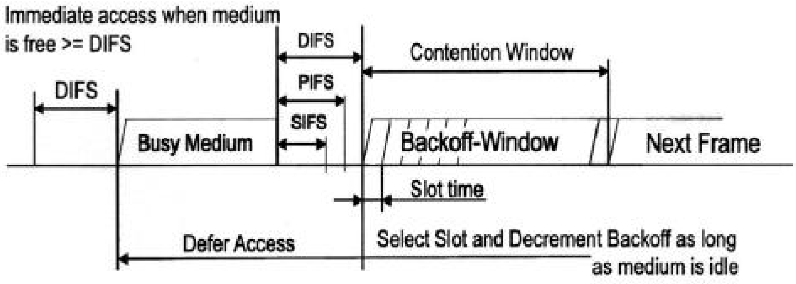
Immediate access when medium is free >= DIFS
DIFS
DIFS
PIFS
SIFS
Contention Window
Busy Medium
Backoff-Window
Next Frame
Slot time
Defer Access
Select Slot and Decrement Backoff as long as medium is idle

Figure 2.2: Interframe Space Usage (IEEE 802.11 standard)

2.3.2 Mechanism for measuring the RTT

From the previous analysis, the idea for obtaining the RTT in this method is that the MT sends a unicast data frame to the AP and counts the time until the corresponding ACK answer from the AP is received. As emphasized before during the explanation of the related work, given the lack of capabilities offered by the IEEE 802.11 protocol to capture the needed time-stamps with enough time resolution to accurately estimate the RTT, the key and challenging point of TOA-based ranging is achieving an alternative mechanism for measuring the RTT, it is to count the time that takes the frames exchange. The different options considered at the beginning of the research were the four approaches explained in the related work.

First of all, a mechanism that entails major hardware modifications was clearly discarded due to the requisites of the pursued ranging method. The alternative based on the available time-stamps provided by the WLAN chipset at hardware level was not explored due to their extremely low resolution, which I assumed far from the minimum one needed to achieve accurate ranging (another essential requisite in my case). This was just an hypothesis, because in that moment (beginning of 2005) no contributions reporting experiences about this alternative existed (the first one [38] was published during 2005); however reported results in [38] and [39] confirmed that it is very difficult to reach accuracy close to one meter by means of this kind of method. The other software-only alternative, consisting of measuring the RTT at operating system level, was not studied at this first stage of the research but kept in mind as a candidate method to further study.

The selected approach is a mechanism that entails minor hardware modifications at the WLAN interface while at the same time tries to reach an accuracy close to the one achieved by the techniques that require deep hardware changes. The first part of the idea is accessing the required events for measuring the RTT (i.e. the event of MAC data frame transmission and the event of MAC ACK frame reception) at the WLAN chipset level (the lowest level), it is at the same level where the before explained 1 microsecond time-stamps are captured, in order to avoid delays due to WLAN card firmware/WLAN driver communication or the operating system intermediation (e.g. interrupt latencies, communications stack processing, etc.). The second part of the idea is, once the RTT events are accessed at hardware level, counting the RTT employing the available WLAN card clock (44 MHz for IEEE 802.11b). This allows achieving much better resolution (22 ns) with respect to the microsecond resolution provided by default. In order to evaluate the feasibility of the method, a lab prototype is implemented. Taking into account that this clock with frequency f_{clk} is employed, the expression for estimating the TOA can be rewritten as:

$$TOA = \left(\frac{RTT - RTT_0}{2} \right) \left(\frac{1}{f_{clk}} \right). \qquad (2.6)$$

2.3.3 The lab prototype

The lab prototype implements the following functionality:

- First, it sends a MAC data frame to a known AP.
- Then it measures the elapsed time between the last bit of the data frame is sent and the first bit of the ACK is received from the AP.
- When the data-ACK exchange is finished, it stores the measured RTT.
- A new RTT measurement to the same AP can be performed if required by the user.

The prototype is implemented in a laptop, as it can be easily equipped with a WLAN card and easily manipulated. Since the RTT measurements are carried out in the mobile device, i.e. the lab prototype in this case, the employed AP is a commercial one without any modification or special configuration. The prototype is divided in two subsystems: the hardware subsystem, in charge of measuring the RTTs and reporting them to the software subsystem, and the software subsystem, which sends the MAC data frames to the AP and then receives the RTT value measured by the hardware subsystem and stores it in a text file. The key part of the prototype is the hardware subsystem, because it requires the manipulation of the WLAN card and the development of a hardware module that counts the RTT.

Hardware subsystem

The design of the hardware subsystem is as follows. Accessing the required MAC events extracting the proper MAC signals from the WLAN card chipset that indicate the transmission of the last bit and the reception of the first bit of a MAC frame, so that they can be used as triggers to start and stop the RTT counting. In addition, access the WLAN card clock extracting the clock signal from the same chipset. These three signals are the inputs of a simple hardware module that counts the RTT and provides it as output to the laptop in units of 44 MHz clock rising edges.

The employed WLAN card is a 3Com AirConnect 11 Mbps Wireless LAN PC Card 3CRWE737A-E1. This card can reach 11 Mbps using the 802.11b standard and it is selected because its MAC chipset is isolated from the Baseband Processor (BBP) chiptset, so that the required MAC signals are accessible.

Extracted signals from the WLAN chipset The frame transmission process works as follows (see [45] for more details). There are 4 signals associated with the BBP and MAC Transmit Ports: TX_{PE} (transmit enable), TXRDY (transmit ready), TXD (transmit data) and TXCLK (transmit clock). In the case of a transmission, the MAC signals the BBP with the signal TX_{PE}. The BBP forms the preamble and header and before the last bit of the header is sent, the BBP begins generating TXCLK to input the serial data on TXD. Then signals the MAC to begin transferring data with the signal TXRDY. This sequence is illustrated in Figure 2.3. When TX_{PE} is de-asserted the BBP holds TXRDY active until the last symbol containing data is modulated.

The reception process works as follows. There are also 4 signals associated with the BBP and MAC Receive Ports: RX_{PE} (receive enable), MDRDY (receive ready), RXD (receive data), and RXCLK (receive clock). The receive demodulator in the BBP is activated via RX_{PE}. When RX_{PE} goes active the demodulator scrutinizes I and Q for packet activity. When a packet arrives at a valid signal level the demodulator acquires and tracks the incoming signal. It then sifts through the demodulator data for the Start Frame Delimiter (SFD). MDRDY is programmed to go active after SFD is detected. This signals the MAC, allowing it to pick off the needed header fields from the real time demodulated bit stream. TXRDY is de-asserted afterwards. Figure 2.4 illustrates this sequence.

From this explanation, there are two signals (TXRDY and MDRDY) that exactly indicate when a frame exchange begins and finishes, as illustrated in Figure 2.5. More specifically, when the last bit of

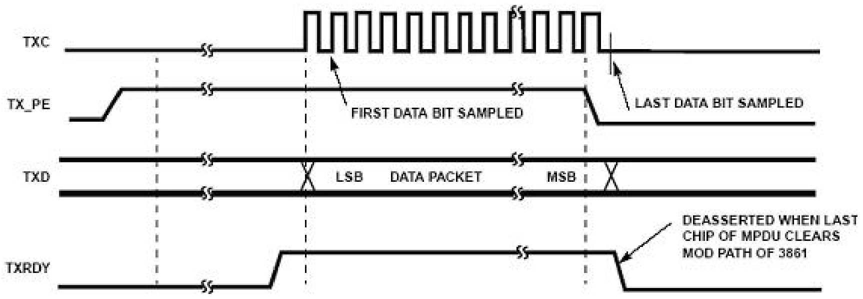

Figure 2.3: Bahavior of the extracted signals when transmitting a frame

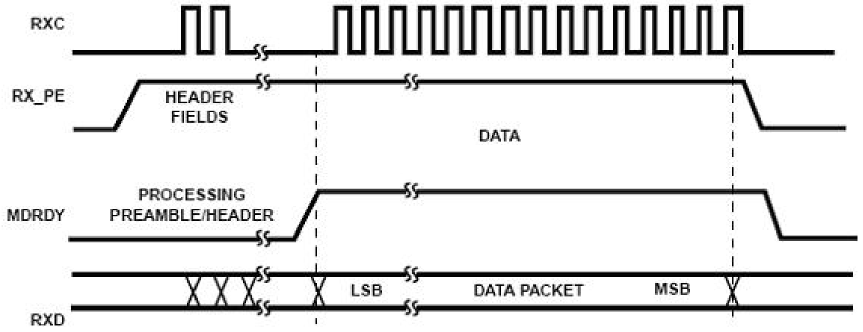

Figure 2.4: Bahaviour of the extracted signals when receiving a frame

the MAC data frame is transmitted the TXRDY signal goes to '0', and when the SFD in the reception of the corresponding ACK is detected the MDRDY signal goes to '1'. Thus the RTT in the proposed measurement mechanism is the time elapsed between the transmission of the last bit of the data frame and the reception of the last bit of the SFD of the corresponding ACK.

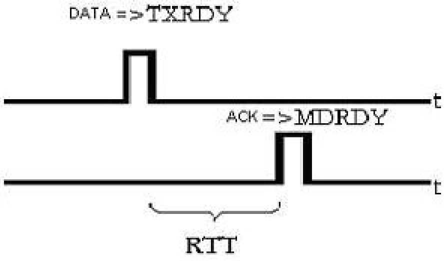

Figure 2.5: Signals indicating the frame exchange

The SFD is a 16 bits field consisting of the pattern 1111001110100000 that is used to mark the start of every frame at the physical layer; the SFD comes just after the 128 bits of synchronization and the overall 144 bits (Synchronization + SFD) compound the Physical Layer Convergence Procedure (PLCP). Then the accurate expression of the RTT at a distance i different to zero in the proposed method is:

$$RTT = 2 \cdot TOA + t_{PROC\_AP} + t_{TX\_PLCP}. \tag{2.7}$$

The differences with respect to the given generic expression of RTT (Equation 2.2) can be easily observed and understood. Firstly the transmission time of the data frame (t_{TX_DATA} in Equation

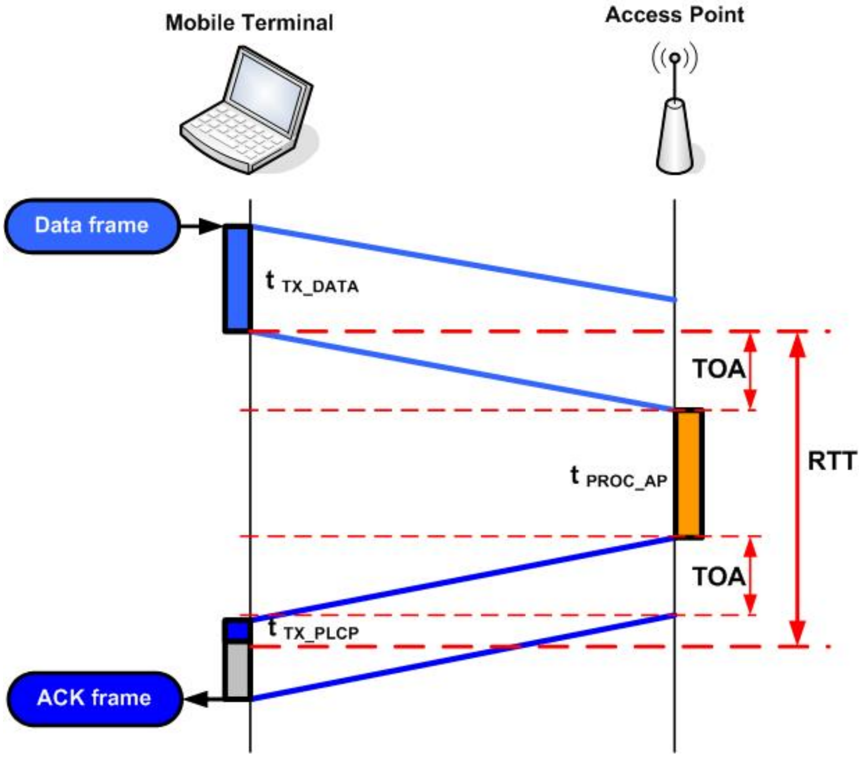

Figure 2.6: Scheme for the RTT in the proposed ranging method

2.2) does not appear in our expression because in the proposed method the hardware subsystem starts counting the RTT just once the transmission of the data frame has finished. Secondly in our method the RTT counting stops when the PLCP of the ACK frame has been received, without waiting to the reception of the whole ACK frame; hence in our expression instead of the transmission time of the ACK frame (t_{TX_ACK} in Equation 2.2) it is found the transmission time of the PLCP bytes of the ACK frame (t_{TX_PLCP}). The RTT in the proposed mechanism can be then depicted as follows:

Analogously, the RTT_0 now is defined by the following expression:

$$RTT_0 = t_{PROC\_AP} + t_{TX\_PCLP}. \tag{2.8}$$

On the other hand, the WLAN clock signal is accessed with the MCLK signal. The three signals (TXRDY, MDRDY and MCLK) can be extracted from the BBP chipset, as shown in Figure 2.7. In order to confirm the validity of this theoretical study, the TXRDY and the MDRDY pins are monitored with a logic analyzer, triggering with the falling edge of TXRDY and studying when the rising edge of MDRDY is. To this end ICMP Echo Requests are sent to the AP LinkSys Wireless-G 2.4 GHz. It is found that the RTT time is between 154.80 μs and 155 μs depending on the location of the mobile AP, with the expected good behavior of bigger RTT between TXRDY and MDRDY for bigger distances between both WLAN nodes.

The RTT counter module A simple circuit in charge of counting the RTT is then designed and implemented. It takes as inputs the three extracted signals from the WLAN card and returns as output the measured RTT through the parallel port to the laptop (i.e. to the software subsystem). For more details about this circuit see [46].

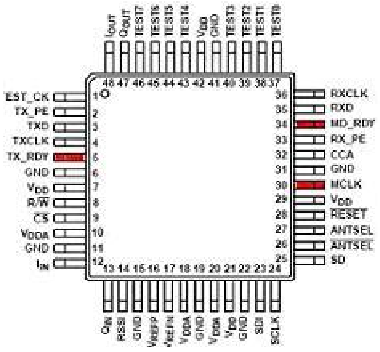

Figure 2.7: BBP chipset

The software subsystem

The software subsystem is composed of a script that runs on the laptop. This script sends ICMP Echo Requests to the AP, assuming that the IP address of this node is known. This way the transmission of a MAC data frame is induced. Once transmitted, the script synchronously waits for new available data (i.e. the obtained RTT value in the counter module) in the parallel port and then reads it. Finally it stores the RTT value in a file, so that it can be easily processed. The number of RTT measurements that performs the script can be configured. Figure 2.8 shows the complete lab prototype. For more details about the software subsystem see [46].

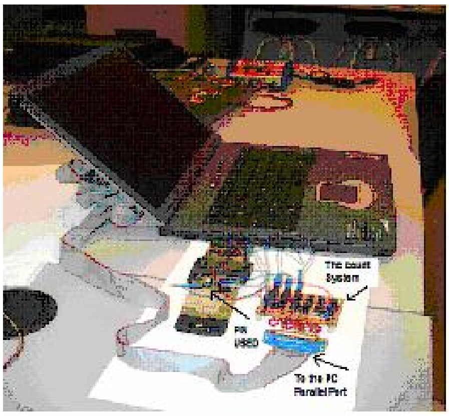

Figure 2.8: Developed lab prototype for ranging

2.3.4 The RTT measurements in practice

Sources of noise

RTT measurements are not expected to be stable due to several sources of randomness and time dispersion, such as:

- Discrete time quantification to one 44 MHz clock cycle.
- Delays due to the WLAN hardware electronics in the MT and the AP, as well as delays due to the additional hardware counter module in the MT.

- Drift of the clock in the MT during the measurement and relative drift between the clocks in the MT and the AP.

- The inherent characteristics of the indoor wireless radio channel [47], multi-path among them.

First experiments conducted with the lab prototype corroborate that RTT measurements present noticeable time variability. As an illustrative example the next table shows the obtained RTT histogram after performing an important number of RTT measurements at a distance of just few meters between both WLAN nodes. As can be seen, there is a group of measurements that form a compact lobe, concretely with RTT values ranging from 6804 to 6815 clock cycles, while others are far apart from that group. These other measurements do not have any sense and are obtained mainly due to errors or malfunctions of the time counter circuit, therefore they are filtered. Focusing on the valid measurements, the before mentioned noise and variability is evident: the range of 11 clock cycles (from 6804 to 6815) mean around 0.25 microseconds, equivalent to a dispersion of 75 meters in terms of distance.

RTT in units of 44 MHz clock cycles	Number of occurrences
6804	1
6805	11
6806	29
6807	77
6808	103
6809	154
6810	161
6811	135
6812	130
6813	90
6814	43
6815	12
7057	1
8618	1
9014	1
9899	1
10526	1
13753	1
14362	1
29992	1
38050	1
41734	1
42671	1
49718	1

Table 2.1: Series of RTT raw measurements

Impact of the radio channel The sources of error that affect the RTT measurements can be divided depending on its origin: some of them are due to the own employed RTT measurement mechanism and the last listed one due to the radio channel through which the signal travels. It is important to quantify the weight of each source of noise in order to better understand the behavior of the method. Thus an experiment to evaluate the impact of the indoor radio channel (mainly due to the multi-path) is performed. This experiment consists of comparing RTT measurements taken in a conventional indoor scenario with others obtained in a scenario with an ideal radio channel, i.e. a channel in which multi-path propagation of the signal does not exist. The ideal scenario corresponds to the anechoic chamber at UPC.

Measurements are carried out with the lab prototype and an IEEE 802.11b/g Linksys Wireless-G Broadband AP in b operation mode, in LOS situation between both nodes. A statistical comparison between sets of 1000 RTT samples taken in the two environments at the same distance is carried out. The distances studied are 0, 1, 3, 6, 9 and 12 meters. Detecting noticeable differences between parameters from the two scenarios would mean that the radio channel affects the performed RTT measurements. The RTT empirical histograms obtained in an anechoic chamber and in a real indoor environment at a distance of 3 m are shown in Figure 2.9. The RTT average figures are 6811.596 and 6811.598 clock cycles, respectively; the RTT standard deviations 1.984 and 2.035, and the number of correct measurements are 835 and 811.

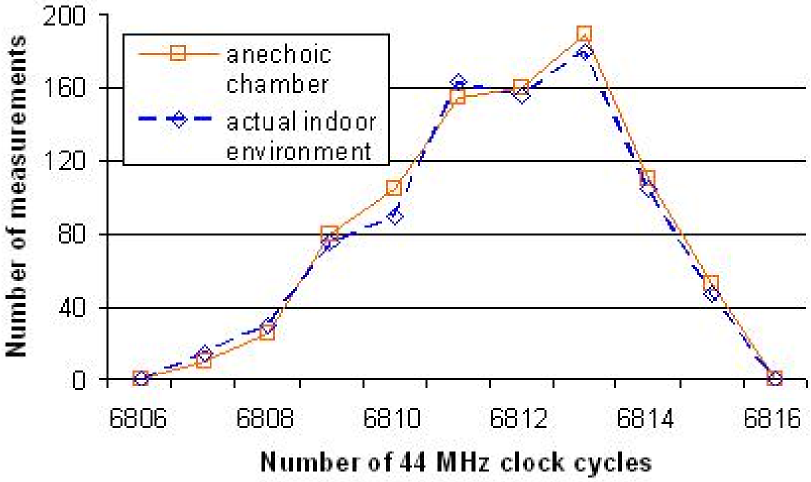

Figure 2.9: RTT histogram for an anechoic chamber and actual indoor environment at 3 m.

The comparison between both situations reveals that only slight differences exist. The measurements in the anechoic chamber are more accurate and stable: they show a shorter distance, smaller standard deviation and higher number of correct measurements. This behavior is also observed in the other tested distances (0, 1, 6, 9 and 12 meters). The RTT autocorrelation is negligible in all cases. The conclusion that can be extracted from the experiment is that the RTT measurements are practically the same in both environments, and therefore in practice the multi-path and other phenomena of the indoor radio channel do not add noise to the measurements. In other words, the observed noise and dispersion on the measurements is due to the inherent characteristics of the RTT measurement mechanism. However it has to be noticed that this does not purely and necessarily mean that the radio channel is not affecting the measurements, because maybe the noise produced by the channel is just disguised by the major magnitude noise produced by the own measurement system.

Correlation of the RTT measurements

The auto-correlation of the series of 1000 RTT measurements is obtained. As mentioned before, the auto-correlation is negligible for all the series (see Figure 2.10). This means that there are not either repetitive patterns or any kind of periodicity on the obtained measurements of a series. In addition, the cross-correlation between all the pairs of obtained RTT series is obtained, with the same result. This corroborates the observations given with the auto-correlation.

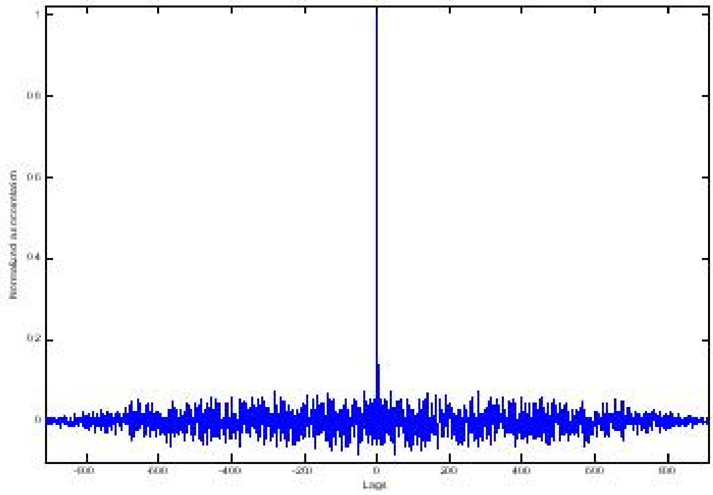

Figure 2.10: Auto-correlation of a series of 1000 RTT measurements

Statistical processing to mitigate the noise

In order to obtain an accurate RTT estimate, mitigating as much as possible the observed noise that affects the RTT measurements, a simple statistical processing is proposed. Thus RTT is dealt with as a Random Variable (RV); this means first of all that is not enough to perform just one RTT measurement for estimating the RTT between the mobile device and the AP, but several samples of the RTT RV have to be collected (i.e. several RTT measurements must be performed). Then a proper statistical estimator can be applied over the samples in order to obtain the RTT estimate. Following Equation 2.11, estimates for RTT_i and RTT_0 are obtained and subtracted to then obtain the TOA.

Number of RTT measurements It is important to know the number of RTT measurements needed to estimate the RTT_i and RTT_0. This number is relevant in order to find a reasonable trade-off between bandwidth used, time employed and accuracy obtained. Given that the RTT is a RV, when a series of RTT measurements is performed it can be said that statistically some RTT samples are obtained from the total RTT population. How certain the average (obtained from the samples) is, given a specific number of samples, is a commonly accepted indicator of the reliability and suitability of this number taken from the total population. This idea of "how certain is" is reflected in the confidence interval for a certain percentile of the time. In this way it is known how certain the average computed from the samples as an estimate of true average (average obtained from the total population) is. Normally it is used a confidence interval of 99, 95 or 90$\%$.

The formula of the confidence interval depends on the premises that can be assumed regarding the RTT distribution and a minimum number of samples needed that is accepted. In this case, since RTT distribution is not normal and 100 is accepted as the minimum number of samples. In addition, the standard deviation of the population is not known, so an estimator has to be used for estimating it. Given these premises, the formula of the average's confidence interval for a confidence level of 95$\%$ of the time is:

$$\eta \in (\bar{x} \pm z_{0.975} \cdot \sqrt{S^2/n}), \tag{2.9}$$

where η is the estimated RTT average, $\bar{x}$ is the population average, S the estimated standard deviation from the population and $z_{0.975}$ the z function value for a confidence level of 95$\%$. The units for this confidence interval are 44 MHz clock cycles. From Equation 2.9, n can be deduced:

$$n = (2 \cdot z_{0.975} \cdot S/A)^2, \tag{2.10}$$

where A is the width of the confidence interval. The value of the z function for 0.975 is 1.96, the estimated standard deviation from the population (S) is 2. Taking into account that every 44 MHz rising clock implies a distance of 7 m., it is considered that only values of A under 0.5 (it is 0.25 rising clocks around the population average) can be accepted. A result of n = 246 was obtained; being aware that usually a small portion of the performed RTT measurements are not valid (due to errors of several types), n = 300 seems to be a conservative figure to accurately estimate the RTT.

RTT estimator As explained before, estimates for RTT_i and RTT_0 must be obtained for then obtaining the TOA estimate between the client device and the AP. The RTT_0 is obtained just once placing the MT and the AP together and then is stored and ready to be used for estimating the TOA whichever the distance is. Regarding RTT_i (i.e. the RTT at a distance i) it is obtained in real time as a first step of each performed distance estimation process defined by the proposed ranging method. From a statistical point of view, it is aimed to accurately estimate the TOA (i.e. the distance) given the following available collected data: a set of samples of the RV RTT_0 and another set of samples of the RV RTT_i. It is known that the RTT samples are contaminated by noise that produces time dispersion in the obtained RV distribution, but the characteristics of this noise are not known. One hypothesis can be that the additive noise produces time dispersion but not bias (i.e. additional time delay) to the measurements; in that case the noise could be mitigated employing the same suitable estimator for RTT_0 and RTT_i. On the other hand, another possibility is that the noise, in addition to the observed time dispersion, adds a positive error bias to the measurements; in that case different estimators for RTT_0 and RTT_i should be probably taken in order to mitigate this bias.

Both commented hypothesis with several RTT estimators are aimed to be tested with real measurements obtained with the lab prototype and an AP, in order to find the estimators that provide better ranging accuracy. The candidate statistical estimators that are tested are: average (η), half range, mode, minimum value, and average minus n times the standard deviation ($\eta - n \cdot \sigma$). The use of the minimum is grounded on the fact that the smallest RTT is expected to correspond to the shortest (i.e., direct) path. The $\eta - n \cdot \sigma$ estimator is based on the same principle but discards the bottom values as outliers; the parameter n should be determined heuristically and would only be useful if stable with distance

2.4 Performance assessment

2.4.1 Finding the best estimator for the RTT

In order to test the different RTT estimators, an important number of RTT series obtained at different distances is needed. The series are obtained indoors in LOS situations at distances increasing by 3 m between 0 and 30 m. Both the lab prototype and the AP (an IEEE 802.11b/g Linksys Wireless-G Broadband in b operation mode) are placed 1.5 m. above the ground in order to preserve the Fresnel zone. For each distance, several series of 1000 RTTs are collected. In Figure 2.11 the obtained RTT empirical histograms for 0, 6, 12, 18, 24 and 30 m are depicted. As expected, the histograms shift to the right as the distance increases. It can be observed again the high time-variability of the RTTs of each series. Given these available RTT measurements, the distance estimate for each tested distance (except for 0 meters) is calculated testing the different candidate estimators. The employed formula for estimating the distance i is:

$$d = c \cdot TOA = c \cdot \left(\frac{RTT - RTT_0}{2}\right)\left(\frac{1}{f_{clk}}\right), \tag{2.11}$$

where f_{clk} corresponds to the 44MHz of the WLAN card clock.

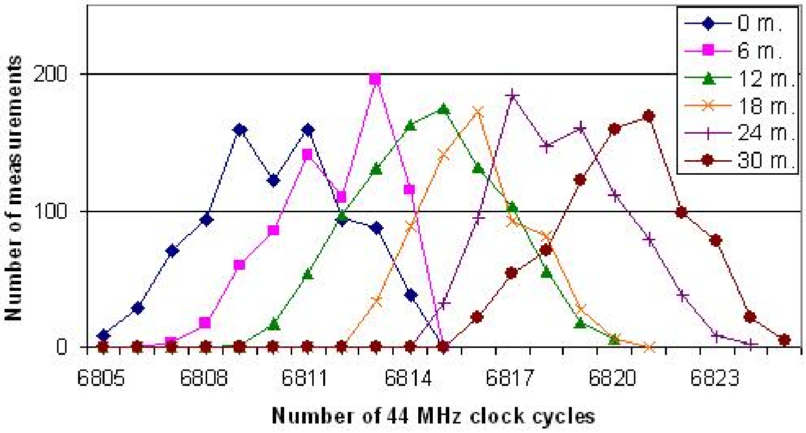

Figure 2.11: RTT histograms for 0, 6, 12, 18, 24 and 30 m.

First the approach of employing the same estimator for RTT_0 and RTT_i is tesded. Table 2.2 shows the average of the distance estimation errors incurred for all distances under study for each tested RTT estimator for RTT_0 and RTT_i. It can be observed that in all cases the distance estimates are longer than the actual distances. This means that the RTT measurements, in addition to the noise that causes their time dispersion, are contaminated by a positive error bias that has not been eliminated with the proposed statistical processing. According to this it seems that the idea of using the same estimator for RTT_0 and RTT_i is not suitable for reaching the best ranging accuracy with the proposed RTT measurement method. Therefore the alternative explained hypothesis is assessed.

Estimator	Error (m.)
Average (η)	2.82
Half range value	4.43
Mode value	2.86
Minimum	2.72
η-σ	2.81
η-2σ	2.80
η-3σ	2.79

Table 2.2: Average ranging errors in LOS situations using the same estimator for RTT_0 and RTT_i

Assuming the conjecture that the different sources of noise provoke a positive error bias to the RTT measurements, the approach is taking different estimators for RTT_0 and RTT_i to mitigate this bias: since RTT_0 is subtracted from RTT_i, the idea is using an estimator for RTT_i that allows smaller values of the obtained RTT distribution with respect to the estimator employed for RTT_0, so that the obtained TOA value from the subtraction is smaller than with the first tested approach. Firstly, since RTT_0 is purely a MAC processing time the average (η) is supposed to be a good estimator for it. Then for RTT_i an estimator "smaller" than the average must be used. This idea can be also explained because such an estimator could fit better for a RV than includes propagation time, which is the case of RTT_i, due to the high relevance of the lowest measurements obtained.

The tests consist again of estimating the distances from 3 to 30 meters with the performed measurements and the provided formula for estimating the distance i, employing the η estimator for RTT_0 (i.e. η_0) and testing the other estimators for RTT_i. Since both σ and the error when using η as an estimator increase with distance, estimators of the type $\eta - n\sigma$ provide lower errors than other estimators. The best value for n (i.e., the one that provided the best accuracy) is heuristically determined as n=1/3, so that $\eta - \sigma/3$ is selected as the estimator for RTT_i. Then the TOA (in units of clock cycles) expression according to Equation is finally:

$$TOA = \frac{(\eta_i - \frac{\sigma_i}{3}) - \eta_0}{2}, \tag{2.12}$$

and therefore the distance estimation formula with the proposed TOA-based ranging method for WLAN is the following:

$$d = c \cdot TOA = c \cdot \left(\frac{(\eta_i - \frac{\sigma_i}{3}) - \eta_0}{2} \right) \cdot \left(\frac{1}{f_{clk}} \right), \tag{2.13}$$

2.4.2 Ranging results

Once selected the statistical processing of the RTT measurements, the ranging method has been totally defined: series of 300 RTT measurements, statistical processing with the selected estimators for RTT_0 and RTT_i and finally the TOA (and distance) estimation. Now a set of distance estimates employing again the lab prototype with the ranging method is performed in order to assess the ranging accuracy of the proposed method. For the AP used in this study (an IEEE 802.11b/g Linksys Wireless-G Broadband in b operation mode), η_0 is 6810.28 44MHz clock cycles. The tested distances are again between 0 and 30 m. Both the lab prototype and the AP are placed 1.5 m. above the ground in order to preserve the Fresnel.

Table 2.3 shows the obtained ranging results. Contrary to what could be expected, the ranging error does not increase with distance: the RTT dispersion (i.e. the RTT standard deviation) and the positive error bias of the RTT average increase with distance, but the ranging accuracy remains stable because the use of the $\eta - \sigma/3$ estimator allows the compensation of both statistical parameters. The average of the obtained absolute errors, taking into account all distances, is 0.81 m. It has to be noted that the proposed ranging method, especially with the processing of the RTT samples, is able to provide an improvement over the theoretical distance resolution of 7 m, as discussed previously, in addition to mitigating the other sources of randomness, noise and time variability in the RTT-based distance calculation.

Dist(m)	RTT Std. Dev.	RTT_i est. $\eta - \sigma/3$	Distance estimate (m)	Abs. error (m)	Rel. Error (%)
3	2.03	6811.05	2.62	0.37	12.47
6	2.12	6811.58	4.45	1.54	25.80
9	2.23	6812.71	8.28	0.71	7.89
12	2.39	6813.66	11.52	0.47	3.93
15	2.33	6814.35	13.88	1.11	7.44
18	2.33	6815.38	17.37	0.62	3.45
21	2.35	6816.73	21.96	0.96	4.58
24	2.43	6817.68	25.21	1.21	5.06
27	2.41	6818.37	27.54	0.54	2.02
30	2.53	6819.25	30.55	0.55	1.83

Table 2.3: Ranging results in LOS situations using the average estimator for RTT_0 and $\eta - \sigma/3$ for RTT_i

2.4.3 Obtained vs theoretical RTT

As explained in the implementation of the proposed RTT measurement mechanism, in the proposed TOA-based ranging method the RTT at a distance i different to zero corresponds to the following expression:

$$RTT_i = 2 \cdot TOA + t_{PROC\_AP} + t_{TX\_PLCP}. \tag{2.14}$$

It is important to validate the obtained RTT estimates with the theoretical RTT time that is expected from the given expression of RTT_i. Let's start with the theoretical RTT. t_{PROC_AP} corresponds to the SIFS time period, i.e. 10 microseconds. The PLCP consists of 144 bits that is always transmitted at a constant rate of 1Mbps (the transmission rate is specified in every frame in a posterior frame called "signal"), thus t_{TX_PLCP} corresponds to 144 microseconds. TOA has three orders of magnitude less than the mentioned times, because TOA corresponds to tens of nanoseconds. Adding these terms, the result is that the theoretical RTT is around 154 microseconds. This corresponds with the obtained RTT, because the obtained RTT estimates are around 6815 clock cycles (see Table 2.1 measuring with a 44 MHz clock, which corresponds to the theoretical value in microseconds. On the other hand, in Table 2.3 can be observed that the increment each 3 meters between the RTT estimates correspond to one clock cycle approximately. This value is totally expected because in one clock cycle (44 MHz clock) the signal travels 6.81 meters, which corresponds to around 3 meters taking into account that two-way ranging measurements are carried out.

2.4.4 Ranging probability distribution

In order to obtain a complete characterization of the ranging accuracy provided by the proposed method, the Probability Density Function (PDF) of the distance estimation is calculated. The PDF is obtained normalizing an empirical histogram, which is calculated taking into account a large number (500) of distance estimations at a fixed distance (11 m.) -this is performing 500 series of RTT measurements- carried out using the lab ranging prototype with the described statistical processing. RTT measurements are carried out indoors at 11m. between the lab prototype and the AP, in LOS situation between them. As in the other measurements campaigns, both the lab prototype and the AP are placed 1.5 meters above the ground in order to preserve the Fresnel zone. Since it has been proved that the ranging error does not vary with the distance, taking the measurements at one specific distance is enough to characterize the accuracy of the proposed method. Ideally, all the distances measured should be 11 m; however due to the slight instability provoked by remaining randomness in the measuring mechanism, the ranging method obtains distances from 8.80 m to 12.80 m. Comparing the resulting PDF with known probability distributions, it is found that the one that best fits is a Gaussian distribution with η = 11.12 m. and σ = 0.84 m., as can be appreciated in 2.12.

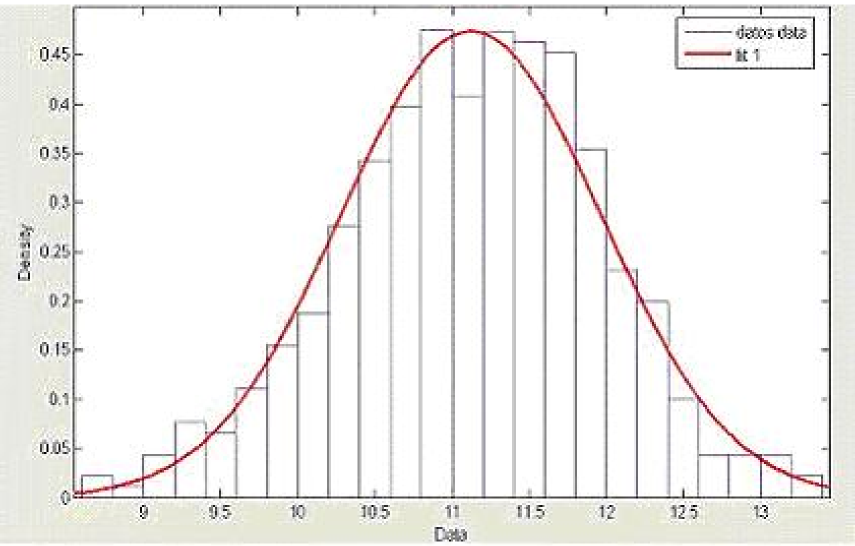

Figure 2.12: PDF of distance estimation with the proposed method

2.5 Conclusion

Distance estimation (commonly known as ranging) is the essential and more challenging step in TOA-based location using WLAN. This chapter has presented an innovative TOA-based ranging method between WLAN nodes. After an accurate study of the state-of-the-art in WLAN TOA-based location in order to identify the limitations of such methods, the intention of the proposed method is going a step forward overcoming most of them. The general objective is achieving good ranging performance while reaching a high degree of cost-efficiency and simplicity of deployment, taking the major advantage of the characteristics of the under-laying WLAN protocol.

The adopted approach is based on performing RTT measurements using standard data/control frames at the IEEE 802.11 MAC layer. The key issue is how to perform the time measurements given the limited capabilities of the WLAN standard equipment. Since on one hand complex solutions that entail deep hardware modifications are avoided, and on the other hand pure-software solutions that make use of standard time-stamps are supposed to not provide enough accuracy, the proposed method consists of performing the time measurements at hardware level using available signals in the chipset of the client's WLAN interface. This way it can be easily implemented as an upgrade of the WLAN card firmware, just adding the time-stamps of the transmission and reception of the MAC frames employing as time base the available clock at the WLAN card and at the same time a good ranging accuracy and stability is feasible because the hardware level access.

A lab prototype implementing the proposed mechanism to measure the RTT has been build. Extensive measurements campaigns have been carried out and the statistical properties of the measurements have been analyzed. It has been seen that the measurements are not deterministic but noticeably contaminated by different sources of noise related with the own measurement system that provoke time dispersion and a positive error bias. RTT is then dealt as a random variable and estimated by means of a proper simple statistical processing that palliates as much as possible the existing noise. The achieved ranging accuracy is around one meter of error, fact that means a noticeable success. The important innovative component of the method and its good results have caused that during the second half of 2007 and the subsequent 2008 some researchers have shown noticeable interested on it. As illustrative examples, [40] and [41] present ranging approaches that take our ranging method as main reference and starting point for their work.

This contribution proves that time stamping of IEEE 802.11 standard frames using the available clock in the WLAN card would suffice to achieve accurate ranging capabilities, so that solutions that provide at the same time very good ranging performance and simplicity can be feasible. Given this fact, a technical contribution that mainly propose to integrate these new time-stamping functionality is submitted to the IEEE 802.11v [44] standard task group. This contribution is included in Appendix B.

Chapter 3

TOA-based ranging technique purely software

The TOA-based ranging technique presented in Chapter 2 provides good results but it requires manipulating the hardware of the client's WLAN interface. Although this modification is slight and could be probably implemented as a firmware update, prevents from deploying the achieved technology to the mobile devices as a pure-software solution. The motivation of achieving a pure-software solution is important and corresponds to several points. First, less complexity and cost of the implementation, avoiding the re-design and enhancement of the WLAN interfaces. Secondly, more flexibility in the deployment of the solution. Last, the independence of the ranging solution from the under-laying hardware.

This chapter presents the research work on a TOA-based ranging technique that completely avoids hardware or firmware modifications over standard WLAN equipment. This work started in 2007 and is still active since it is one of the main research interests nowadays. The chapter is organized as follows. First the specific objectives of this work, which translate on requirements of the pursued ranging method, are explained. Then, since the most relevant related work about TOA-based ranging with WLAN has been already explained in Chapter 2, the specific software-based proposals existing in 2007 are explained. This is essential to understand why the current design approach is adopted for our solution. After that the details of the researched ranging method are provided presented, including the exploration of the different alternatives given a specific problem. Finally conclusions are given.

3.1 Objectives

The objective of the research work presented in this chapter is assessing the feasibility of achieving good ranging performance, i.e. close to the one provided by the hardware-based technique presented in Chapter 2, by means of a technique that can be totally implemented with software over standard consumer WLAN equipment. The mentioned good ranging performance can be summarized with these specific goals:

- Good ranging accuracy, enough to satisfy the requirements of most LBS.
- Low complexity of implementation.
- Easy deployment.
- Immunity to the environmental changes.
- Low bandwidth consumption.
- Low computational cost.

It can be observed that the goals of this research work are very ambitious. The good results obtained with the hardware-based ranging technique and the gained experience during this first work encouraged me to face these more stringent requirements, with the important motivation of exploring the current limits of WLAN TOA-based ranging.

3.2 Related work

When exploring TOA-based ranging with WLAN totally avoiding manipulations of the under-laying hardware, the general adopted approach consists of performing two-way ranging (i.e. TOA estimation from RTT measurements) at MAC or network layer, being the key point again how to perform these time measurements. Two basic alternatives exist. From one hand, using standard hardware time-stamps with a resolution of 1 microsecond that most of WLAN cards record (i.e. at hardware level) when MAC frames are received. On the other hand, using mechanisms at operating system level capturing the needed time-stamps for the RTT at the driver of the WLAN interface, when the hardware interrupt of the WLAN card notifies the operating system about the transmission or reception of a frame.

In 2007 when my research on the pursued software-based ranging technique started, the existing reported experiments about pure-software TOA ranging with WLAN belonged to the former group. The reported results evidenced that, although the RTT measurements are taken at the more proper point (hardware level), the low resolution of the employed time-stamps (1 μs error corresponds to 300 meters) provokes big quantization error that does not allow reaching very good ranging accuracies. A representative contribution can be found in [38]. Given this existing research background, it was decided to explore the other alternative of design.

3.3 Analysis

3.3.1 Basic design issues

The more generic design decisions about the ranging approach are the same with respect to the hardware-based ranging approach explained in Chapter 2, because the philosophy of taking the major advantage of the characteristics of the under-laying WLAN protocols is also followed now. Thus, a two-way ranging approach (i.e. RTT) employing the IEEE 802.11 MAC data-ACK frame exchange is adopted: the client device sends an unicast data frame to the AP and counts the time until the corresponding ACK answer from the AP is received. The employed frames belong to the MAC layer because this is the lowest network layer accessible from the client device via software; the lowest possible layer is preferred in order to avoid extra delays due to processing between network layers. The RTT at distance zero (RTT0) has to be previously obtained and stored in the mobile device in order to estimate the TOA from the RTT at a distance i different to zero. Then, as in the hardware-based approach, the TOA can be calculated with the following expression:

$$TOA_i = \frac{RTT_i - RTT_0}{2}, \tag{3.1}$$

3.3.2 Mechanism for measuring the RTT

As explained before, the approach of capturing the needed time-stamps at operating system level from the driver of the WLAN interface is explored. The idea is measuring the RTT via a software mechanism that makes use of the existing hardware of an off-the-shelf client mobile device enabled with WLAN. The first key feature that is exploited is the optimum time resolution that the CPU clock of the MT can provide as a time-base to timestamp the transmission and reception of frames to measure the RTT (a resolution of 1 ns could be achieved with a 1 GHz CPU). This was briefly mentioned in [48]. The other mechanism that we take advantage of is that some current WLAN drivers can be modified and rebuilt so that the necessary events to obtain a RTT measurement (i.e. the transmission and reception of a MAC layer frame) can be accessible via software. The proposed mechanism to obtain the RTT is time-stamping in the WLAN client device the instant when the MAC data frame is transferred to the physical layer for transmission and the instant when the ACK frame is received at the MAC layer; then both instants are subtracted. The difference between the time stamps is an estimate of the RTT.

Measurements at WLAN driver level

Since hardware-based mechanisms are intentionally avoided, the adopted solution is accessing and time-stamping the above-mentioned events from the code of the driver that controls the WLAN interface. The driver's code implements the IEEE 802.11b/g MAC protocol. It is important to notice that, although the WLAN card driver allows direct access to the WLAN MAC level events at the code level (see the architecture overview in Figure 3.1), the Operating System (OS) of the MT interfaces with the WLAN card via hardware interruptions. The driver code is executed once the OS has handled the interruption and the OS process scheduler has allocated the CPU to execute the corresponding network interruption process. Despite the higher scheduling priority of the hardware interruptions with respect to most other processes, this means that a slight time delay exists for the instant between the reception of an IEEE 802.11 frame at MAC level and the moment that this reception is time-stamped by the OS in the driver code. In practice, this entails non-negligible measurement error when measuring the RTT. Since this delay in the attention of the interruption (i.e. the interrupt latency) is supposed to partially depend on the load in the CPU or the state of the OS, it is expected that it would have noticeable variability. Then the theoretical problem is that the RTT measurements can present big randomness since they are affected by the mentioned interrupt latency.

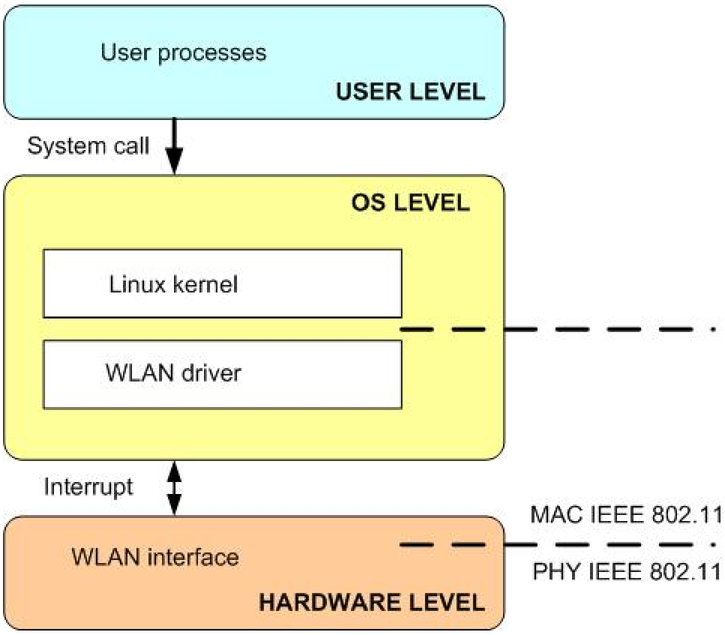

Figure 3.1: Overview of the OS-IEEE 802.11 architecture

Time base for measuring the RTT

As commented before, the first idea is employing the CPU clock of the MT as time-base to time-stamp the frames in the RTT measurements, because it provides very good time resolution and it is accessible from the code of the WLAN card driver. Theoretically, this resolution would achieve ranging errors below one meter. The Time Stamp Counter (TSC) is a 64-bit register that counts cycles of the CPU oscillator, and it has proved to be reliable to count time differences [49]. It is stored in the EDX and EAX 32-bit registers and can be accessed by programmers with the *rdtsc()* (read TSC) function, which returns its value (the 32 least significant bits, stored in EAX) at that specific instant in the number of clock cycles. The call to *rdtsc()* is added to the code of the WLAN driver in the proper functions, as will be seen later. Once the returned values from *rdtsc()* for the data frame transmission and the ACK frame reception are subtracted to obtain the RTT measurement in clock cycles units, it is converted into time units with the CPU frequency. However, in addition to the CPU clock, the clock of the WLAN card is also available and accessible form the WLAN card driver code through the *ath5k_hw_reg_read()* call. Despite its smaller resolution, this latter clock is also taken into consideration in my research due to its probable better stability since it is less affected by the OS mechanisms. Thus, in order to reach the better ranging performance as possible, it is considered useful to perform a comparative study about the performances of both clocks when measuring the RTT. This study is shown in the next chapter, which explains the design and implementation issues of a lab prototype that allows assessing the proposed TOA-based WLAN ranging technique in practice.

3.3.3 The lab prototype

As in the case of the hardware-based ranging prototype, the current lab prototype is implemented in a laptop, as it can be easily equipped with a WLAN card and easily manipulated. Since the RTT measurements are carried out in the mobile device, i.e. the lab prototype in this case, the employed AP is a commercial one without any modification or special configuration. The lab prototype is intended to implement the following basic functionality:

- First, it sends a MAC data frame to a known AP.
- Then it measures the elapsed time between the last bit of the data frame is sent and the first bit of the ACK is received from the AP.
- When the data-ACK exchange is finished, it stores the measured RTT.
- A new RTT measurement to the same AP can be performed if it is required.

As explained during the last chapter, capturing the time-stamps for the RTT measurements at driver level entails that the operation of the OS of the MT probably affects the measurements in a high degree. Thus in this technique the configuration of the OS becomes a crucial issue when implementing the ranging lab prototype. The adopted OS to run on the laptop is Linux, because it allows the manipulation of the source code and therefore the optimization of the system for the best possible ranging performance. In this chapter different possible configurations of the Linux OS are explained. Notice that this was totally omitted in the hardware-based ranging prototype because the OS had no impact on the ranging performance. First of all the developed software -including the modification of the specific WLAN driver and the needed scripts- is explained.

Developed software

The employed WLAN driver is the Madwifi one [50]. Madwifi stands for Multiband Atheros Driver for Wireless Fidelity, and it is a driver developed for Linux that is compatible with WLAN interfaces based on Atheros chipsets. This driver has been chosen because its code is open-source and in addition is compatible with a big number of wireless interfaces. This driver provides total compatibility with Linux, because it has been specifically designed for this OS and allows the use of the Wireless extension API, which allows configuring the WLAN interfaces with the most common Linux commands (ifconfig, iwconfig, iwlist...)

The driver code allows accessing the events of the IEEE 802.11 MAC layer of the WLAN interface. More specifically the needed events for calculating the RTT, i.e. the transmission of a data frame and the reception of the corresponding ACK, can be accessed. The key component of the developed ranging software is the modification of the Madwifi driver code for time-stamping these two events using an available clock as time-base and storing the calculated RTT measurements in a way that they can be subsequently processed. The other software component consists of a script that is in charge of inducing the transmission of the MAC data frames to the AP and processing the obtained RTTs to finally estimate the TOA and the distance.

Driver modification This is probably one of the key points of the developed application, because it has direct impact on the achieved ranging performance. The idea is time-stamping the desired MAC events as soon as possible once the hardware interruption has been handled by the OS, in order to minimize as much as possible the delay provoked by the interrupt latency. This way the randomness of the time delay that affects the RTT due to the OS operation could be reduced, making the RTT measurement as much deterministic as possible.

It is important to know how interruptions work in Linux. The OS stores in a table the information of each programmed interruption. When a hardware interruption occurs, the table is consulted in order to assign a handler routine to the interruption. This handler routine has a common part for all the interruptions and another one that is specific for each of them, which basically consists of a portion of driver code of the device that has provoked the interruption. Before the driver code is executed, the interruption is disabled. The interruption handling is executed as a system thread with higher priority with respect other system or user processes. The following Figure 3.2 illustrates this general explanation.

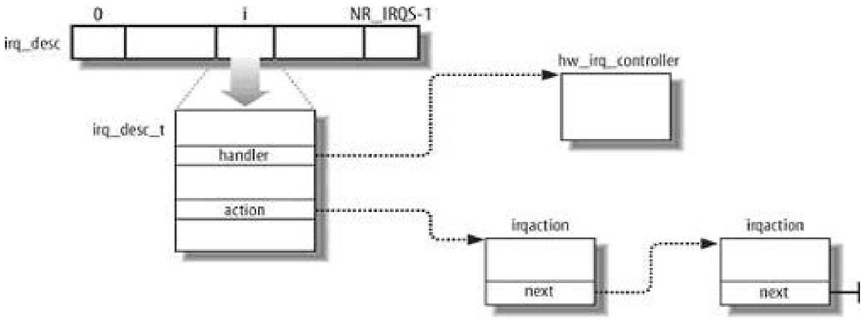

Figure 3.2: IRQ descriptors

Entering more into detail, when the interruption is received at OS level the handler is immediately executed, but the specific treatment of the interruption is queued in the Linux *task_queue*. Then the execution returns from the interrupt and the CPU is assigned to the task that was being executed before the interruption arrived. The process execution scheduler of the OS will decide later when the specific code of the driver can be executed, this is depicted in the next Figure 3.3. For a deeper explanation of the interruptions with the Linux OS please see [51, 52].

Given this explanation, it can be clearly seen that although the hardware interrupt has maximum priority and is immediately attended, the later execution of the corresponding specific driver function can suffer an extra delay that strongly depends on the execution state of the OS and other processes. Then it is important that the time-stamp is performed before this extra delay occurs, it is at the first possible point. This is feasible capturing the time-stamps in the general interrupt handler of the driver (ath_int function). The implementation consists of storing in a variable the time-stamp at this point, in the general interrupt handler, accessing the TSC register of the CPU or the WLAN clock as can be seen in the next piece of code.

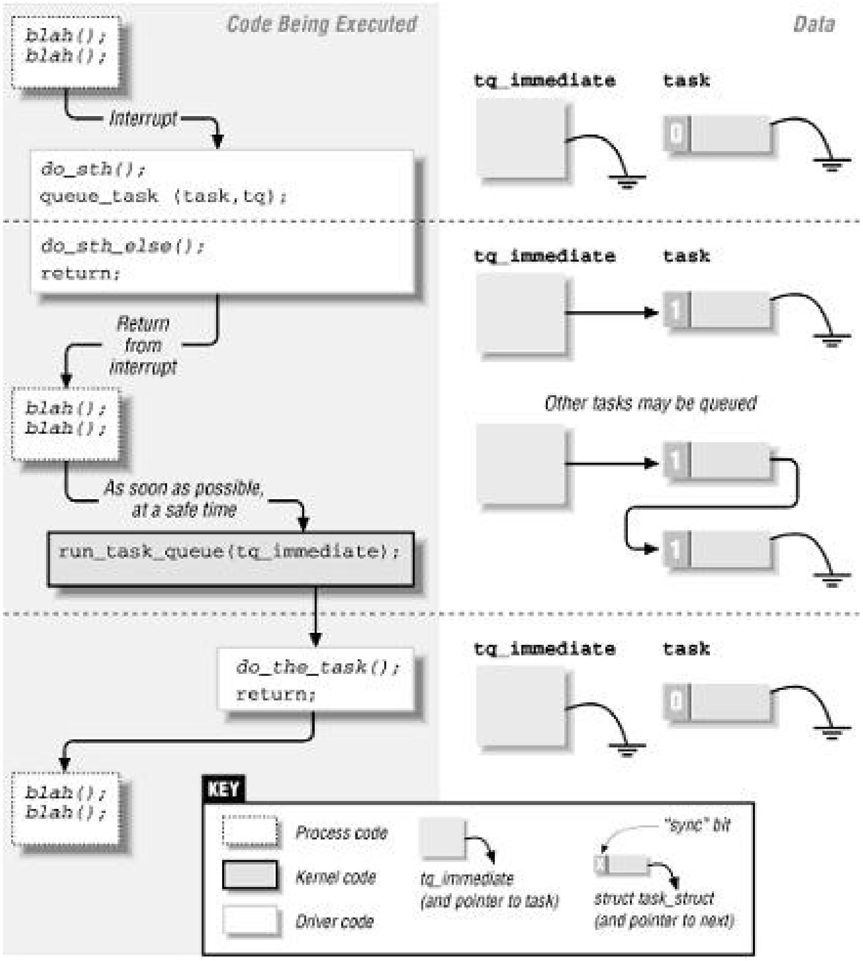

Figure 3.3: Code execution during an interrupt

```
/*
 * Interrupt handler.  Most of the actual processing is deferred.
 */
irqreturn_t
#if LINUX_VERSION_CODE >= KERNEL_VERSION(2,6,19)
ath_intr(int irq, void *dev_id)
#else
ath_intr(int irq, void *dev_id, struct pt_regs *regs)
#endif
{
struct net_device *dev = dev_id;
struct ath_softc *sc = dev->priv;
struct ath_hal *ah = sc->sc_ah;
HAL_INT status;
int needmark;

//////////////////////////////////////////////
rdtscl(clocks);
wlan_clocks = ath5k_hw_reg_read(sc,AR5K_PROFCNT_CYCLE);
//////////////////////////////////////////////
```

Later when the interrupt is really handled in the corresponding specific function of the driver, it is decided if the previously captured time-stamp corresponds to an event that has to be taken into account for the RTT or if it has to be discarded. This is done in the function that treats the frames receptions ($ath\_rx\_tasklet$), checking if the received frame is an ACK that has our interface as destination and controlling that the frame has not suffered retransmissions, fact that means discarding the time measurement in order to avoid wrong RTT measurements. If the conditions are satisfied, the RTT is calculated subtracting the reception time-stamp value from the previously stored transmission time-stamp. An ACK frame can be distinguished checking the frame type ("control") and subtype ("ACK") of the MAC header (see next Figure 3.4); in addition it is checked that the destination MAC address corresponds to our WLAN interface. This way the targeted frames for the RTT calculation (data and ACK) can be filtered from the others.

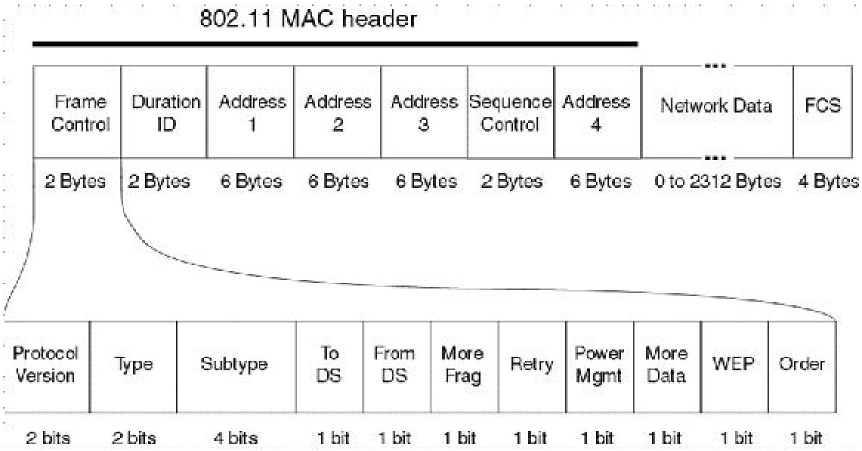

Figure 3.4: Header format of the MAC IEEE 802.11 frame

Once the RTT measurement is obtained, it has to be transferred to the application -which is executed on the same MT- that calculates the distance between the prototype and the AP. Since it is not possible to use the typical functions from the $stdio.h$ libraries (e.g. print, scanf...) from the driver's code, the adopted mechanism is sending the RTT figure as kernel message with the $printk$ function.

This function can be used in a similar way as the $printf$ function, with the additional option of assigning a certain priority to the message; the eight priority levels that are available can be found in the header file *include /linux/kernel.h* (see [Levels Linux] for detailed information). In order to minimize the delay that could affect subsequent RTT measurements, in our case the highest priority level ($KERN\_EMERG$) is employed. Then the sent messages, i.e. the RTT measurements, are captured by means of the $syslogd$ daemon [53]. The execution of this daemon is automatically started when the OS boots and is in charge of storing the logs about the system operation, receiving the messages sent from different parts of the system and handling them according to the configuration file $/etc/syslog.conf$. I configured this file so that the $kern\_emerg$ messages are stored in a text file that I specifically created to this end. All the modifications done on the Madwifi driver are detailed in the Appendix C.

Ranging script In addition to the WLAN driver modifications, the ranging software is composed by a script that is in charge of two main tasks: inducing the transmission of the MAC data frames to the AP, and processing the obtained RTT measurements at a certain distance to finally estimate the TOA and the distance. As it is shown later, one RTT measurement is not enough to accurately estimate a distance and therefore several RTT measurements have to be carried out and processed. In this latter task the script takes the RTT measurements from the before explained $/etc/syslog.conf$ file and applies a simple processing to them in order to estimate the TOA and the distance. This processing mainly consists of filtering the wrong RTT measurements and applying a good estimator over the resulting ones, and it is detailed later when the RTT measurements obtained with the lab prototype are analyzed.

On the other hand, the former task is accomplished sending Internet Control Message Protocol (ICMP) echo requests to the AP with the ping command. This can be easily explained with an example of the employed complete command:

```
ping 192.168.0.50 -i 0.1 -s 0 -c 1000 -W 0.0005
```

The first parameter is the IP address of the AP. The next one, -i 0.1, specifies the time (in seconds) between consecutive ping transmissions (1 second by default). An alternative to reduce at maximum this interval is the flooding optin (-f), but in that case the high transmission rate could affect the capturing of the time-stamps and therefore the accuracy of the RTT measurements; in our case 0.1 seconds represents a good trade-off between accuracy and latency. The next parameter (-s 0) indicates the datagram length, which is set to the minimum in order to reduce as much as possible the transmission time. The last two parameters indicate the number of pings to be sent and the transmission time-out.

The script schedules both tasks as follows. It opens the file where the RTT measurements have been stored, keeps the pointer to the end of the file and closes the file. Then invokes the ping command so that the new series of RTT measurements are performed and after that it opens the file again to collect the new values.

Configuration of the OS

First of all, the employed Linux distribution corresponds to Ubuntu [54], due to its friendly working environment and its recent kernel enhancements. Given that the operation of the OS is expected to affect the RTT measurements, it is important to evaluate different possible configurations of the OS in order to obtain the better ranging performance as possible. In the next paragraphs the assessed configurations are commented.

Basic configuration The first considered configuration is the "default" one. This configuration is very interesting in terms of simplicity and flexibility of the system deployment, because it does not entail any patch installation or kernel recompilation with respect a raw Linux installation. In addition, it is supposed to be the preferred configuration by the end user to execute standard working and entertainment software. However at the same time it is expected that other configurations can provide better ranging performance, because this default configuration is conceived to optimize the general performance of the system. This fact can be detrimental to the performance of the RTT measurement mechanism, because the more relaxed handling of the hardware interruptions by the OS and the possible lack of stability of the CPU clock signal. In order to try to palliate these issues as much as possible keeping at the same time the default Linux environment, what is done is making some few changes with simple commands just before performing the RTT measurements. These changes mainly consist of setting up a specific frequency for the CPU clock (with the command *sudo cpufreq-selector -g performance* and leaving enabled only one CPU in case the device has more than one (with *echo 0 > /sys/devices/system/cpu/cpu1/online*.

Real-time configuration Another considered option is a real-time OS, because it can optimize certain aspects of the system such making the interruption latency more deterministic, so that it could translate in a better ranging performance. The next Figure shows de architecture of a real-time OS. The idea such an OS is trying to minimize the work load of the system kernel, transferring as much load as possible to the user execution mode. This way only the more basic OS tasks, e.g. the interruptions handling, are executed in system mode.

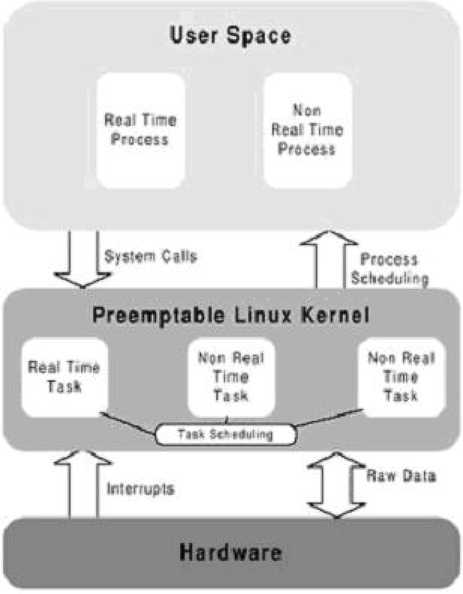

Figure 3.5: Architecture of a real-time OS

In our case the RT-Preempt path is employed [55], which makes the Linux kernel act as a Real-time OS. The patch and the additional library dependencies can be easily installed with the command *sudo apt-get install linux-rt*. In addition, as in the case of the basic configuration, a specific frequency for the CPU clock is set (with the command *sudo cpufreq-selector -g performance* and only one CPU is enabled in case the device has more than one (with *echo 0 > /sys/devices/system/cpu/cpu1/online*).

The patch is based on the principle of executing the interruptions handling in user mode instead of system mode. This is achieved converting the handlers in soft-irq, which become threads that are executed since the system boot, fact that allows modifying their priority and, the most important thing, their execution can be suspended to execute another process or thread with higher priority. This way a process conceived to work in real-time can actually be executed in real-time because it is suspended less often even though hardware interruptions occur. The patch is applied over the default Ubuntu distribution, without any extra modification. The ranging performance improvement with respect to the basic configuration that can be expected in practice with the real-time configuration is uncertain, because although it allows assigning more priority to the WLAN interruptions, the fact of converting the interruptions into threads could affect the behavior of the system.

Kernel without ACPI The last considered alternative is re-compiling the Linux kernel disabling the Advanced Configuration and Power Interface (ACPI) modules. ACPI defines platform-independent interfaces for hardware discovery, configuration, power management and monitoring. The ACPI specification is central to Operating System-directed configuration and Power Management (OSPM), a term used to describe a system implementing ACPI, which therefore removes device management responsibilities from legacy firmware interfaces. In other words ACPI allows the OS to have complete and exclusive control the power management of the hardware, as opposed to the previous BIOS central system, which relied on platform specific firmware to decide power management and configuration policy. ACPI defines several states for the global system, the performance, the processor and the computer devices, states that usually range from fully working to a close to powered off state.

It is expected that the ACPI of the OS makes increase the time variability and randomness of the RTT measurements (i.e. less deterministic measurements), because the changing commented states dynamically scale the power of the devices and the frequency of the processor. Thus, when hardware and power are not managed via ACPI the performance and working parameters of the processor and the other devices are expected to be constant, allowing higher stability of the time measurements. The main drawbacks of this configuration are the more complex system deployment due to the need of re-compiling and customization of the OS kernel, and on the other hand the negative impact of the ACPI disabling on the operation of the mobile device and their user applications. These drawbacks directly affect the feasibility of this implementation in practice, so that it can be only understood as a research proof of concept. The idea is, in the case of the hypothesis are confirmed and this configuration provides the best performance, comparing the RTT measuring performance of the commented configurations with (basic and real-time) and without ACPI (i.e. the current configuration) in order to analyze the measurement noise produced by the ACPI. This could allow mitigating this noise from the measurements performed with ACPI enabled.

The main steps to implement this configuration are:

1. Download the source code of the Linux kernel.
2. Create the kernel configuration file with the command make menuconfig or make xconfig, disabling the ACPI modules
3. Compile the code to create the image of the kernel and install it. When the ACPI is disabled, automatically only one CPU is enabled and in addition its frequency is set to the maximum, so that is not necessary to execute the corresponding specific commands as in the case of the basic and real-time configurations.

3.4 Performance assessment

The first part of this chapter includes two important studies:

- The assessment of the different possible configurations of the Linux OS explained in the last section.
- The assessment of the two commented clocks available to count the RTT, in order to take the major advantage -in terms of ranging performance- of both of them.

These two first issues are part of the design and implementation but are included in the current chapter because a great number of measurements with the lab prototype are performed to test the performance in each different case. In fact important results that give a fair idea of the performance of the technique are provided while presenting the results of both mentioned studies. After these two studies that test the different design alternatives, the design and implementation of the ranging technique, through the lab prototype, can be considered finished. Then in the second part of this chapter the finally achieved lab prototype is tested in different real environments and for different distances, analyzing the impact of the sources of error and trying to understand the behavior of the complete system.

3.4.1 Assessment of the OS configurations

The different considered OS configurations at the ranging lab prototype (basic, real-time and no ACPI) are comparatively evaluated by means of series of RTT measurements. The lab prototype (laptop with a 2 GHz CPU, an Atheros-based WLAN card and the Madwifi driver) is properly configured according to each configuration, the ranging software is installed, and then an important number of RTT measurements are carried out in different indoor and outdoor environments employing a Netgear D-Link AP. Series of 1000 RTT measurements are performed. Both nodes operate in IEEE 802.11b mode at 11Mbps.

Configuration without ACPI

The RTT empirical histogram Figure 3.6 shows the empirical histogram of a series of RTT measurements with this configuration. On the x axis the number of CPU clock cycles with intervals of 100 cycles is shown. The number of valid samples is 855, it is 85.5% of the 1000 theoretical samples. The obtained histogram seems to follow a distribution similar to the Gaussian. In order to corroborate it, one issue can be checked: the percentage of measurements that can be found in the interval $[\eta - 2\sigma, \eta + 2\sigma]$ is 855, which corresponds to a 97.07% of the total amount of samples, very close to the theoretical 95%.

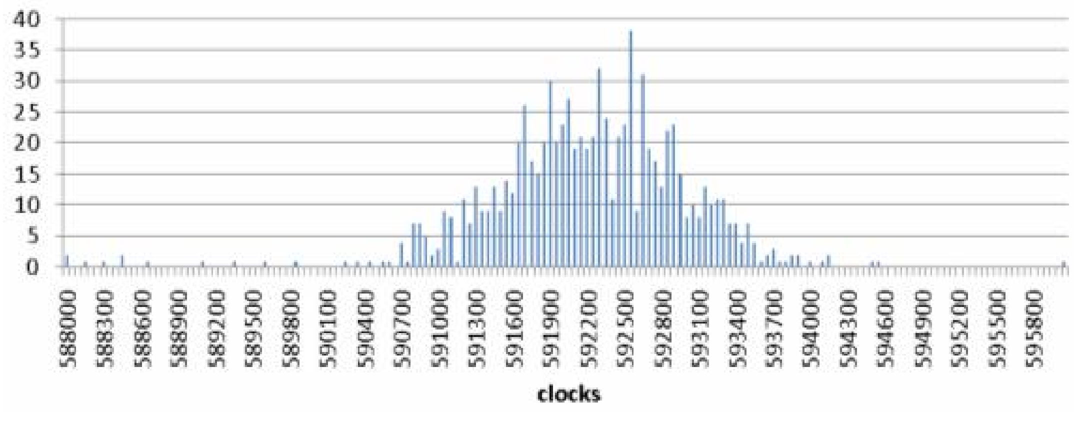

Figure 3.6: Histogram of a series of 1000 RTT measurements with configuration without ACPI

In Table 3.1 the main statistical parameters of the distribution depicted in Figure 3.6 can be found, expressed in CPU clock cycle units and in μs.

average (clocks)	592182.4269	Standard dev.(clocks)	813.867808
average (μs)	269.8290243	Standard dev. (μs)	0.37084038

Table 3.1: Main statistical parameters of a RTT distribution with configuration without ACPI

Equation 3.3 shows that although the standard deviation is really small with respect to the average value (Equation 3.2), fact that would lead to think that the time dispersion of the measurements is really low, the standard deviation is still noticeable in terms of distance.

$$\frac{std\_dev}{average} = \frac{813.867808}{592182.4269} = 0.1374. \tag{3.2}$$

$$distance\_of\_stdev = stdev(\mu s) \cdot c = 0.37084 \cdot 10^{-6} \cdot 3 \cdot 10^{8} = 111.25m. \tag{3.3}$$

At this point the first step of the processing of the collected RTT samples should be commented. This first step mainly consists of filtering the wrong measurements (spurious values); since the obtained histogram for this OS configuration is Gaussian it mainly corresponds to discard the values that do not belong to the main lobe of the distribution. This is achieved applying two windows: first the samples that are farther than 1/20 of the average are discarded in order to be focused on the candidate samples and then we filter again with a window of 3σ centered on the average.

Stability of the RTT measurements Once the distribution of a series of RTT samples and its dispersion have been analyzed, it is essential to assess the stability of the series of measurements. To this end, different series of RTT measurements have been carried out in the same environment and distance at different instants of time and then they have been compared. Figure 3.7 summarizes the obtained results about this issue (the x axis correspond to the different series).

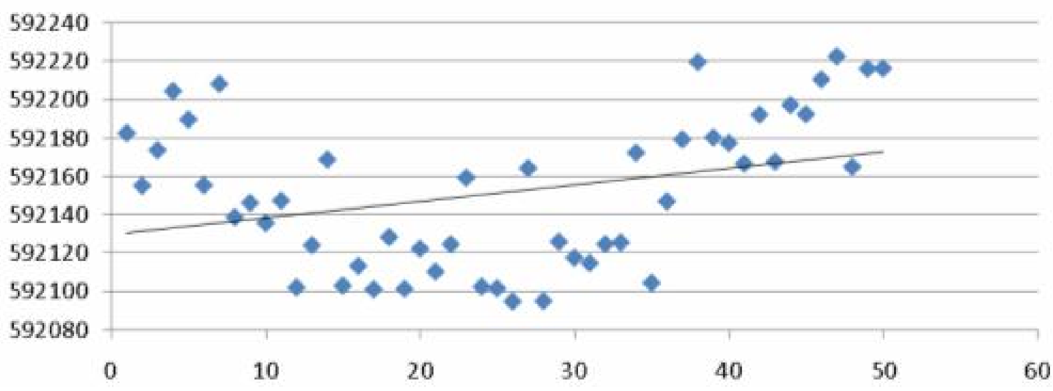

Figure 3.7: Stability of the series of RTT measurements with configuration without ACPI

The figure above shows the average values of 50 RTT series (as commented, see the x axis) in units of CPU clock cycles (the y axis). The mean value of the different 50 averages and the mean value of the 50 standard deviations are calculated and presented in Table 3.2. The standard deviation of the averages is about 800 clock cycles, which can be considered as a good result.

Mean average (clocks)	592151.700	Mean Standard dev.(clocks)	817.9968006
Mean average (μs)	269.8150236	Mean Standard dev. (μs)	0.372721764

Table 3.2: Mean values of the averages and standard deviations with configuration without ACPI

In order to have a clearer idea of the stability of the ranging method with this configuration, it is important to calculate the standard deviation of the 50 averages, in other words the time dispersion between the average RTT values of the performed series of measurements. This value is 38.869 CPU clock cycles, which translates to 5.19 m. in terms of distance and only 2.59 m. in terms of estimated distance with an RTT (two way) ranging method. This means that a ranging error of only 2.59 m. can be caused by the instability of the measurements. If we calculate the ratio between the deviation of the averages (the latter value) and the average value of the deviations (817.996 cycles), we have that the deviation of the averages is only 4.75% of the average of the deviations.

Basic configuration

The RTT empirical histogram Figure 3.8 shows the empirical histogram of a series of 1000 RTT measurements (841 of them are valid and therefore depicted on the histogram) with this configuration. On the x axis the number of CPU clock cycles with intervals of 175 cycles is shown. Although a main lobe can be distinguished on the histogram, an important number of samples grouped in small secondary lobes at noticeable bigger RTT values than the main lobe can now be seen.

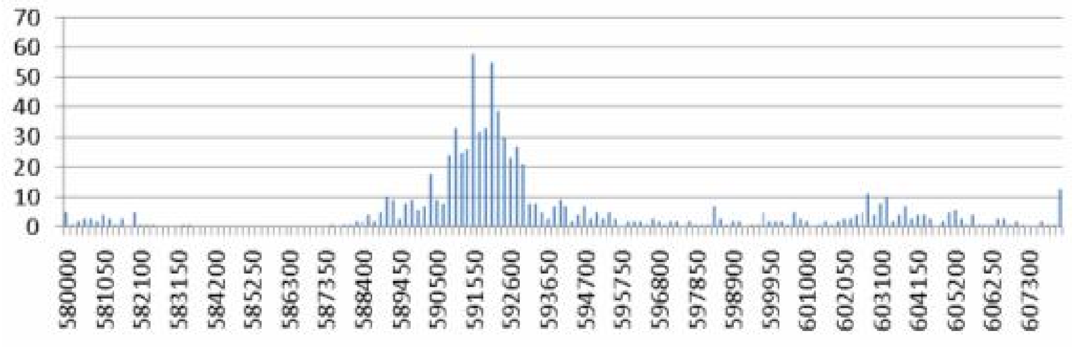

Figure 3.8: Histogram of a series of 1000 RTT measurements with basic configuration

This can be quantified taking a look at the average and standard deviation of the presented series. As expected, the average and especially the standard deviation are much bigger than with the configuration without ACPI. In fact, since most of samples that do not belong to the main lobe of the histogram (Figure 3.8) have bigger values than the ones on the main lobe, the dispersion of the measurements causes an offset on the average value.

average (clocks)	593759.9643	Standard dev.(clocks)	5716.672624
average (μs)	270.5478322	Standard dev. (μs)	2.604812515

Table 3.3: Main statistical parameters of a RTT distribution with basic configuration

The ratio between the standard deviation and the average (Equation 3.4) shows the bigger dispersion that is obtained with this basic configuration. Although this ratio is still below the 1%, Equation 3.5 calculates the standard deviation in terms of distance and evidences the unacceptable dispersion of the measurements for ranging.

$$\frac{std\_dev}{average} = \frac{5716.672624}{593759.9643} = 0.9627. \tag{3.4}$$

$$distance\_of\_stdev = stdev(\mu s) \cdot c = 2.6048125 \cdot 10^{-6} \cdot 3 \cdot 10^{8} = 781.44m. \tag{3.5}$$

In order to mitigate the commented positive offset (i.e. measurement noise) of the RTT average and to lessen the dispersion of the measurements a window is applied over the main lobe of the histogram (as can be seen in Figure 3.9) as first part of the processing of the samples, in a similar way than it was done in the configuration without ACPI. The big difference is that now a bigger number of wrong measurements are filtered, so that only the 74.67% of the 841 samples are considered valid.

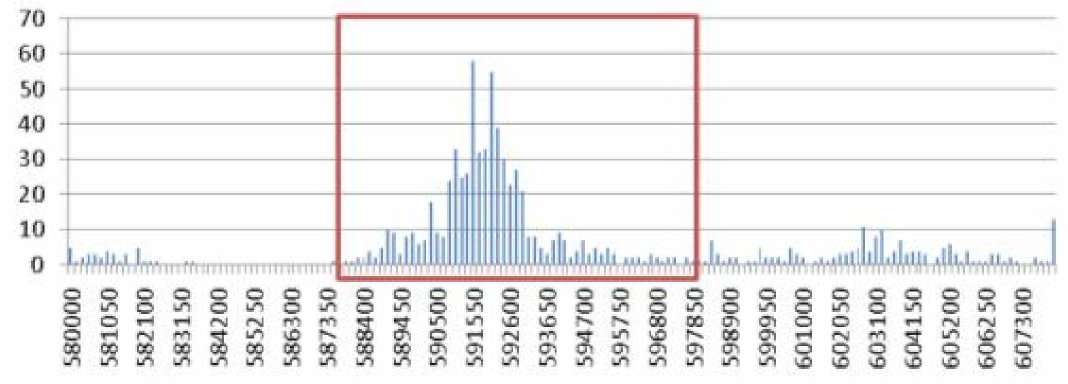

Figure 3.9: Window over the histogram of a series of 1000 RTT measurements with basic configuration

Next figure shows the histogram once the window has been applied. The secondary lobes, most of them to the right of the main lobe, have been eliminated. Then as expected the average and the standard deviation have been drastically reduced (see Table 3.4), in fact the average has became even slightly smaller than with the configuration without ACPI. Consequently, now the ratio between standard deviation and average is 0.2724%, closer to the one obtained with the configuration without ACPI. In addition, the average value corresponds to the RTT value at the centre of the main lobe.

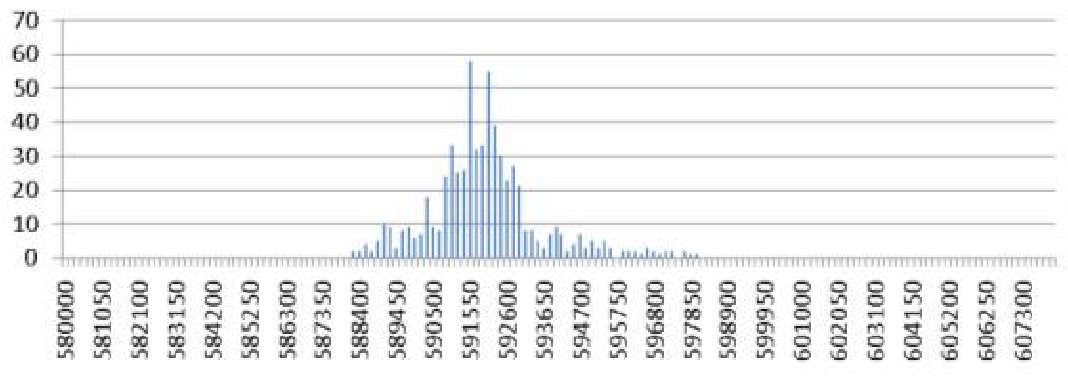

Figure 3.10: Histogram of a series of 1000 RTT measurements with basic configuration after the filtering

average (clocks)	591901.4697	Standard dev.(clocks)	1612.389677
average (μs)	269.7010057	Standard dev. (μs)	0.734688355

Table 3.4: Main statistical parameters of a RTT distribution with basic configuration

Stability of the RTT measurements In order to assess the stability of the series of measurements, again 50 series of RTT measurements have been carried out in the same environment and distance at different instants of time. Figure 3.11 summarizes the obtained average values for these series.

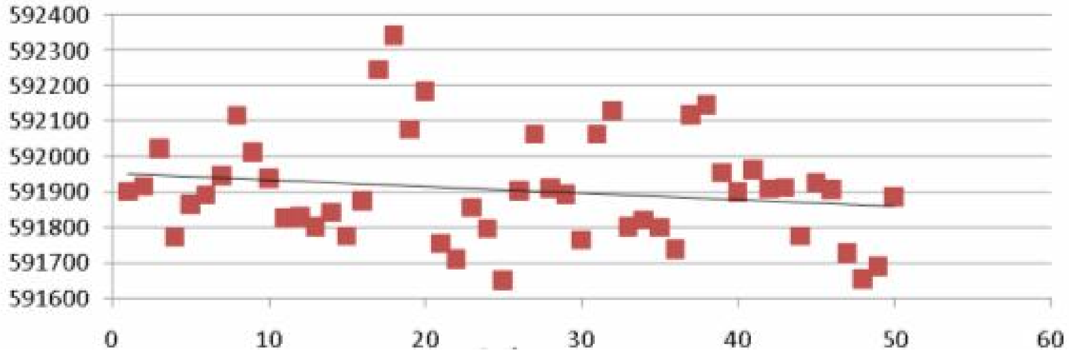

Figure 3.11: Stability of the series of RTT measurements with basic configuration after the filtering

Next table shows the mean of the averages of the 50 series and the mean of the standard deviations. Although the values are very similar to the ones presented in Table 3.2, fact that may lead to expect a good stability between series, the deviation of the averages of the series is 152.061105 CPU clock cycles. This value translates to 20.77 m. in terms of distance, which means a ranging error due to the instability of the measurement system of 10.38 m., an unacceptable value that corresponds to four times the error obtained with the configuration without ACPI.

Mean average (clocks)	591906.2221	Mean Standard dev.(clocks)	1660.455929
Mean average (μs)	269.7031711	Mean Standard dev. (μs)	0.756589833

Table 3.5: Mean values of the averages and standard deviations with basic configuration

Given this observed behaviour and these results, the situation is that we have an OS configuration that is very interesting in terms of simplicity and flexibility of the system deployment and in terms of coexistence with other end user software, but unfortunately it provides noticeably worse ranging performance than the OS configuration without ACPI. Then the objective is improving the performance achieved with the basic configuration. Since performed experiments have shown that the measurement noise of the basic configuration is mainly due to the ACPI operation, the approach consists of trying to mitigate this noise from the measurements. The idea to characterize this noise is performing a comparative analysis between the results obtained with the basic and the without ACPI configuration. Obtaining good results with this basic configuration would allow achieving the objectives of this research work about the software-based ranging technique.

Comparative analysis between the basic and without ACPI configurations to improve the basic configuration performance

Figure 3.12 allows an easy comparison between the RTT histograms obtained with both configurations, without applying any filter. It can be observed that the main lobes present a big overlap, so that it seems feasible to find a relation between both distributions. On the other hand, it can be seen the effect of the ACPI operation on the basic configuration: the height of the main lobe is shorter and a little bit wider, and an important number of measurements are spread to the right of the main lobe.

Figure 3.13 shows both histograms after applying the explained filters of RTT measurements for both configurations. It can be observed that the filtering for the basic configuration seems to be suitable, although the performed tests showed that the standard deviation of the measurements with

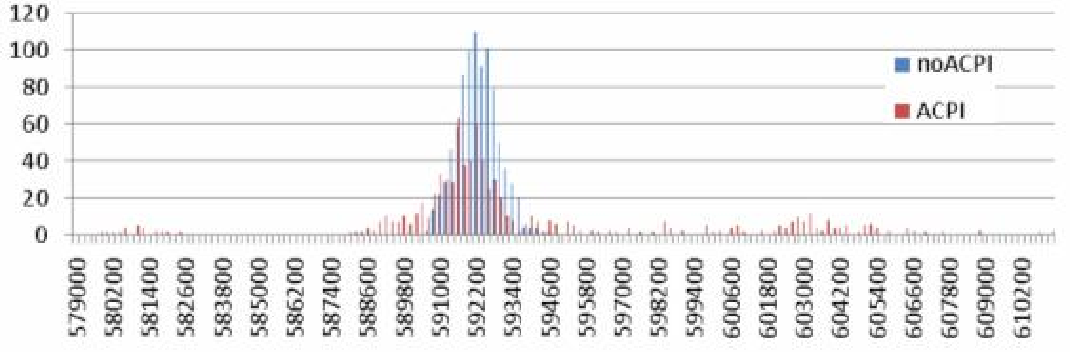

Figure 3.12: Comparison between RTT histograms with basic and No ACPI configurations

the basic configuration is two times the deviation reached with the configuration without ACPI. In addition, it seems that the central point of the basic configuration lobe is a little bit more to the left than the other one, fact that corroborates the lower RTT average values obtained in Section 3.4.1.

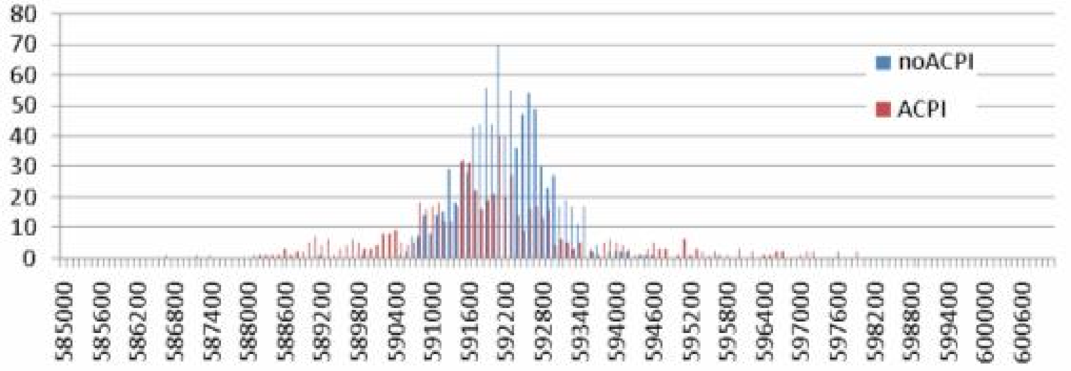

Figure 3.13: Comparison between RTT histograms with basic and No ACPI configurations after the filtering

The smaller RTT averages obtained in most cases with the basic configuration can be clearly observed in Figure 3.14, which depicts the average of the 50 series of RTT measurements carried out with both configurations. This figure also allows to observe the bigger unstability of the basic configuration, because the dispersion between the averages of the different series is noticeably bigger; in fact the before reported tests about stability showed an unstability four times bigger for the basic configuration. This bigger unstability makes that some (but few) of the averages with the basic configuration are bigger than with the wihout ACPI configuration.

Improving the stability of the basic configuration The first objective is to palliate as much as possible the instability of the RTT measurements obtained with the basic configuration. The approach consist on identifying some characteristic of the RTT histogram that allows detecting a series with the average noticeably bigger or smaller than the mean average (which corresponds to the black line that crosses the averages in Figure 3.14) and then rectifying the average RTT value. It has been found that this characteristic is the value of the standard deviation combined with the number of samples that are out of the main lobe. This is based on the idea that when the average is far from the mean average is because a bigger number of samples than usual are out of the main lobe (to the left of the

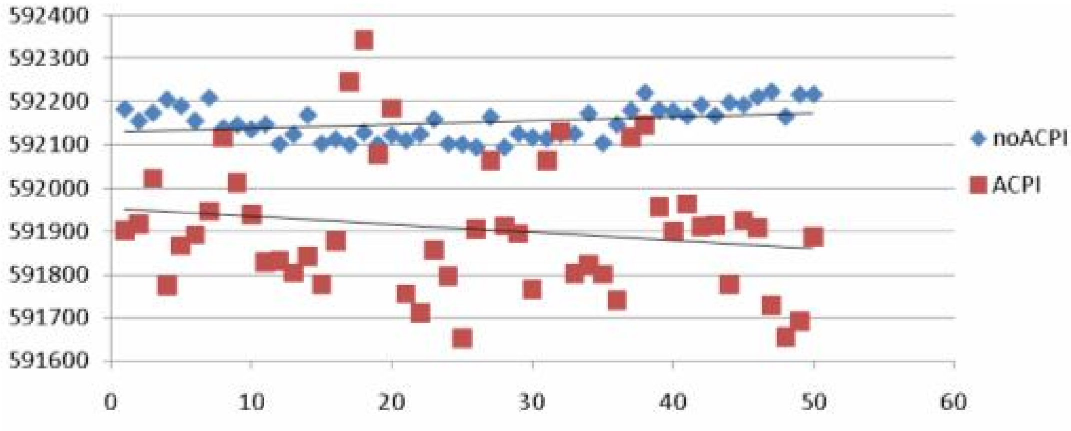

Figure 3.14: Comparison between the stability of the RTT measurements with basic and No ACPI configurations

main lobe if the average is smaller and to the right in the case of bigger average), fact that entails a bigger dispersion of the samples.

The way to proceed in practice is counting the samples that have smaller values than the average minus 3000 clock cycles ($\eta - 3000$, let's call it interval 1) and on the other hand counting the samples bigger than ($\eta + 5000$, let's call it interval 2) and in the interval [$\eta + 4000$,$\eta + 5000$], i.e. the interval 3. If more than 3% of the samples belong to the second interval, 1/10 of the standard deviation is subtracted from the average value. If it occurs for the third interval, 1/20 of the standard deviation is subtracted. To correct the lowest average values, 1/15 of the standard deviation is added to the average if the number of samples in the first interval is bigger than 3% or if the number of samples in the second interval is smaller than 0.5% and less than 1.5% in the other intervals. Next Figure 3.15 shows the achieved average values for the 50 RTT series obtained with the basic configuration before and after applying this correction approach. Numerically, the deviation between averages has been reduced in a factor bigger than two. The mentioned intervals and fractions of the standard deviation have been heuristically obtained; although they are not detailed in the text, an important number of candidate intervals and fractions have been tested and the ones that provide better accuracy have been selected.

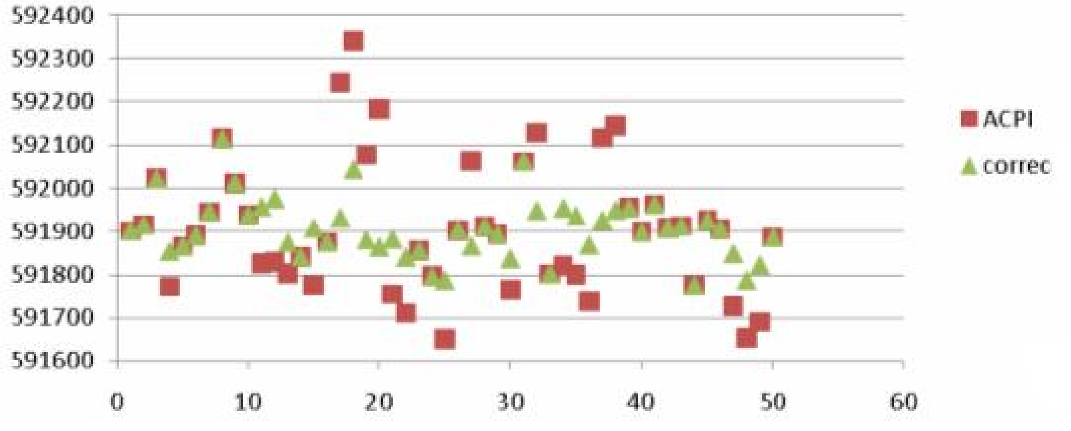

Figure 3.15: Stability improvement with the correction over the basic configuration

Next Figure 3.16 shows the averages of the 50 RTT series obtained with both configurations once the correction over the basic configuration series has been applied. It can be easily seen that now the stability in both configurations (basic and without ACPI) is similar, thus part of the RTT measurement noise due to the ACPI operation that affects the measurement with the basic configuration has been mitigated.

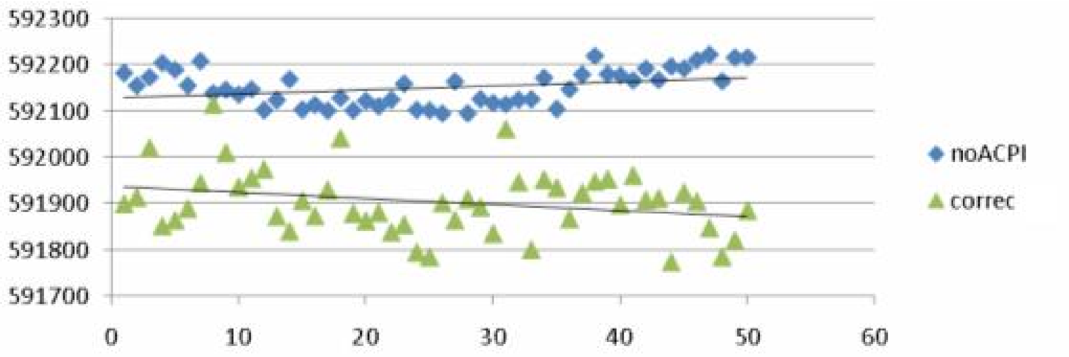

Figure 3.16: Comparison between the stability of the corrected basic configuration and the No ACPI configuration

Improving the ranging accuracy of the basic configuration The stability of the ranging performance when the basic configuration is employed has been improved. Now, taking as reference the RTT estimates obtained with the configuration without ACPI, the second objective is to mitigate the rest of the measurement noise due to the ACPI operation from the measurements taken with the basic configuration. This will allow enhancing the ranging accuracy achieved with the basic configuration. The idea is eliminating the offset that exists between the averages of the RTT series comparing the averages of both configurations. The mean RTT average considering all the series of measurements with the basic is 591904,427 CPU clock cycles, while with the configuration without ACPI it is 592151,700 (an offset of 247 cycles). The approach is very simple and consists on adding this offset to the obtained average RTT value. Thus the complete processing for the RTT samples that is proposed when employing the basic configuration consists of:

1. Filtering of the RTT samples using the above described window
2. Employ the average as RTT estimate
3. Applying the commented method to improve the scalability
4. Applying the commented method to improve the distance estimation accuracy

Real-time configuration

This is the last tested configuration. In our case an important problem occurs with the execution of the ranging software when trying to set up the complete real-time configuration explained in Section 3.3.3. The real-time OS processes manager assigns the thread corresponding to the WLAN card interruption handler to the second CPU of the laptop, which is disabled during the setup. Thus the time-stamps for the RTT measurements can not be captured. Then the tests have been performed without disabling the second CPU.

The obtained RTT measurements are slightly better than the ones obtained with the basic configuration without disabling the second CPU, but when the complete basic configuration is applied the real-time results are very similar, so that no advantage is taken from the real-time operation. Obviating the configuration problems explained above, it is important to try to understand why the real-time configuration does not provide an important enhancement with respect to the basic configuration. As explained when describing how this configuration works, the ranging performance improvement with respect to the basic configuration that could be expected in practice was uncertain, because although it allows assigning more priority to the WLAN interruptions, the fact of converting the interruptions into threads could affect the behavior of the system. According to the obtained results, this conversion does not positively affect the performance of the interruption handling, maybe because setting up higher priority is conceived for less priority process (which is not the case of interruptions).

The conclusion is that, given the same conditions, the real-time patch provides a slightly better performance but very similar to the basic one. However, since the optimum configuration of the laptop with real-time can not be reached and given the needed the extra real-time customization, this configuration is discarded to be applied to the ranging lab prototype.

3.4.2 Study of the available clocks to count the RTT

As mentioned before, two clocks are available to be used as time base to time-stamp MAC events from the WLAN card driver code: the clock of the CPU of the device and the clock of the WLAN card. The intention of this study is to perform a comparative evaluation between the performances of both clocks when measuring the RTT. This is a necessary study in order to reach the best performance as possible when measuring the RTT from the WLAN card driver code. A priori, two different approaches to measure the RTT could be derived from the results of this study: using one of the two available clocks because of its better performance in comparison to the other one, or using both clocks because it is observed that combining them properly provides better results than using whichever of them as standalone.

Previous discussion

Combining the measurements from both blocks can be feasible if it is found that measurements with one clock show desirable properties that the other clock does not have and vice-versa; then measurements taken with both clocks could be combined in a way that we take advantage of the desirable properties of both clocks while the worst issues of them are filtered. More specifically, an hypothesis for thinking about this possibility in our case is that it would be expected that the CPU clock provide better accuracy due to its bigger resolution and on the other hand that the WLAN card clock provide better stability because it is not affected for any mechanism of the OS, in contrast with the CPU clock. However, measurements with CPU and WLAN card clocks are both affected by some common issues -when accessed from the WLAN card driver code- that may lead to hypothesize a similar behavior between them:

a. Events at the physical and MAC layers on the WLAN card do not occur at any instant of time but at discretized instants that are governed by the WLAN card clock. Therefore it seems that it does exist an inherent discretization in time that affects the events to be time-stamped, regardless of the clock that is used to time-stamp those events. This fact could make useless in terms of accuracy using clocks with more resolution than the one at which events occur (it is the WLAN card clock), because at the end the bigger time-stamp clock resolution would not mean an accuracy improvement in the measurements.

b. A WLAN MAC event provokes a hardware interruption that is handled by the general hardware interruption handler routine of the OS and then by a routine in the device driver code once the type of interruption has been identified; finally the event can be time-stamped. The attention of the hardware interruption by the OS is supposed to be immediate because it has higher priority with respect to the other processes scheduled at the CPU, but anyway the instant of time when

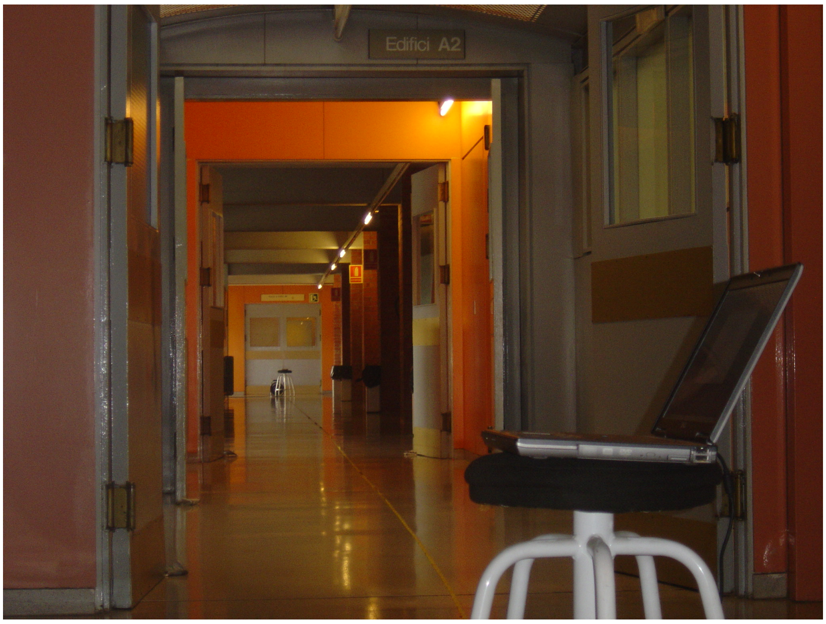

Figure 3.17: Indoor RTT measurements at UPC

it is handled is governed by the CPU clock. Hence the event time-stamps seem to be affected by this clock regardless of the clock that is employed as time base to time-stamp.

The study is aimed to allow checking the validity of all these raised conjectures, in order to know the most suitable use of both available clocks for measuring the RTT. The general experiment consists of an exhaustive comparative analysis of a great number of RTT measurements obtained at different distances employing both tested clocks of the lab prototype explained in the last paragraphs. The desirable properties and the weak points of each clock's performance are aimed to be identified, in order to fairly evaluate the suitability of each of them.

Analysis of RTT measurements

Measurements have been carried out with the lab prototype (laptop with a 2 GHz CPU, an Atheros-based WLAN card and the Madwifi driver) and a Netgear D-Link AP. The OS of the prototype is configured with the ACPI disabled, in order to minimize the impact of the OS on the CPU and WLAN clocks and therefore to allow a fair comparison of both clocks. Both nodes operate in IEEE 802.11b mode at 11Mbps. The measurements campaign has been carried out in indoor and outdoor environments, always in a LOS situation between both nodes, at distances from 0 to 70 m. Figure 3.17 shows the scenario of the indoor measurements. For each environment and distance, 10 series of 500 RTT measurements are collected. The transmission of the MAC data frames from the MT to the AP is induced sending ICMP Echo Request datagrams with minimum length. Each MAC event that defines the RTT is time-stamped from the Madwifi driver code with the CPU and the WLAN card clocks at once, so that the comparison between the performance offered by both clocks measurements is fair. Once the measurements campaign is finished, the RTT series obtained with both clocks are stored in a measurements database for their posterior analysis.

Quantitative analysis We take each pair of RTT measurements series corresponding to the same data-ACK frame exchanges measured with both clocks (one series for each clock). Then, for each one of the 500 RTT measurements, we obtain the ratio between the measured number of CPU clock cycles and the measured number of WLAN card clock cycles. This way we obtain 500 ratio values for each pair of CPU-WLAN clock series. Next Figure 3.18 show the obtained ratio for distances of 0 (first 5000 samples), 10 (from sample 5000 to 10000) and 20 m (the rest of them). The average ratio is 49.87557 at 10 m. and 49.87563 at 20 m., with a ratio standard deviation lower than 0.001 in both cases. This low standard deviation shows that the ratio remains almost constant for all the RTT measurements. Furthermore, considering all the distances between nodes it is observed that this ratio is almost constant regardless that distance. The observed behavior of the ratio suggests that the measurements obtained with both clocks are always very similar. On the other hand the obtained ratio value, close to 50, is logical because it corresponds to the scaling factor between the frequencies of both clocks.

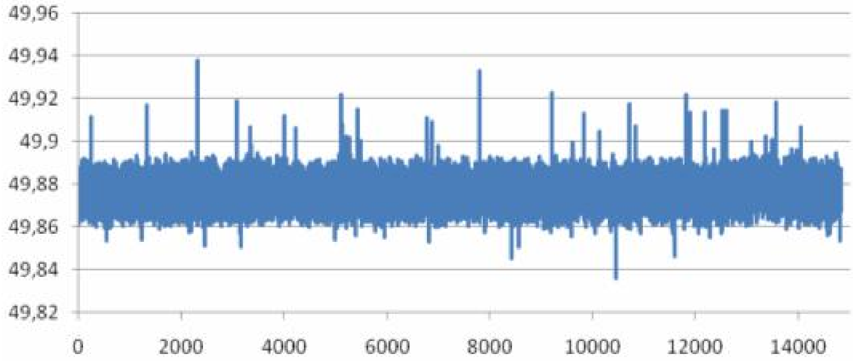

Figure 3.18: Ratio between the CPU and WLAN card clocks obtained from RTT measurements

In order to corroborate this first results observing the ratio, the pairs of RTT series with both clocks are compared measurement by measurement in order to detect differences between them in certain measurements. Each RTT measurement with the WLAN card clock is multiplied by the average ratio obtained for the series and the resulting value is compared with the corresponding RTT measurement obtained with the CPU clock. This comparison corroborates that the values of the RTT measurements with both clocks are almost the same, even when spurious RTT measurements occur. Figure 3.19 shows a pair of complete series of 500 RTTs each one measured with the corresponding clock (CPU or WLAN card); the spurious measurements can be clearly appreciated.

Since in Figure 3.19 it is difficult to distinguish between both series, Figure 3.20 depicts a zoom that allows appreciating the big similarity between them. In some pair of samples the accuracy is slightly better with one clock and in other pairs with the other clock, but the conclusion is that they are almost the same for the big majority of the samples.

Statistical analysis In order to better understand the statistics of the measurements obtained using each clock, the average, standard deviation and auto-correlation are obtained for each RTTs series at the tested different distances and environments. The obtained RTT averages with two compared series from both clocks do not present any noticeable difference: the ratio between both averages results almost the same value of the average ratio obtained before (e.g. for 10 m. the ratio between both averages is 49.87967 and the obtained average ratio was 49.87557). Regarding the auto-correlation, both series present the same degree of no correlation, as can be seen in the next Figure 3.21.

However, the comparative analysis of the standard deviation (after filtering the evident spurious measurements) allows appreciating a non-negligible difference. While the standard deviations for RTTs series measured with the CPU clock oscillate between 700 and 800 CPU clock cycles, when using the WLAN card clock they oscillate between 15 and 17 WLAN card clock cycles. Taking into account the scaling factor between both clocks (i.e. the obtained ratio, close to 50), it is observed that the measurements performed with the WLAN card clock contain bigger standard deviation (a difference of around 1 WLAN card clock cycle).

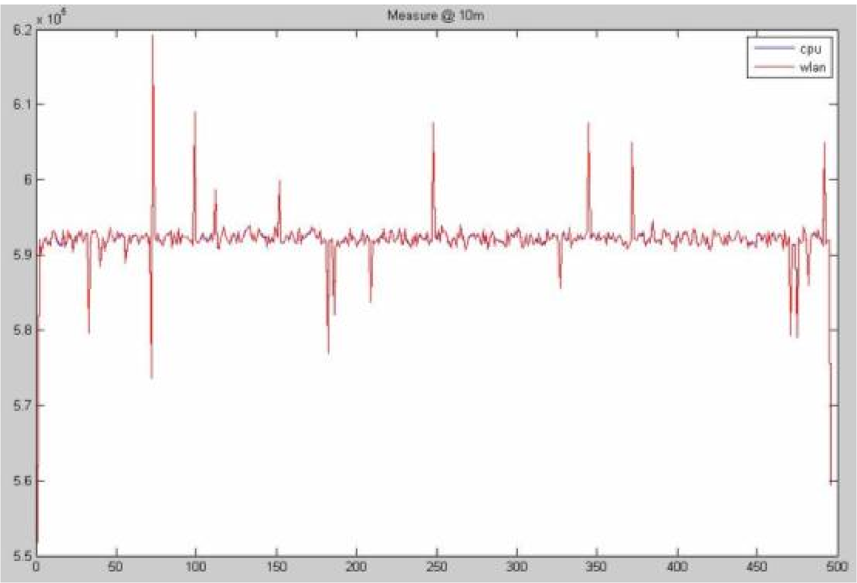

Figure 3.19: RTT series at 10 m. taken with the CPU and WLAN card clocks

From one hand this statistical analysis suggests that, since the average behaves very similar with both clocks, the expected maximum accuracy in both cases should not be so different. However, on the other hand the bigger standard deviation of the WLAN card clock measurements seems to indicate a major stability of the CPU clock when measuring the RTT. This fact does not seem so logical, as explained before, and therefore it has been studied in more detail to better understand this observed phenomenon.

Deviation of the measurements Beacon interval measurement

The objective is to understand the cause of the observed deviation of the RTT measurements obtained time-stamping the MAC events with both clocks, especially paying attention to the unexpected bigger deviation when using the WLAN card clock. When performing RTT measurements following our driver layer approach, the resulting deviation is mainly caused by several factors: the drift of the local WLAN card clock of the MT, the drift of the remote clock at the AP, the carrier sensing time before transmitting, the possible non-constant delay of the hardware interrupt handling by the OS of the MT, and of course the drift of the clock employed to time-stamp the MAC events (CPU or WLAN card clock). The idea consists of isolating as much as possible this latter mentioned factor (i.e. the one that is object of study), avoiding the effect of the other factors. The adopted way to try achieving this is measuring from the WLAN driver code in the MT the time between the transmission of consecutive beacons, which is supposed to be constant. This way the measurement of the drift of the remote clock at the AP and part of the carrier sensing time are avoided. In addition, considering that the hardware interrupt handling is supposed to be so deterministic due to its high priority, it seems to be a good experiment to measure de deviations of the CPU and WLAN card local clocks.

The WLAN card is configured now in AP mode with the following commands:

```
sudo wlanconfig ath0 destroy
sudo wlanconfig ath0 create wlandev wifi0 wlanmode ap
```

The beacon interval is set as deterministic to a certain period of time and the Madwifi driver code is properly modified to time-stamp the MAC events of beacon transmitted. Series of 500 time measurements between beacons are collected for both clocks and then the standard deviation for

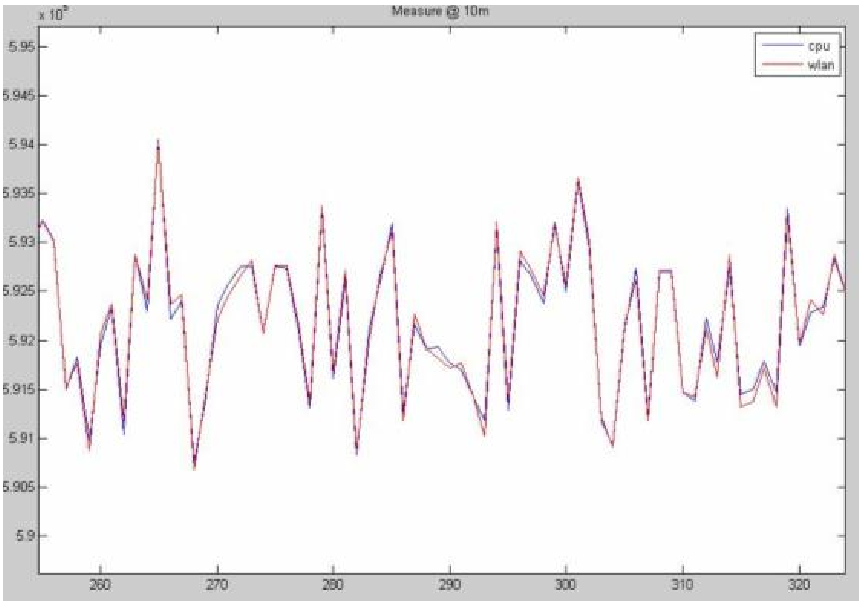

Figure 3.20: Zoom in on the RTT series at 10 m. taken with the CPU and WLAN card clocks

each series is calculated. The unexpected fact is that, at the contrary of it was expected, the standard deviations are much bigger than they were for the RTT measurements (around 50-60 WLAN clock cycles) , so that the experiment is not valid for our purpose. After studying the possible causes for that, the explanation is that the beacon interval can only be configured with a resolution of one time unit which equals to one microsec., time that means an inherent uncertainty for the beacon interval of 50 WLAN clock card cycles (much bigger than the deviations around 15 cycles obtained before for the RTTs). Despite not being successful, this experiment has been reported because we think the general approach is valid but unfortunately we have not found out the mechanism to properly implement it.

Call time measurement

A possible cause for the observed major deviation of the WLAN clock measurements with respect to the CPU clock ones has been studied. The studied issue is the time spent to perform the call that reads the clock cycles (for both clocks). This can not be considered a characteristic of the nature of the clocks, but has a direct impact on the time measurements in practice. In order to measure the call time, time-stamps are collected before and after the call is invoked. Obtained results show that the call to read the CPU clock cycles (i.e. the rdtsc() call) spends a deterministic amount of 66 CPU clock cycles. On the other hand, the call for the WLAN card clock ($wlan\_time = ath5k\_hw\_reg\_read(sc, AR5K\_PROFCNT\_CYCLE)$) oscillates between 5200 and 6200 CPU clocks, which means a maximum -> standard deviation of 100 CPU clocks (i.e. around 2 WLAN clocks). This can explain why the RTT measurements performed with the WLAN card clock contained bigger standard deviation (around 1 WLAN card clock cycle) as shown in the statistical analysis, so that it can be assumed that it was not caused by a major stability of the CPU clock.

Conclusions

First a quantitative comparative analysis of the collected RTT measurements employing both clocks as time base has been performed. It has been observed that the ratio between the measurements performed with both clocks is almost constant and that the time-stamped time value with both clocks is almost the same, even in the case of spurious RTT measurements. Then a statistical comparative analysis has been performed. The reachable RTT estimation accuracy with both clocks does not present noticeable difference because the average RTT values coincide. This corroborates the previous anal-

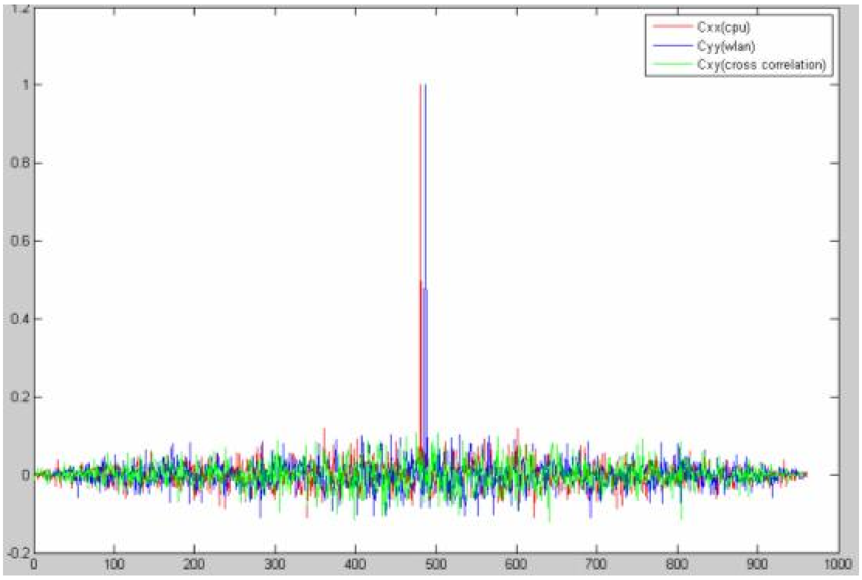

Figure 3.21: Correlation of the RTT measurements with the CPU and WLAN card clocks

ysis and the given hypothesis that questions the use of a clock at higher frequency than the WLAN card clock for achieving better accuracy: it seems that the accuracy is limited by the time resolution resulting from the clock that governs the IEEE 802.11 MAC operation (i.e. the WLAN card clock). Therefore it seems that it has no sense to count the RTT with a clock at higher frequency than the WLAN card clock.

On the other hand, it is detected that the measurements performed with the WLAN card clock present slightly bigger standard deviation than with the CPU clock. It has been seen than this is not really caused by the clock as itself but for the randomness of the time that takes time-stamping with the former clock. In any case, it is clear that in practice both clocks present very similar stability, which translates in a very similar deviation of the measurements taken with both different clocks. We think this can be due simply to an actual similar stability of both clocks or perhaps due to the given hypothesis that posed the importance of the hardware interruption handling schedule -common to the measurements with any of both clocks- even in the case of a WLAN card clock more stable than the CPU one.

The obtained results show as overall conclusion that the CPU and the WLAN card clock behave very similar when time-stamping events from the WLAN driver layer. It seems clear that they do not offer non-common desirable properties because in practice are both affected by decisive common issues; therefore a way of combining them to improve the RTT estimation performance can not be explored. Given its commented slightly smaller deviation, the CPU clock is more suitable than the WLAN card clock for estimating the RTT.

3.4.3 Ranging tests with the prototype

Once the different OS configurations and the available clock signals to count the RTT have been assessed, the main design and implementation decisions regarding the lab prototype can be considered closed. Now an extensive campaign of RTT measurements is carried out with the lab prototype that implements the proposed ranging technique, in order to totally define the ranging algorithm and to test then the ranging performance that is feasible employing the proposed TOA-based method. The prototype (laptop with a 2 GHz CPU, an Atheros-based WLAN card and the Madwifi driver) is employed with the configuration without ACPI and the CPU clock, in order to fairly assess the best achievable performance. A Netgear D-Link AP is employed in all the tests.

First the ranging algorithm has to be totally defined. As can be directly deduced from the explanations given in Section 3.3.1, the employed formula for estimating a distance i is:

$$d_i = c \cdot TOA_i = c \cdot \left(\frac{RTT_i - RTT_0}{2} \right) \left(\frac{1}{f_{clk}} \right), \tag{3.6}$$

where f_{clk} corresponds to the 2 GHz of the CPU clock. The pending issues in order to totally define the distance estimation algorithm using this formula are: testing that the RTT estimate at distance zero (RTT_0) is a really proper reference model (some preliminary tests showed anomalous behavior of the RTT measurements at this distance), and selecting the more accurate estimator for the RTT random variable.

Reference model

Ten series of 1000 RTT measurements are collected placing the prototype and the AP together, in order to test the behavior of the ranging model at distance zero. The average and standard deviation of the series are included in the following Table 3.6.

Series	Average(clocks)	Standard dev.(clocks)
1	592104.290	824.0584270
2	592031.688	849.1498403
3	592105.931	749.4174595
4	592148.453	684.2389132
5	592075.333	678.7083699
6	592116.273	762.0105546
7	592031.069	803.7004081
8	592083.627	664.6461750
9	591955.381	717.5423620
10	592103.328	723.5217641

Table 3.6: Average and standard deviation of RTT series at distance zero

Apparently the series seem to behave properly -low standard deviation as expected, similar averages- but analyzing the dispersion between the averages of the series (see the following Table 3.7) one can realize that this value is bigger than expected (38 clock cycles obtained in the prior stability study against the 55 obtained now). Figure X corroborates this. This unstable behavior can not be accepted for a reference model that will be used in all the distance estimates. In fact the higher instability at shorter distances between the prototype and the AP is a tendency that has been observed since the first system tests. Then the alternative design approach for the ranging algorithm is choosing a longer distance to obtain the reference model of the RTT measurements.

	Average(clocks)	Average(μsec)
Mean	592075.537	269.7803199
Standard dev.	745.699427	0.339779331
Deviation of means	55.6799369	0.025370667

Table 3.7: Dispersion between the averages of the series at distance zero

The considered distance is 10 m., because is long enough to avoid the undesired effect of shorter distances and at the same time is short enough to take the measurements easily. Next two Tables 3.8 and 3.9 summarize the characteristics of the RTT series collected at 10 m. The important thing is that

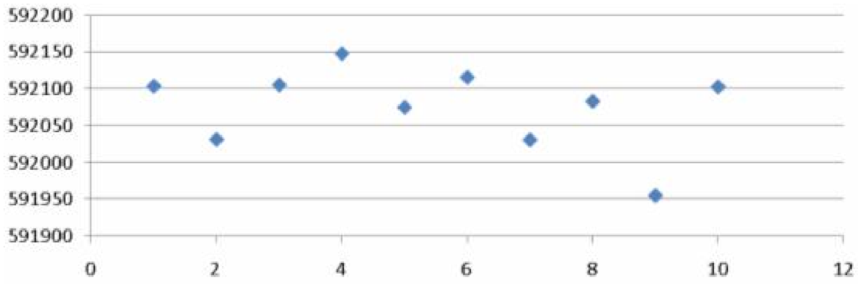

Figure 3.22: Mean RTT value for different series at distance zero

now the deviation between the averages of the series is only 18 CPU clock cycles. This good stability of the reference model will allow achieving better ranging accuracy when calculating whichever distance.

Series	Average(clocks)	Standard dev.(clocks)
1	592182.4269	813.8678083
2	592155.0358	904.1062737
3	592173.5832	836.7472858
4	592204.3105	773.6092369
5	592189.5154	796.8109877

Table 3.8: Average and standard deviation of RTT series at distance 10m.

	Average(clocks)	Average(μsec)
Mean	592180.9744	269.8283625
Standard dev.	825.0283185	0.375925688
Deviation of means	18.35123993	0.008361777

Table 3.9: Dispersion between the averages of the series at distance 10m.

Hence the adopted reference model for the RTT estimation is the RTT at 10 m. The expression for estimating the distance is then the following:

$$d_i = c \cdot TOA_i = c \cdot \left(\frac{RTT_i - RTT_{10}}{2} \right) \left(\frac{1}{f_{clk}} \right), \tag{3.7}$$

where RTT_{10} is the RTT estimate at distance 10m.

RTT estimator

Once the RTT samples have been collected and the explained filtering has been done, a proper estimator of the RTT RV has to be applied in order to obtain the RTT value. As has been observed, RTT measurements present important time variability and dispersion, due to some sources of noise already commented for the hardware-based ranging method, such as:

- Discrete time quantification to one CPU clock cycle
- Delays due to the WLAN hardware electronics in the MT and the AP

- Drift of the clock in the MT during the measurement and relative drift between the clocks in the MT and the AP

In addition to this factors, as expected another source of noise that drastically affects the RTT measurements exists, which is the already explained interrupt latency of the OS. Numerically, now with the software-based prototype we obtain RTT standard deviations around 800 CPU clock cycles (which means 0.4 μs. and 120 m. in terms of distance) while with the hardware-based ranging prototype we had less than 20 m. of deviation. This source of time randomness did not exist with the hardware-based technique and it is clear that it provokes the noticeably bigger dispersion of the RTT measurements carried out with the software-based lab prototype with respect to the ones performed with the hardware-based ranging prototype.

This new dominant source of measurement noise makes in addition that the observed effect with the hardware-based method, which consisted of the hidden noise produced by the indoor wireless radio channel (i.e. the environment), is now even stronger. In other words, in the software-based measurements the noise due to measurement system is noticeably stronger and therefore the possible noise due to the multi-path of the signal propagation is totally disguised and can not be noticed. In fact some RTT measurements carried out in the anechoic chamber at UPC with the software-based lab prototype totally corroborate this hypothesis.

The other important consequence of this new source of noise is the election of the RTT estimator that provides better accuracy. It seems clear that, as in the case of the other noise sources, this new source adds dispersion and positive bias to the measurements. Since when estimating the TOA and the distance (see Equation 3.7) two RTT estimates are subtracted (the reference RTT at 10 m. is subtracted from the RTT at the actual distance), one can think that this bias is mitigated with this subtraction. In the case of the hardware-based technique this bias was not totally erased with the subtraction, and it had to be employed the average to estimate the reference RTT but a "smaller" estimator than the average for RTTi, so that the bias could be totally mitigated. In the case of the software-based system, performed tests have shown that the best choice is employing the average for both the actual and the reference RTT estimates of the ranging formula, so that the bias seems to be totally erased with the RTT subtraction. The explanation is that now the previous filtering is more aggressive and makes possible reducing the positive bias of the RTT measurements.

Ranging accuracy

A new extensive set of distance estimates employing again the lab prototype with the proposed ranging method is performed in order to assess the achievable ranging accuracy. The tested distances are between 0 and 70 m in a stringent indoor environment at the UPC. Both the lab prototype and the AP are placed 1.5 m. above the ground in order to preserve the Fresnel. For each distance, several series of 1000 RTTs are collected. Next Figure 3.23 show the RTT average value of each series at each tested distance (x-axis), in units of CPU clock cycles (y-axis).

It can be observed a linear behavior of the RTT estimates with the distance, a desirable characteristic for a ranging system. Analyzing the results quantitatively, the stability of the measurements (i.e. the deviation between RTT averages at a given distance) is very similar to the ones obtained in the previously reported system design tests, ranging from the 55 clock cycles at zero m. to only 30 clocks for distances longer than 40 m. In addition the standard deviation (i.e. the dispersion) of the RTT samples of a series follows the obtained model in previous reported campaigns.

These obtained RTT series are employed to estimate the corresponding distances applying the already described ranging complete algorithm for the OS configuration without ACPI, i.e. the explained filtering and processing of the RTT samples, the RTT estimation and the distance calculation using the previous Equation 3.7. As explained, the employed reference RTT is RTT_10. Next Table 3.10 summarizes the obtained ranging results. The mean error considering all the distances is 1.802 m., which is a really encouraging result.

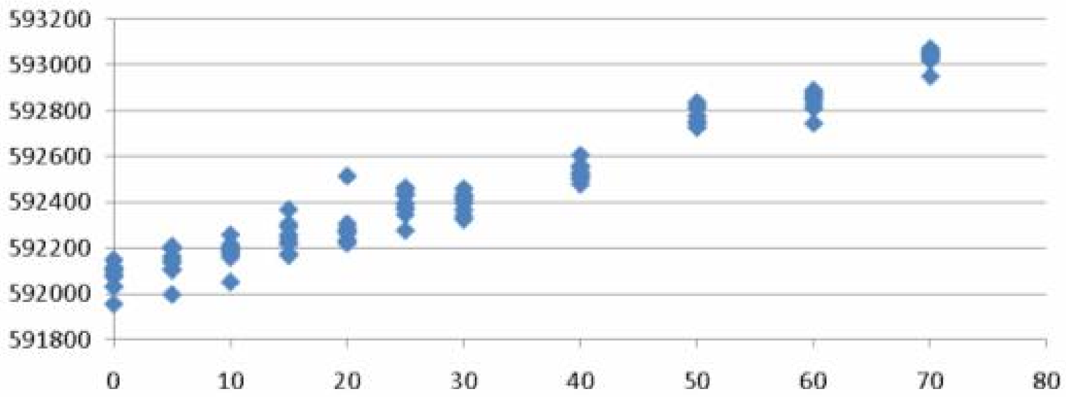

Figure 3.23: Behavior of the mean RTT values with distance

Distance (m)	Distance estimate (m)	Abs. error (m)
5	7.13391736	2.13391736
10	10	0
15	14.4557461	0.54425392
20	18.3450813	1.65491865
25	24.8154128	0.18458722
30	26.4623937	3.5376063
40	33.7307694	6.26923063
50	50.3842977	0.38429771
60	56.7947281	3.20527192
70	70.1077174	0.10771744

Table 3.10: Ranging results for distances between 0 and 70m.

3.5 Conclusion

This chapter has reported my research about TOA-based ranging using WLAN, with the special objective of totally avoiding any hardware manipulation of the WLAN devices. Previous to this research, really good results were obtained with the attempt to achieve a TOA-based ranging method with WLAN slightly manipulating the WLAN client's interface (see Chapter 2). Since this needed hardware manipulation constitutes a limitation of the technique in terms of complexity and cost of the implementation and flexibility in the deployment of the solution, the main objective of the reported work in the current chapter has been achieving a good ranging performance as close as possible to the one achieved with the hardware-based method but by means of a pure-software solution over a standard WLAN infrastructure.

The more basic design principles of the researched software ranging technique are the same as in the hardware-based technique, which can be summarized as employing RTT measurements using standard data/control frames at the IEEE 802.11 MAC layer to estimate the TOA between transmitter (the MT) and receiver (the AP). Given this starting point, the real challenge consists of obtaining an accurate estimate of the RTT by means of a pure software mechanism on the MT. Considering some existing experiences of other researchers pursuing the same objective, it was decided to explore and innovative approach that mainly consists of capturing the time-stamps of the needed MAC events at OS level (instead of at hardware level) and measuring the RTT with the CPU clock signal. To this end the driver of the WLAN interface has to be properly modified and the OS configured to provide the better performance as possible. This latter issue can be considered the key point of the research, because the RTT measurements are directly affected by the WLAN interrupt latency at the OS and the performance of the CPU clock. Different OS configurations have been tested (basic, no ACPI and real-time) by means of a developed lab prototype that implements the ranging method. The result was the configuration without ACPI allows achieving noticeably better results than the other configurations. However, since the basic configuration has the important advantages of more feasibility and simplicity in the deployment, an approach to reduce the noise of the ranging results with this configuration has been presented.

Extensive campaigns of RTT measurements performed with the lab prototype have demonstrated that, as expected, the time dispersion and inaccuracy of the RTT measurements is noticeably bigger with respect to the hardware-based method, mainly due to the interrupt latency and the impact of the OS mechanisms on the measurements. For that reason the adopted processing of the RTT samples to estimate a single RTT is now more stringent and the number of needed samples is bigger. Distances estimates indoors performed with the complete implementation of the proposed method on the lab prototype show average errors smaller than 2 meters and a good stability of the system. This is a good result, because it allows demonstrating that is feasible to achieve accurate ranging by means of a pure-software solution over WLAN standard equipment. In my opinion, this constitutes an important step forward with respect to the current indoor WLAN location state of the art.

Chapter 4

The obstructed path problem

4.1 Introduction and objective

During the research on the TOA-based ranging techniques over WLAN (reported in the last Chapters 2 and 3) it has been shown that the multi-path propagation that the radio signal suffers does not affect the ranging performance when MT and AP are in LOS situations. However, since the ranging technique is intended to operate indoors, it is expected that physical obstructions between the MT and the AP (i.e. NLOS situations [56, 57]) exist often in some office and industrial environments. In NLOS the hypothesis is that the obstruction can degrade the performance of the TOA-based ranging method. This hypothesis is based on the fact that the TOA needed to estimate the distance is the one belonging to the signal arriving through the Direct Line Of Sight (DLOS) [58]; thus TOA estimation could be inaccurate if the DLOS is blocked because then it has to be obtained from signals following other -longer- paths. In other words, it can be expected that the obstruction of the direct path can cause important ranging accuracy degradation when the obstruction is severe. Therefore it is essential to mitigate the possible negative impact of this phenomenon, so that the ranging technique is totally robust to this obstructed path problem.

This problem poses a challenge in TOA and distance estimation and for that reason I have done an important research work on that topic. This chapter reports it. The first experiment that has been done is evaluating and quantifying the negative impact of the obstructed path phenomenon on the researched hardware-based ranging technique. Once this is done, in the case accuracy degradation is observed, an approach to mitigate this negative effect is then researched. The method to make the ranging technique robust to the obstructed path problem must not represent degrading other aspects of the ranging technique; hence it must work in real time, requiring low computational cost and without entailing any hardware modification. The most relevant research works related with the different specific experiments and methods proposed in this chapter are commented in the proper parts of the text.

4.2 Impact of the obstructed path phenomenon on the hardware-based ranging

This section discusses the impact of the obstructed path phenomenon on the accuracy of the ranging technique, as a first and mandatory step in achieving a robust solution to this phenomenon. It is expected that multi-path distorts TOA estimations differently depending on the kind of obstruction (material) between the MT and the AP, i.e. depending on the channel profile, because this obstruction is expected to directly affect the indoor propagation of the signal.

4.2.1 Related work

Over the last two decades, most indoor propagation studies were performed for communication applications ([47, 59]). Channel profiles were often categorized as LOS or NLOS. In 1998, a new framework to model the radio propagation characteristics for ranging and geo-location applications was provided ([58, 60]), in which multi-path characteristics are divided into three classes according to the availability and strength of the direct path with respect to other paths: Dominant Direct Path (DDP), Non Dominant Direct Path (NDDP) and Undetectable Direct Path (UDP). In these works, the performance of traditional GPS and RAKE-type geo-location receivers that detect the TOA is evaluated. In [56, 61] and [62], the performance of different TOA algorithms (inverse Fourier transform (IFT), Direct Sequence Spread Spectrum (DSSS) and super-resolution Eigenvector (EV) algorithm) in DDP, NDDP and UDP conditions is compared. This latter classification (DDP/NDDP/UDP) seems to be more suitable for our purposes than the LOS/NLOS because it considers the degree of obstruction rather than using a too simplistic binary classification. Since the TOA arriving through the DLOS path is needed to estimate the distance, it seems reasonable to assume that degradation of the ranging accuracy will increase with the degree of obstruction (DDP/NDDP/UDP) that suffers the direct path.

4.2.2 Theoretical analysis

The goal is to evaluate the performance of the TOA (and distance) estimation algorithm in different types of channel profiles, which may differ depending on the percentage of signal arriving through the direct path. Entering more into detail, the impact of an obstructed path on the TOA estimation is expected to be larger when the path is obstructed by an obstacle that attenuates the signal in such a way that it reaches the AP with less power than its sensitivity threshold (UDP scenario). When this is the case, in the exchange of the frames for RTT measurements it is expected that either the signal picked up by the physical layer at the AP corresponds to the signal reaching the AP through the reflected or diffracted paths that are longer than the DLOS (resulting in a positive bias in the TOA estimate), or the total amount of signal is below the AP's sensitivity threshold (i.e., the frames are lost). In cases where an obstacle introduces less attenuation (e.g., narrow walls), the signal that reaches the AP through the direct path is likely to be weaker than if it follows reflected or diffracted paths, but it still may be stronger than the AP sensitivity, so that the DLOS path is not fully obstructed (NDDP scenario). It is expected that some of the signal corresponding to a frame will be picked up at the physical layer of the AP through the direct path while the other portion will arrive through reflection or diffraction. The percentage of the frames signal that arrive through reflection or diffraction is expected to be lower than in the UDP, so the estimated TOA may be affected with a smaller bias. Finally, in a situation of direct visibility (DDP scenario) between the MT and the AP, almost all the signal is likely to arrive through the DLOS path, so that the error introduced by multi-path should be noticeably lower than in the two previous cases.

Although both UDP and NDDP are considered NLOS situations, the impact on the RTT empirical distribution and TOA estimation is expected to be different. It should also be noticed that the DDP case can occur even in a NLOS situation, if the obstructing obstacle introduces sufficiently low attenuation that the DLOS path remains the strongest one. Thus, the DDP/NDDP/UDP [58] classification is more adequate for the study of the impact of multi-path on our ranging system than the LOS/NLOS, as it allows for these distinctions to be made. In this first stage, the impact of the obstructed path phenomenon on the TOA estimation accuracy will be evaluated in all possible multi-path conditions: DDP, NDDP and UDP.

4.2.3 Experiment to assess the impact

Given the explanations of the theoretical analysis, the idea is performing a comparative evaluation of the performance of the ranging method in DDP, NDDP and UDP conditions. To this end, a complete RTT measurements database is built storing an adequate number of sets of RTT samples collected in the three different types of channel profiles, at several known distances, using the developed hardware-based ranging lab prototype and a LinkSys Wireless-G 2.4 Ghz AP, both operating in IEEE 802.11b mode. In order to guarantee that each set of RTT measurements is carried out in a scenario corresponding to the desired channel profile, the Channel Impulse Response (CIR) is previously obtained through a network analyzer and checked using a similar technique to the one explained in [63]. The measurements are performed inside the building of the Telematic Engineering Department at UPC in Barcelona, an office building with indoor walls and metallic objects, including an elevator. The locations for transmitter and receiver for NDDP and UDP situations are depicted in Figure 4.1. As can be seen, UDP cases are found in some situations in which the elevator obstructed the DLOS path, due to its metallic surfaces; NDDP cases occur in some situations in which walls obstructed the DLOS path.

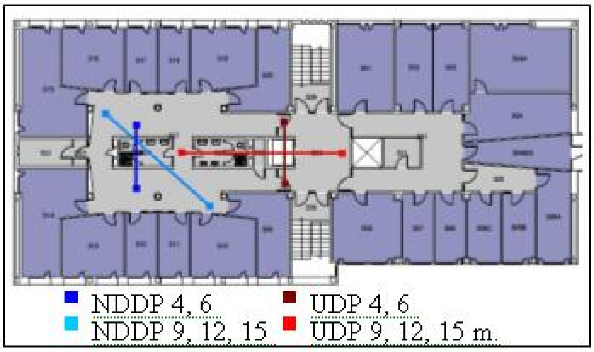

Figure 4.1: Measurement campaign: NDDP and UDP

To classify the CIR obtained in one channel profile, the following procedure is followed: the ideal TOA between the transmitter and receiver is calculated from the actual distance between them, and then the peak of the CIR that corresponded to that time mark (the DLOS path peak) is identified. Afterwards, by observing the power strength of this peak and applying the criteria of the definition of DDP, NDDP and UDP [58], the classification can be made. The sensitivity of the AP used in the latter RTT measurements is -88dBm. Hence, if the CIR shows that the power strength of the peak corresponding to the DLOS path is not the strongest, but it is higher (in fact, significantly higher, to ensure the reliability of the decision) than this threshold, the scenario would be classified as NDDP. Figure 4.2 shows the obtained CIRs at 6m. In this figure for each CIR a vertical line at 22 nsec. (x-axis) indicates the TOA of the DLOS path between transmitter and receiver.

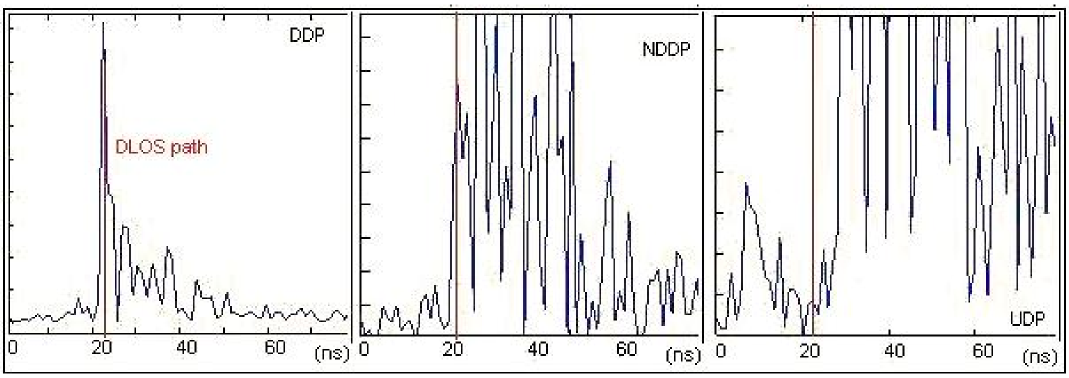

Figure 4.2: CIR at 6m for DDP, NDDP and UDP

Once the RTT measurements are completed, distances are estimated by employing the distance estimation algorithm of the presented hardware-based ranging method (Chapter 2 for LOS situations. Although in Chapter 2 complete details can be found, this ranging algorithm can be summarized with the following equation:

$$d = c \cdot TOA = c \cdot \left(\frac{(\eta_i - \frac{\sigma_i}{3}) - \eta_0}{2} \right) \cdot \left(\frac{1}{f_{clk}} \right), \tag{4.1}$$

where d is the resulting distance estimate between the MT and the AP, $\eta_i - \frac{\sigma_i}{3}$ is the estimator for the RTT at this distance d, η_0 is the estimator for the reference RTT at distance zero and f_{clk} is the frequency of the WLAN card clock (i.e. 44 MHz) employed as time-base for measuring the RTT.

Then, the impact of multi-path is assessed by comparing the obtained values of the distance estimates at each specific real distance between the different multi-path channel profiles. Table 1 summarizes the results obtained, showing how the distance estimation figures increase with distance (as needed) and with the degree of obstruction of the direct path (expected though undesired). Thus, as expected, the strongest impact of multi-path on RTT (and distance) estimation occurs in UDP situations and the distance estimation errors in such conditions are noticeably larger than in DDP situations. On the other hand, results show that, in NDDP cases, the performance of the ranging system does not vary substantially with respect to DDP. The slightly longer estimated distances for NDDP than for DDP corroborate that, in an NDDP situation, the first signal to reach the receiver followed a reflected or diffracted path in a certain percentage of the RTT measurements performed for the TOA estimation. Notice how in Table 4.1 the average absolute error for UDP (2.1958 meters) is larger than the error for both DDP and NDDP (around 0.9 meters). These differences agree with the conjectures presented in the theoretical analysis of this section.

Distance (m.)	DDP	NDDP	UDP
4	3.2628	4.4770	6.1446
6	5.3502	6.7757	9.6546
9	8.1964	10.6855	12.8131
12	11.2372	11.7029	13.1334
15	13.5932	13.7552	14.7664
Average absolute error	0.8719	0.8960	2.1958

Table 4.1: Distance estimations for different multi-path situations using the ranging formula for LOS

In order to better understand the impact of multi-path when the path is obstructed, the normalized histogram of the distance estimation at an actual distance of 9 m is displayed for each multi-path condition in Figure 4.3. This is performed by taking 1000 samples of the RTT series in groups of 50 and estimating the distance for each group using Equation (8). Figure 4.3 shows that the accuracy degradation in a UDP situation is significant: most of the estimates obtained range from 12 to 14 m. Consequently, the distance estimation algorithm proposed in Chapter 2 (i.e., for LOS) is no longer valid for UDP situations. These results suggest the need for an alternative method to mitigate the accuracy degradation from which the ranging technique suffers when an obstacle obstructs the DLOS path between the MT and AP, especially in UDP situations. The proposed method is detailed in the next section.

4.3 Method to mitigate the obstructed path problem

Since the impact of the obstructed path phenomenon is different depending on the radio channel profile, our proposed method works in two steps. First, the channel profile is identified in order to discriminate the multi-path condition (i.e., the channel profile) between MT and AP, with emphasis on the UDP case. Next, ranging sensitive to the identified multi-path condition is achieved by employing

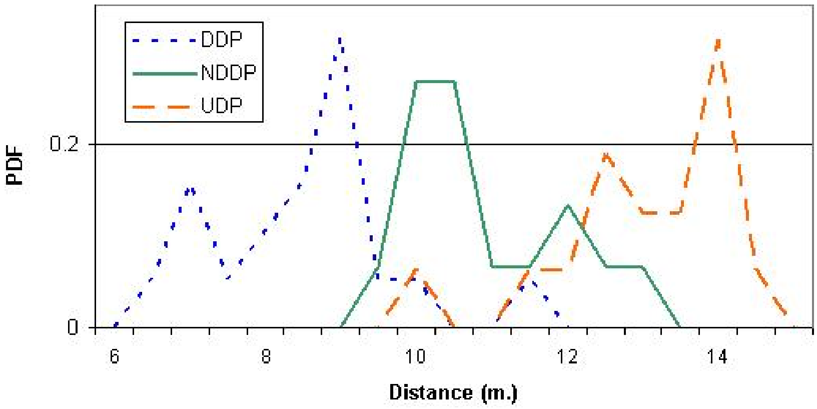

Figure 4.3: PDFs of distance estimation at 9 m for DDP, NDDP and UDP employing the LOS algorithm

specific distance estimation algorithms that depend on the identified profile, in order to obtain the maximum accuracy regardless of the multi-path condition existing between the MT and AP.

4.3.1 Channel profile identification

The identification of the radio channel profile between the MT and AP has to be achieved from the RTT measurements at the IEEE 802.11 MAC layer that are carried out as part of the process to estimate the distance (see Chapter 2 for more details about this ranging process), since in normal use it is not possible to obtain the CIR of the radio channel. This can be seen as establishing a correspondence between the RTT empirical histogram and the CIR of the existing radio channel. The key idea is to find a statistical parameter of the obtained RTTs that varies with the profile (i.e., the multi-path situation) in order to allow the pursued profile identification.

Related work

This idea has been inspired by some past publications, in which researchers have tried to establish whether a measured range corresponds to an NLOS or an LOS situation. In [64], a non-parametric method comparing the probability density function (PDF) is proposed, but it involves a high level of complexity and is not tested using real ranging measurements. In [65], the standard deviation of time history of the range measurements is used. These theoretical proposals focus on ranging measurements performed in cellular networks like GSM, without considering indoor scenarios affected by severe multi-path conditions. In [66], the authors propose a method for mitigating the effect of multi-path on TOA estimation when an LOS situation exists between an MT and a GSM base station. Other approaches regarding the NLOS error mitigation in location systems are based on tracking [67]; essentially, this approach proposes including the TOA bias introduced by NLOS as a new state in the Kalman algorithm applied to UMTS networks and not dealing with the establishment of a correspondence between the ranging measurements and the multi-path case. Regarding the reported attempts to deal with the obstructed path problem, the most relevant are the following: using frequency diversity to orthogonalize multi-path with respect to direct path [36], and implementing a multi-path decomposition block that uses a Maximum Likelihood algorithm to calculate the delay parameters [37]. In both cases the approaches are applied to WLAN ranging methods that work at physical level, entailing important dedicated hardware modules to estimate the distance, and then the approaches take advantage of this fact. In our case the idea is totally different, because the pursued method to deal with the obstructed path problem must be less complex and purely based on statistical processing software. In addition, as far as we know our current contribution is novel since it contains the first attempt to discriminate the type of indoor radio channel between DDP, NDDP and UDP, employing TOA or RTT measurements taken at the MAC layer.

The proposed method

As stated before, the basic idea to discriminate the channel profile from the collected RTT samples is to find a statistical parameter of the obtained RTTs that varies with the profile. In order to test the feasibility of this approach, key statistical parameters that would allow the discrimination (i.e., a complete statistical characterization including average, standard deviation, Coefficient of Variation (CV), coefficient of skewness, and autocorrelation) of all the RTT series of the database mentioned in Section 4.2.3 are determined. Then, an exhaustive comparison between RTT statistical parameters corresponding to the same actual distance for different channel profiles is performed. Focusing on the RTT standard deviation, Figure 4.4 shows that, regardless of the actual distance, a clear trend exists that links the standard deviation with the multi-path situation: the greatest values occur in UDP, the lowest occur in DDP, and the values between them correspond to NDDP. As expected, the standard deviation increases with distance. Only as a clarification, notice that in Figure 4.4 the RTT standard deviation values are given in 44 MHz clock cycles. Regarding the CV, a trend that depends on the multi-path situation can also be observed, but the differentiation seems to be more difficult, especially at a separation of 15 meters. Other statistical parameters such as the coefficient of skewness and autocorrelation are also analyzed. The former describes asymmetry in the histogram, and it is not possible to relate it to the different channel profiles. In the latter case, all the RTT series present no correlation regardless of the profile and the distance.

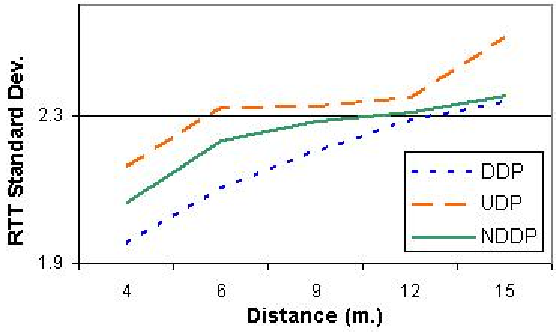

Figure 4.4: RTT standard deviation for DDP, NDDP and UDP

Following this analysis it can be concluded that, in order to discriminate the multi-path situation, both the average and standard deviation of the RTT are needed. Using these two parameters, it is possible to decide in which region of Figure 4.4 the measurements lay, and hence the channel profile can be obtained. Which is the optimum decision method to discriminate the region is a system design issue out of the scope of this research work. On the other hand the required previous knowledge of the curves depicted in Figure 4.4 is not expected to entail any system pre-calibration, because the obtained curves are expected to be valid for deployments in similar environments. Since the channel profile is likely to be used to refine the estimation of the distance from the RTT samples, a practical system should include the following steps:

1. estimate of the RTT average and standard deviation
2. decision about the channel profile
3. refinement of the distance estimation, accordingly.

This sequence could be iterated.

4.3.2 Distance estimation sensitive to a multi-path situation

Since it has been shown that UDP cases can be clearly discriminated from DDP, and UDP exhibits unacceptable ranging errors, our objective is to define a distance estimation algorithm to provide good accuracy for UDP situations. This idea follows the same approach that the algorithm summarized with Equation , (see Chapter 4.2.3) but it employs a new estimator for RTT_i (i.e. the RTT estimate for a distance i between the MT and the AP) instead of the $\eta_i - \frac{\sigma_i}{3}$. Figure 4.5 depicts the PDF of the RTT measurements obtained in UDP at 9 meters. In order to better understand which could be the more suitable estimator for RTT_i in the UDP case, the ideal (i.e., free space) RTT_i figure for 9 m (6812.92 cycles of the 44 MHz clock), the standard deviation of the RTT measurements and the Gaussian fit of the PDFs are also depicted.

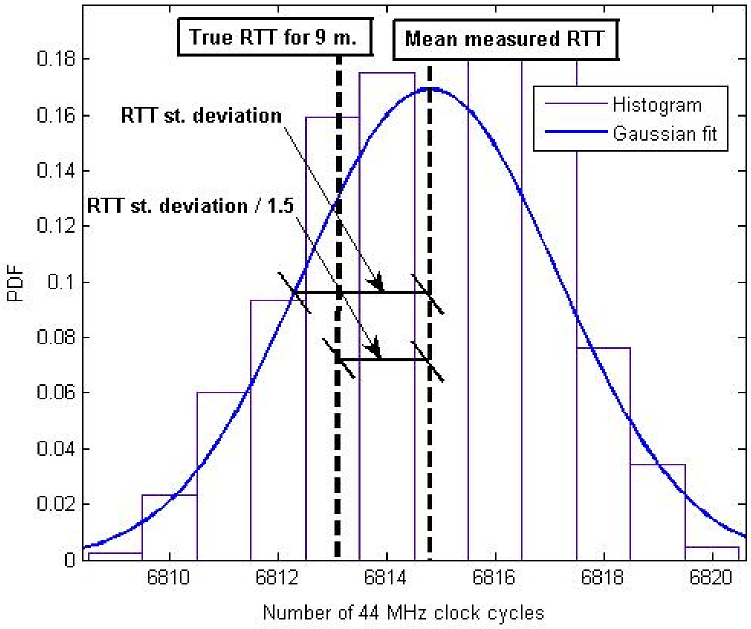

Figure 4.5: Selection of the RTT estimator for UDP (RTT histogram for UDP at 9 m)

The picture suggests that the average minus n times the standard deviation ($\eta - n\sigma$) may be a good estimator of RTT_i for UDP. The value of n=1/1.5 that gives the best results is heuristically obtained and, if kept constant, the relative error remains constant at approximately 10%, as shown in Table 4.2.

Distance (m)	Distance estimate using $\eta - \sigma$	Distance estimate using $\eta - \frac{\sigma}{1.5}$	Distance estimate using $\eta - \frac{\sigma}{2}$
4	1.2582	3.7014	4.9230
6	4.4273	7.0410	8.3478
9	7.4154	10.0859	11.4211
12	7.7925	10.4629	11.7981
15	11.3573	13.4028	14.4255
Average absolute error	2.7497	1.1119	1.2936

Table 4.2: Distance estimations for UDP situations using different estimators for RTT

The estimators $\eta - \sigma_i$, $\eta_i - \frac{\sigma_i}{1.5}$ and $\eta_i - \frac{\sigma_i}{2}$ for RTT_i are tested for distances between 4 and 15 m. Table 4.2 includes the distance estimations for the considered distances in UDP obtained using each one of these estimators. The average absolute errors obtained corroborate the considerations shown in Figure 4.5 because the $\eta_i - \frac{\sigma_i}{1.5}$ estimator provides better accuracy (close to 1 m average error). Hence, the selected ranging algorithm for UDP is the following:

$$d = c \cdot TOA = c \cdot \left(\frac{(\eta_i - \frac{\sigma_i}{1.5}) - \eta_0}{2} \right) \cdot \left(\frac{1}{f_{clk}} \right), \tag{4.2}$$

Thus, the proposed ranging technique employs multi-path sensitive algorithm to estimate the distance, it is: the algorithm represented by Equation 4.3.2 in the case of a DDP or an NDDP situation and the one proposed in this section (Equation 4.2) for UDP. It can be stated that the ranging system is able to provide good ranging accuracy -1.11 m. of error for UDP, 0.89 for NDDP and 0.87 for DDP- in all the possible multi-path situations. This shows a noticeable enhancement of the ranging accuracy with respect to the results presented in Table 4.1 (in that table ranging estimates always using the same algorithm without taking the channel profile into account were reported). This can be easily observed by comparing Figure 4.3 presented above with next Figure 4.6, in which the PDFs of the distance estimations at 9m. for the different situations now using the finally adopted ranging technique are depicted.

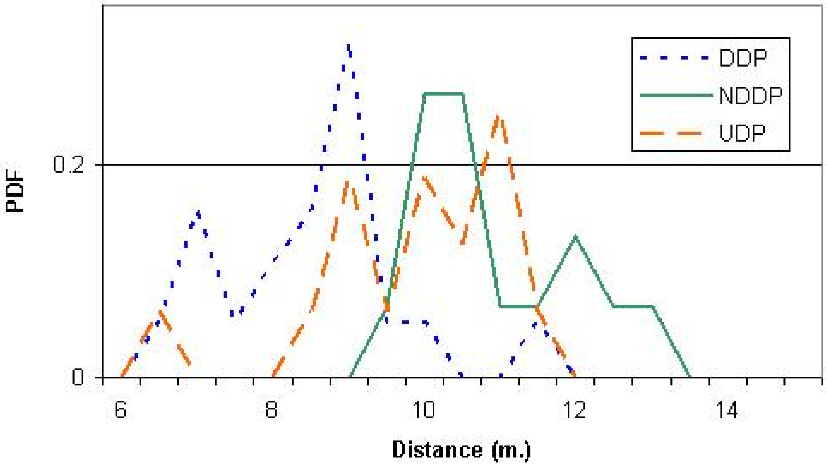

Figure 4.6: PDFs of distance estimation at 9 m for DDP, NDDP and UDP employing the multi-path sensitive algorithm

As a conclusion to this section the bloc diagram of the achieved ranging technique, robust to multi-path and obstructed path problem, is depicted in Figure 4.7.

4.4 Conclusion

The multi-path with the obstructed path problem when performing distance estimations between a MT and a reference point (an AP in WLAN location) is expected to provoke important accuracy degradations. Since in indoor environments is likely to occur such situations when ranging (and then positioning) with a MT, it is important that a ranging method is robust to this phenomenon. Conducted experiments with the developed lab prototype that implements the proposed hardware-based TOA ranging technique corroborates the hyphotesis and show that this phenomenon has a negative impact on the ranging accuracy of the technique in NDDP and UDP (i.e. with severe obstructions) situations, with UDP exhibiting higher error than NDDP. An approach to detect and overcome this problem has been presented in which the identification of the multi-path situation (i.e. the knowledge of the degree of obstruction affecting the direct path) between the MT and the AP is the first and essential step. Once this identification is achieved, a multipath-sensitive ranging algorithm is applied.

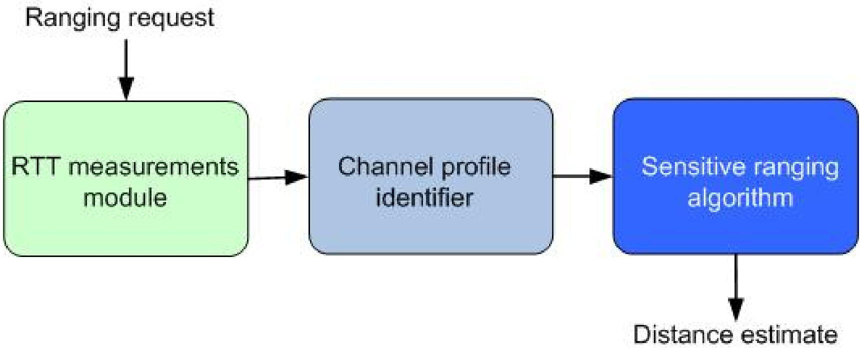

Figure 4.7: Block diagram of the proposed ranging technique

The identification of the situation is feasible in real-time given the collected RTT measurements for the ranging. It is demonstrated therefore the feasibility of multi-path situation identification from IEEE 802.11 MAC layer measurements, establishing a correspondence between physical and MAC layer. The identification is performed by using the standard deviation of the RTT combined with the average as parameters for the differentiation. The identification of the channel profile allows, as second step of the method, the use of specific distance estimation algorithms that depend on the profile, making it possible to obtain high ranging accuracy (e.g., average error close to 1 m). This research work makes possible that the proposed hardware-based TOA ranging technique is thus robust to multi-path and obstructed path problems.

Part III

From ranging to positioning

Chapter 5

Trilateration

5.1 Introduction

Once the distance estimates between the MT to be located and a set of APs have been calculated, and assuming that the coordinates of these APs are known, the position of the MT can be calculated by means of an algorithm. Although the main research work performed during this PhD thesis deals with distance estimation (i.e. ranging) techniques -more specifically TOA-based techniques over WLAN, deeply explained in Chapters 2, 3 and 4- important efforts have been also dedicated to study the position calculation, in order to maximize the reachable location accuracy and availability of the explored indoor location technique. In terms of system design, positioning can be seen as an upper layer with respect to the under-laying ranging layer. Consequently, from the implementation point of view, in practice the positioning part of TOA-based methods corresponds to a software module which is not directly affected by the characteristics of the under-laying network technology as in the case of the ranging part. In fact the basis of most of algorithms employed for the position calculation in indoor location proposals have been available for classical positioning (i.e. for outdoors) since many years ago.

The simplest way to obtain the MT position in TOA-Based location is employing a trilateration algorithm. Trilateration strictly employs the distances and the APs coordinates as input data and tries to obtain the position estimate of the target assuming that the distance estimates are noisy (i.e. contain certain error). On the other hand, tracking algorithms constitute another family of possible solutions to calculate the position. Tracking goes one step further than trilateration and, in addition of using the mentioned input data, tries to benefit from the knowledge of past estimated positions of the target (with trilateration each position calculation is independent from the others). Most of times tracking allows better accuracy and smoother estimated trajectories than trilateration, although its complexity is typically higher. Figure 5.1 shows an illustrative example of the quantitative difference between trilateration and tracking when calculating the positions of a MT. The black line corresponds to the real trajectory followed by the MT, the blue points depict the estimated MT positions employing trilateration and finally the red line is the estimated trajectory with a tracking algorithm. It can be clearly appreciated that while the estimated trajectory provided by trilateration follows erratic jumps, tracking allows a smoother and more accurate estimation.

The first research work in the PhD thesis strictly belonging to the position calculation consisted of the exploration of several trilateration algorithms and their application to indoor positioning. This study has been performed employing the results of the TOA-based ranging technique research explained in Chapter 2. This work can be seen as the first tests of the positioning performance that could be reached with the researched TOA-based techniques over WLAN. The current chapter reports this work. After this work, the logical step was trying to improve the achieved location performance with trilateration exploring the use of tracking algorithms. To this end a more extensive research is done focused on improving the accuracy and the availability of the positioning service, by means of the

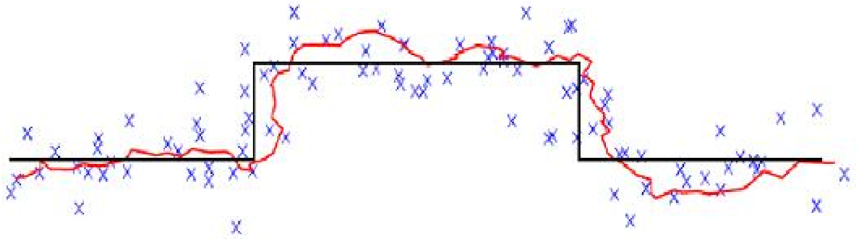

Figure 5.1: Example of trilateration and tracking

design and implementation of a tracking algorithm for our TOA-based location approach. This is reported in Chapter 6.

5.2 Trilateration algorithms

Trilateration in two dimensions (2D) can be seen as a method for determining the intersection of several circumferences known their centers and radii. Applied to a position calculation process, these centers correspond to the known coordinates of the reference points (WLAN APs in our case) and the radii correspond to the calculated distance estimates. Each reference point (and its distance to the MT) defines then a circumference, which can be expressed with the following formula:

$$d_i = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{5.1}$$

where d_j is the distance between the MT and the j^{th} AP, (x_j, y_j) are the coordinates of the j^{th} AP and (x_i, y_i) are the coordinates of the MT (i.e. the unknown variables). The point of intersection of these circumferences is obtained by solving the n nonlinear equations simultaneously to eliminate two coordinates, where n is the number of APs. Due to the non-linearity of these equations, if a solution of n-1 dimensions has to be found, n equations are required. Thus in the case of 2D positioning, at least three APs are needed to determine each position. However, solving a system of n nonlinear equations simultaneously is not feasible because this results in a high-degree nonlinear equation, which makes calculating the MT's position more complex. Linearizing the system of equations geometrically turns the problem into one of finding the point of intersection of several planes. Since approximate distances are used, the solution of the linear system is not completely determined. Consequently, the criterion of minimizing the mean square error between the estimated position of the MT (i.e. the position aimed to be obtained) and the real position has to be determined using specific algorithms if good results are to be obtained. Some trilateration algorithms have been explored.

5.2.1 Linear Least Squares

Although this algorithm is not very accurate, it provides an initial position which can be used afterwards by other positioning algorithms (i.e. Nonlinear Least Squares and Classical Least Squares) as the initialization value for their iterations. First, the system of n non-linear equations mentioned above must be linearized. In order to do so, the j^{th} constraint is used as a linearizing tool [2]. Adding and subtracting x_j and y_j in the typical circumference equation gives

$$(x - x_j + x_j - x_i)^2 + (y - y_j + y_j - y_i)^2 = r_i^2, \tag{5.2}$$

with (i = 1,2,...,j-1,j+1,...n). Expanding and regrouping the terms and selecting the first constraint (j = 1) leads to the following linear system of (n-1) equations with two unknowns:

$$\begin{aligned}(x-x_1)\cdot(x_2-x_1)+(y-y_1)\cdot(y_2-y_1)&=\frac{1}{2}\left[r_1^2-r_2^2+d_{21}^2\right]=b_{21}\\(x-x_1)\cdot(x_3-x_1)+(y-y_1)\cdot(y_3-y_1)&=\frac{1}{2}\left[r_1^2-r_3^2+d_{31}^2\right]=b_{31}\\&\qquad\qquad\qquad\qquad\quad M\\(x-x_1)\cdot(x_n-x_1)+(y-y_1)\cdot(y_n-y_1)&=\frac{1}{2}\left[r_1^2-r_n^2+d_{n1}^2\right]=b_{n1}\end{aligned} \tag{5.3}$$

This linear system can be simplified into a matrix form:

$$A\vec{x}=\vec{b}, \tag{5.4}$$

where

$$A=\begin{bmatrix} x_2-x_1 & y_2-y_1\\ x_3-x_1 & y_3-y_1\\ L & L\\ x_n-x_1 & y_n-y_1\end{bmatrix}$$

$$\vec{x}=\begin{bmatrix} x-x_1\\ y-y_1\end{bmatrix}$$

$$\vec{b}=\begin{bmatrix} b_{21}\\ b_{31}\\ L\\ b_{n1}\end{bmatrix}$$

Once the system has been linearized, and since the distances r_i are only approximate, the problem requires the determination of x such that $Ax \approx b$:

$$A^TA\vec{x}=A^T\vec{b}. \tag{5.5}$$

There are several methods for solving this equation. Since it is assumed that the APs will be logically placed (not in a row or colinearly), this means that is non-singular and well conditioned. Therefore, the method used is the following:

$$\vec{x}=(A^TA)^{-1}A^T\vec{b}. \tag{5.6}$$

5.2.2 Nonlinear Least Squares (Newton)

The nonlinear least squares algorithm uses an initial estimation of the position, which has already been obtained by the Linear Least Squares algorithm, and minimizes the sum of the squares of the errors in the distances. Minimizing the sum of the square errors is a fairly common problem in applied mathematics for which various algorithms are available. The Newton iteration was selected to find the 'optimal' solution P(x, y). Introducing f, g and the Jacobian matrix J, and applying the Newton iteration, gives

$$\vec{R_{k+1}} = \vec{R_k} - (J_k^T J_k)^{-1} J_k^T \vec{f^k}, \tag{5.7}$$

where $\vec{R_k}$ denotes the k^{th} approximate solution. The subscript k in J and $\vec{f}$ means that these quantities are evaluated at $\vec{R_k}$. Obviously, $\vec{R_1}$ corresponds to the mentioned initial estimation of the position obtained by the Linear Least Squares. Finally, using the explicit form of the function $f_i(x, y)$ gives:

$$J^T J = \begin{bmatrix} \sum_{i=1}^{n} \frac{(x-x_i)^2}{(f_i+r_i)^2} & \sum_{i=1}^{n} \frac{(x-x_i)(y-y_i)}{(f_i+r_i)^2} \\ \sum_{i=1}^{n} \frac{(x-x_i)(y-y_i)}{(f_i+r_i)^2} & \sum_{i=1}^{n} \frac{(y-y_i)^2}{(f_i+r_i)^2} \end{bmatrix}$$

$$J^T \vec{f} = \begin{bmatrix} \sum_{i=1}^{n} \frac{(x-x_i)f_i}{(f_i+r_i)} \\ \sum_{i=1}^{n} \frac{(y-y_i)f_i}{(f_i+r_i)} \end{bmatrix}$$

Using these matrices and equations, the algorithm performs iterations until the difference from the previous iteration and the next one is smaller than δ which is a completely modifiable threshold.

5.2.3 Classic Least Squares

This algorithm is known for solving the navigation equations of the GPS system [68] without using the Kalman filter. The algorithm needs an initial approximate position of the device to be located. The observable for each AP is the prefit residual, which is the difference between the measured and the modeled estimation of the distance. In our case, the modeled distance corresponds to the distance between the AP and the initial estimated position. The solution to these equations (navigation equations in terms of GPS) is the correction that must be applied to the initial position:

$$P_j + c \cdot dt_j - \delta_j = \sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2} + c \cdot dt, \tag{5.8}$$

where j=1,2,...,n (n=4). Every equation is linearized using the Taylor approximation around the initial position, in order to apply the least squares method to solve the equation system. Applying this method yields to:

$$\begin{bmatrix} P_1 - \rho_{1,0} + c \cdot dt_1 - \delta_1 \\ M \\ P_n - \rho_{n,0} + c \cdot dt_n - \delta_n \end{bmatrix} = \begin{bmatrix} \frac{x_0 - x_1}{\rho_{1,0}} & \frac{y_0 - y_1}{\rho_{1,0}} & \frac{z_0 - z_1}{\rho_{1,0}} & 1 \\ M & M & M & M \\ \frac{x_0 - x_n}{\rho_{n,0}} & \frac{y_0 - y_n}{\rho_{n,0}} & \frac{z_0 - z_n}{\rho_{n,0}} & 1 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \\ c \cdot dt \end{bmatrix}$$

The algorithm is applied iteratively for better results. It reconstructs the navigation equations by using the position obtained and recalculating the new iteration position, until the difference between the results from one iteration and those of the next is smaller than δ, which once again is a completely modifiable threshold.

5.3 Experiments with trilateration

5.3.1 Methodology

The objective of these experiments is comparatively evaluating the commented trilateration algorithms (Classic and Non Linear Least Squares, using the Linear Least Squares algorithm to obtain the needed initial position estimates) when applied to location of WLAN enabled MTs. More specifically, the positioning accuracy that can be reached employing each one of the algorithms is tested, taking as distance estimate inputs the results obtained with the researched TOA-based ranging technique explained in Chapter 2. In order to perform this assessment, the algorithms are implemented using Matlab. Then several simulations considering a basic scenario of three APs and one MT to be located are performed. The accuracy statistics of each simulated position calculation are obtained according to the following methodology:

1. The positions of the three APs and the true position of the MT are configured in the Matlab-based trilateration simulator.

2. The estimation of the distance between the MT and each one of the three APs for obtaining then a single position estimation is simulated like this:

   (a) The true distance from each AP to the MT is obtained given the position of the MT and the AP.

   (b) The distance estimation is simulated taking the mentioned exact distance and applying the Gaussian pdf of the distance estimated with the hardware TOA-based ranging method in order to obtain a sample (see Section 2.4.4 for details about this pdf). This can be seen as a RV, which is the distance estimation, and a sample of this RV which corresponds to the distance estimate obtained at this specific moment. As deeply explained in Section 2.4.4, this pdf was obtained from true measurements performed with the implemented ranging prototype. The same type of pdf is used for all distances because previous results (Section 2.4.2) show that there are no major statistical variations depending of the true distance, but obviously the pdf average varies depending on the true distance, as represented in Figure 5.2 for an example of simulation with APs at distances of 6, 9 and 14 m. from the MT.

3. The position of the MT is estimated taking as input data the three estimated distances and the APs coordinates, employing the specific trilateration algorithm under study.

4. Steps 2 and 3 are repeated with the same static position of the four involved WLAN nodes, so that a large number of MT's position estimations are performed with the trilateration algorithm. The key idea is that in every position estimation the distance estimates are different, since are obtained from a RV pdf., so that all the different combinations of the distance estimations to the three APs are employed. After each single position calculation, the obtained value is subtracted from the MT's real position to find the position estimation error.

5. At the end of the process a large number of position estimation errors for this single static scenario are obtained. Then they are used to obtain the cumulative probability function (cdf) of the position estimation error.

The simulations considered several physical scenarios (i.e. different ways of placing the four involved WLAN nodes) because the relative geographical situation between the MT and the three APs affect the reached positioning accuracy. Since APs are assumed to be rationally deployed (non-colinearly, for instance), the Geometric Dilution Of Precision (GDOP) [69] in representative scenarios is expected to be good.

5.3.2 Results

In a scenario in which the MT is located within the triangle formed by the three APs (i.e. best case), accuracy is superior to 1.4m. for the 66% (2 m. for the 90%) of the cases. The CDF of the obtained positioning error is shown in Figure 5.3, with the absolute positioning error in meters represented on the x-axis.

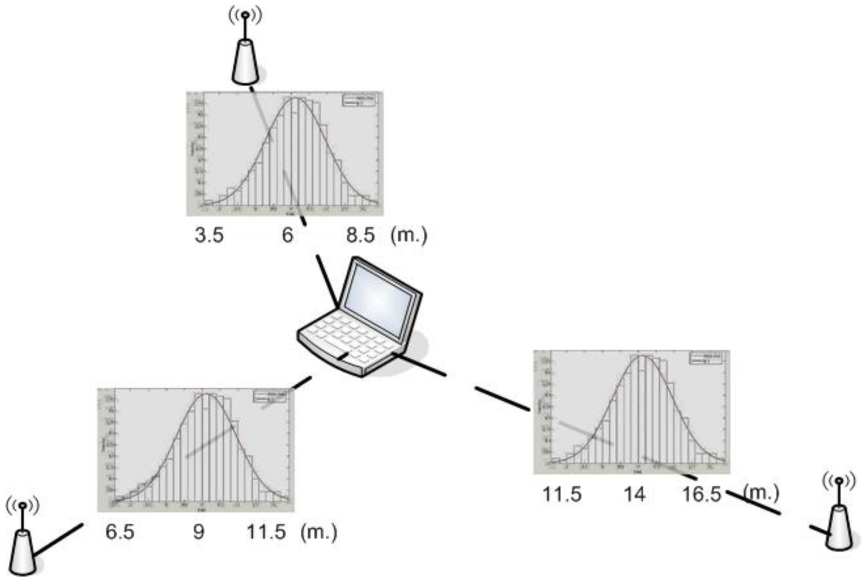

Figure 5.2: Simulation of TOA-based trilateration

In a situation in which the MT is not within the triangle of APs but APs are properly deployed (i.e. GDOP is not bad, no alignment of APs) accuracy is better than 1.8 m. with a probability of 66% (Figure 5.4). It can be also seen that the Nonlinear Least Squares (Newton) algorithm slightly outperforms the Classic Least Squares algorithm in both cases.

5.4 Conclusions

This chapter has presented my first work about how to compute the MT's location in real-time once the distances from the MT to several APs have been estimated taking advantage of the WLAN infrastructure. This work explores the simplest way to achieve positioning in TOA-based location, which is employing a trilateration algorithm taking as inputs the estimated distances. Positioning is the direct and main application of the achieved results on TOA-based ranging (Sections 2, 3 and 4).

Some known trilateration algorithms have been implemented and tested through simulations, taking as distance estimations the ones obtained with the TOA-based hardware lab prototype (Section 2). Obtained results, a positioning accuracy better than 2m. for the 90% of the cases, show the feasibility of locating WLAN devices with good accuracy using a TOA-based technique (with the advantages in terms of flexibility, simplicity, etc that allows this location technique with respect to the others) that only requires minor hardware modifications over WLAN consumer equipment. This technique can be summarized in two parts: the ranging module to calculate the distances between the MT and the APs (see Section 2), and the positioning module to compute the MT's position, basically compounded by the Newton trilateration algorithm.

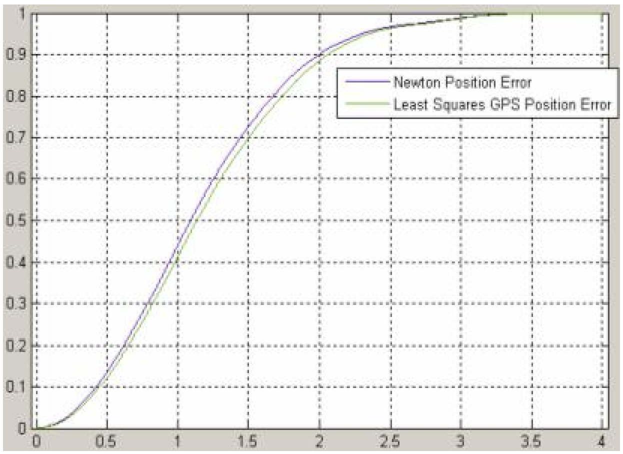

Figure 5.3: CDF of the positioning error with trilateration (good GDOP)

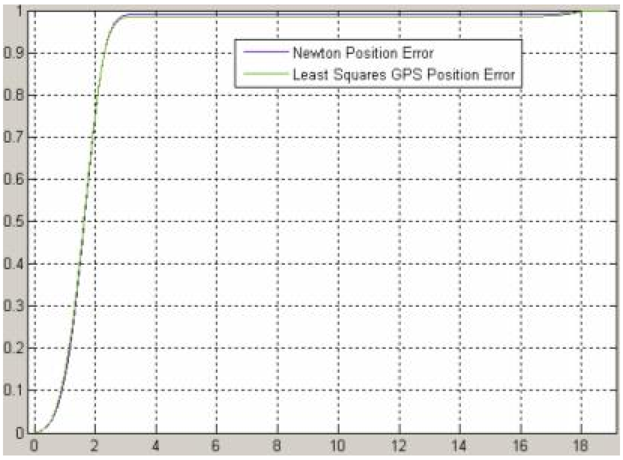

Figure 5.4: CDF of the positioning error with trilateration (bad GDOP)

Chapter 6

Tracking

6.1 Introduction and goal

Once assessed and known the performance that can be reached employing trilateration on the researched WLAN TOA-based location technique, the main goal of the research became adding tracking features to the mentioned proposal in order to enhance the positioning accuracy of the system, replacing the existing trilateration module with a tracking module. By applying tracking principles it is possible to benefit from past estimated positions. Availability can be also enhanced by allowing localization even in constrained situations with only two APs in sight (instead of the three APs commonly needed). The idea is that the real-time tracking module includes two different algorithms for both situations: the first is based on Kalman filtering to increase the accuracy with three or more APs, and the second is specifically designed to provide the location when only two APs are in range of the MT. Thus, when a position has to be calculated, the proper tracking algorithm is first selected depending on the number of available distance estimates provided by the ranging module as depicted in Figure 1. Without loss of generality, the ranging and tracking modules are assumed to be implemented in the MT; this maximizes scalability and user privacy. Hence, the designed tracking algorithms must show a low complexity to allow implementations in energy-constrained and processor-limited terminals.

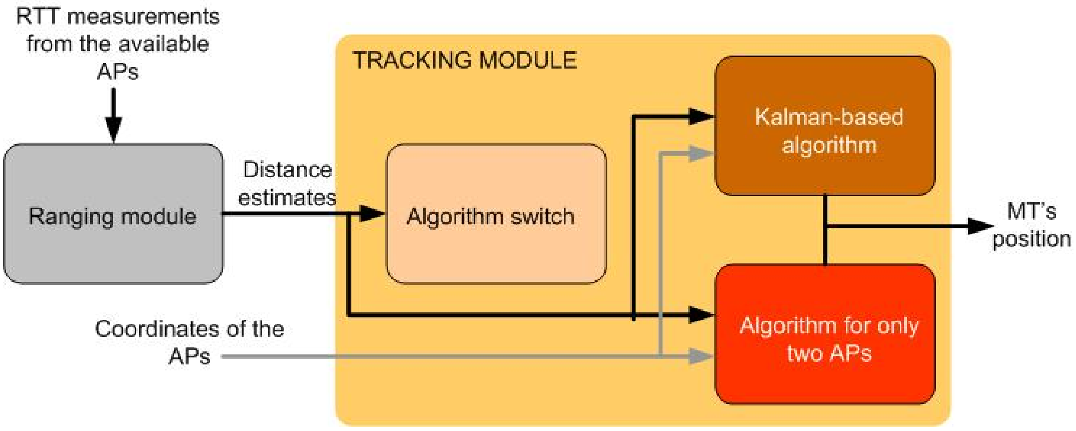

Figure 6.1: Modular diagram of the TOA-based tracking technique

6.2 Related work

The use of Kalman algorithms to track mobile devices has produced vast literature [70] but it has been scarcely applied to localization in indoor environments. In several reports that propose Kalman filtering for mobile location, the algorithm is applied to smooth the data measurements instead of using it to directly obtain the target location estimate. In [71], a biased Kalman filter is used to mitigate the effect of Non Line of Sight (NLOS) in TOA based systems. In [72], a received signal strength indicator (RSSI) based technique is enhanced with an EKF based on pre-calibration of measurement vectors and individual position block usage in final estimation of the target's position. A proposal to apply EKF to estimate patients' location in a hospital is presented in [73]. In [74] the use of the EKF for outdoor localization taking TOA measurements as inputs is presented with no optimization with respect to the classical EKF approach. The Kalman Filter has also been applied to enhance the tracking performance of the WLAN fingerprinting positioning technique: a representative proposal can be found in [75], while in [76] the performances of Kalman and Particle filters for WLAN fingerprinting are compared for indoor location. Kalman Filter has also been employed for data fusion of different location technologies, e.g. in [77] a system that uses inertial sensors and a location method based on UWB is presented. In all above-mentioned approaches, Kalman filtering seems to be a suitable tool for mobile tracking, but the achieved accuracy is poorer than 1-2 m of error. In addition, the position cannot be computed when only two reference stations are available. This research work demonstrates that it is possible to achieve accurate and robust location estimation with simple optimizations of this algorithm, even in situations of poor reference point availability.

6.3 The Kalman-based tracking algorithm and adaptation

Despite the existence of more sophisticated filters ([15, 78]), Kalman filtering was chosen as the basis of this algorithm because it minimizes estimation error variance when the observables -ranging measurements in our case- are inaccurate. The probability distribution of the measurement noise in the considered ranging system is Gaussian (see Section 2.4.4), so the premises assumed by the filter are satisfied. We intend to use a Kalman-based filter, taking the MT location as the state vector and the distance estimates as observables. The Extended Kalman Filter (EKF) is used because the equations that relate the state vector to the measurements are non-linear and need to be linearized. Using tracking algorithms to refine TOA or distance estimates in order to mitigate errors due to NLOS conditions remains outside the scope of this work. The reader is addressed to [79] for a detailed description of the Kalman filter.

The Kalman filter bases the state estimation on the weighted average between the measurement Z_k at time $t = t_k$ and the prediction of the state from the state estimate obtained in the previous epoch ($t = t_k - 1$). Because we intend to apply the filter to track a target's trajectory, the state corresponds to the target's position and the measurements (i.e. the observables) are the noisy distance estimates between the target and the APs. This weighted average estimation process at a given time works in two steps: a) the filter estimates (i.e. predicts) the current position from the past ones (time update or prediction step), b) it obtains the feedback from the noisy measurements in order to improve the accuracy of the estimate (measurement update or correction step). The equations corresponding to each step are detailed in Figure 6.2. The weight of every source of information depends on its reliability. The first equation of the prediction step represents the linear process that models the trajectory of the MT; the process noise has covariance matrix Q. The matrix R that appears in the first equation (Kalman gain K_k equation) of the correction step corresponds to the covariance of the measurement noise. Both noises are assumed to be independent, white, and normally distributed. The matrix A of the prediction step corresponds to the transition state matrix. Q and A should be chosen depending on the motion behavior of the target, as shown below. The other matrices are assumed to change at each step: H is the measurement matrix; K is the Kalman gain matrix, which minimizes the final device position estimation error at each time; and P is the covariance matrix of the position estimation error, which is updated at every step. The algorithm needs an initial estimation of the MT's position and the P matrix.

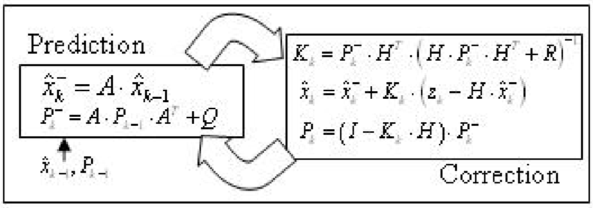
Prediction
$\hat{x}_k^- = A \cdot \hat{x}_{k-1}$
$P_k^- = A \cdot P_{k-1} \cdot A^T + Q$
$\hat{x}_{k-1}, P_{k-1}$
$K_k = P_k^- \cdot H^T \cdot (H \cdot P_k^- \cdot H^T + R)^{-1}$
$\hat{x}_k = \hat{x}_k^- + K_k \cdot (z_k - H \cdot \hat{x}_k^-)$
$P_k = (I - K_k \cdot H) \cdot P_k^-$
Correction

Figure 6.2: Scheme of the discrete Kalman filter

6.3.1 Improvement of the prediction step

The Kalman algorithm adaptation defines in detail the prediction step and makes decisions to optimize some specific points of the algorithm design. Q and A have to be chosen in order to model the motion behavior of the MT. These matrices can be defined following three motion models: static, cinematic, and random walk. Assuming the target's average speed v=1 m/s (e.g. typical for pedestrian motion), the random walk model is a suitable candidate. In this case, the A matrix is the identity one, and Q depends on T (i.e. the scalar time elapsed between two consecutive position estimates) and v. Conceptually, this value of A means that the MT does not vary its position during T while Q is used to model the variation. Hence, Q is a diagonal matrix where each non-zero value corresponds to the squared maximum variation of a coordinate during T:

$$Q = \begin{bmatrix} (vT)^2 & 0 \\ 0 & (vT)^2 \end{bmatrix}$$

Initial tests performed showed that the Kalman algorithm including the described prediction step provides low accuracy enhancement with respect to pure positioning. To improve performance, work was directed towards enhancing the prediction of the next position from the past ones, using a geometrical approach. This new approach assumes that the target is going to follow the straight trajectory defined by the line that joins the last two estimated positions, with the same speed and direction, as shown in Figure 6.3. Regarding the Kalman equations, the first prediction step is replaced by this new estimation. Since the order of the filter is increased (i.e. now the position estimated in $t = t_k - 2$ is also used for the prediction), when the tracking subsystem starts running using this algorithm, the first two position estimates are calculated using the Newton trilateration algorithm [2], and it is in the third one when the filter really starts working. Another aspect of this approach is the estimation of the target speed v. In order to guarantee accuracy, target speed is estimated from the last n position estimations, performing an average between the last n speeds taking pairs of consecutive positions. Simulations proved that using n=5 is sufficient for an accurate average.

This change of the prediction step involves a redefinition of the prediction error covariance matrix Q. Now Q represents the estimation error present if the MT follows the supposed straight trajectory. This error depends on the accuracy of the two last position estimations and the accuracy of the v estimation. For this reason, it is difficult to predict the value of this matrix and it has been adjusted empirically from the simulations.

6.3.2 Design issues

For Kalman-based algorithms, an initial MT position estimate is necessary when the filter starts working. This initial position is also necessary at the beginning of each correction step in order to obtain

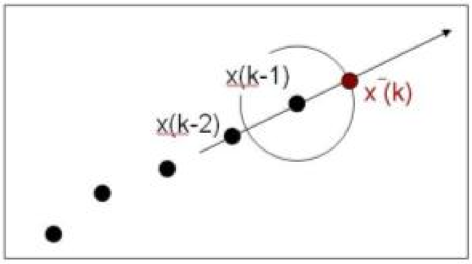

Figure 6.3: Geometrical scheme of the improved prediction step

the measurements matrix H and the measurements prefit residuals vector Z_k through a Taylor approximation (EKF). Hence, Newton with Linear Least Squares was chosen, because initial position estimation is not required and accurate position estimations can be provided without high computational cost. When carrying out the preliminary tests of the algorithms, we realized that it was desirable to perform several iterations of the filter's correction step to achieve a better position estimate. Five iterations proved to be sufficient.

6.4 The algorithm for situations of only two references

6.4.1 Problem description

This algorithm deals with situations in which classical positioning algorithms based on trilateration [2] are not able to provide estimates. As explained in Section 5.2, at least three APs are needed to calculate a 2D position with a trilateration-based method. If there are only two references, trilateration algorithms cannot be applied because of the ambiguity generated by the two intersections of the two circles (see Figure 6.4), which correspond to the two solutions of the nonlinear equation system (one equation for each reference point). The algorithm proposed here discards one of these two solutions keeping only the solution closer to the last estimated position. Next section explains this approach in detail.

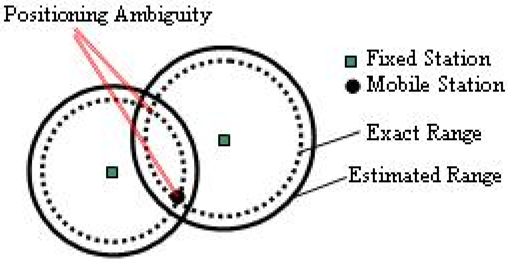

Figure 6.4: Trilateration in 2D with two reference points

6.4.2 Proposed algorithm

The proposed algorithm estimates the position of the MT in several stages:

1. Obtain the estimation of the two intersection points from the two available circle equations (see Equation 5.1), once knowing the coordinates of both APs and the distances from the MT (i.e. the observables provided by the ranging module).

2. Select one of the two intersection points according to the criterion of minimum distance between the selected point and the last estimated position of the MT, as depicted in Figure 6.5.

3. Predict the MT position based on the trajectory followed by the MT and the assumed motion model, without using any observable.

4. Combine the position estimations obtained with Stages 1-2 (i.e. position from observables) and 3 (i.e. position from prediction) to obtain the final estimation of the MT position. The specific weight assigned to each source is inversely proportional to its level of uncertainty and inaccuracy.

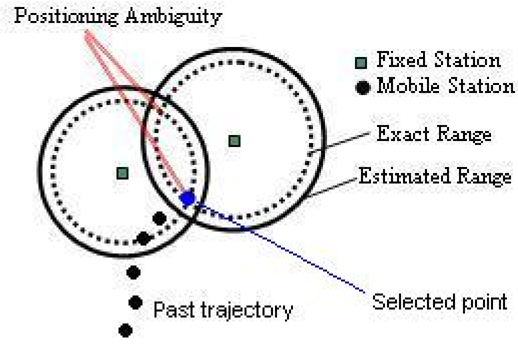

Figure 6.5: Solving the ambiguity of trilateration in 2D with two reference points

The prediction in stage 3 corresponds to the approach employed in the improved prediction step before detailed in Section 6.3.1. The combination process in stage 4 is as follows. The expression to obtain the final estimated position is the weighted average of the position obtained from the observables ($\hat{x}_o$) and the one obtained from the prediction ($\hat{x}_p$), so that a linear combination of these two variables is employed. This idea is behind well-known tracking algorithms such as Kalman and particle filters [18], specifically in the correction (i.e. update) step. The expression is straightforward:

$$\hat{x} = C_o \cdot \hat{x}_o + C_p \cdot \hat{x}_p, \tag{6.1}$$

where C_o is the coefficient for the position based on observables and C_p is the coefficient for the predicted position. Each coefficient represents the uncertainty and inaccuracy of the associated estimation. The coefficients must be normalized so that they add up to unity. It is important to find suitable values for each weight coefficient in order to achieve accurate estimations. In our case, the mean of the positioning error (ϵ) is used as the statistical parameter. Regarding the expression that defines the coefficient, two different functions have been tested: the inverse and the exponential. Both were selected because they are simple and monotonically decreasing with the error. Equations 3 and 4 show the expressions of Equation 6.1 using the aforementioned function:

$$\hat{x} = k \cdot \left(\frac{1}{\epsilon_o}\right) \cdot \hat{x}_o + k \cdot \left(\frac{1}{\epsilon_p}\right) \cdot \hat{x}_p, \tag{6.2}$$

$$\hat{x} = m \cdot e^{-\epsilon_o \cdot f} \cdot \hat{x}_o + m \cdot e^{-\epsilon_p \cdot f} \cdot \hat{x}_p, \tag{6.3}$$

where k and m are the normalization factors. If the inverse function is used, large values of ϵ entail very small values of the C coefficient, so the weight of the associated position will be very low.

On the other hand, taking values smaller than 1 for the scaling factor f the use of the exponential function would allow a more balanced behavior, due to the smoother slope of the resulting function with respect to the inverse. The value of this scaling factor must be properly tuned.

To obtain the weight coefficients, the expected errors eo and ep must be obtained first. The approach used consists of estimating the errors computationally through simulations, i.e. to find the mean positioning error from a large number of route simulations when (a) only prediction is used and (b) only the observables are taken into account, to calculate ep and eo respectively. Thus, a large number of routes for the MT were generated according to the MT motion model defined in the overall system test and positions were estimated every step T. In the calculation of eo, the position was estimated after the two first stages of the algorithm: the distance estimation from the MT to the APs was emulated (detailed in Section 5) and both circle intersections were obtained, one of which was selected as the final position estimation. The absolute error was computed for each position and then averaged for all points on all routes; the obtained value of eo was 1.75 meters. To calculate ep, the same routes were used and the positions were estimated in this case with just the prediction mechanism, it is without taking into account the observables. The estimations were performed using the previous two real positions. The mean absolute error was computed in the same way and the figure obtained was ep = 0.35 meters.

6.5 Simulations setup

Simulations of MTs following different trajectories from the same mobility model and being tracked using the developed algorithms have been carried out in order to asses the accuracy of the system. The scenario consists of a square area of 50 x 50 meters with APs placed in two or four corners, depending on the algorithm to be tested. This scenario is simple enough to assure that the MT and the AP are in Line Of Sight (LOS) situation. In addition, this is a typical industrial scenario in which location-based applications can be deployed. The time interval T is set to one second. The motion model used is a pedestrian model that ensures the existence of different types of route intervals including changes of direction in order to fairly evaluate the algorithms, and is characterized according to the following rules:

- The probability of changing direction at a given time is governed by a geometric random distribution, with probability of change equal to 0.3.
- The speed of the MT follows a normal random variable of mean 1 m/s and variance 0.2 m/s. The speed is allowed to vary only when the direction changes. Drastic variations of speed are not considered under test in this contribution.
- The change in direction is uniformly distributed between 0 and ±45 degrees with respect to the former direction.

Figure 6.6 shows an example of a generated route. As in the performed experiments with trilateration, the observables employed in the simulations correspond to the ranging model obtained with the developed hardware-based WLAN TOA lab prototype (see section 2) and an AP in LOS situation indoors, which are the proper conditions given the tracking simulations scenario. As explained, this ranging model follows a Gaussian distribution of the error (see Section 2.4.4.

Two different types of situations have been considered:

1. At least three APs in range of the MT. Four APs are placed at the four corners of the test layout. If more than three APs are in range, the MT uses the three nearest ones. The EKF-based algorithm is used to track the MT, considering the two variants explained in Section 6.3 for the prediction step:
   (a) Kalman-1, incorporating the classical prediction step
   (b) Kalman-2, incorporating the novel proposed prediction step

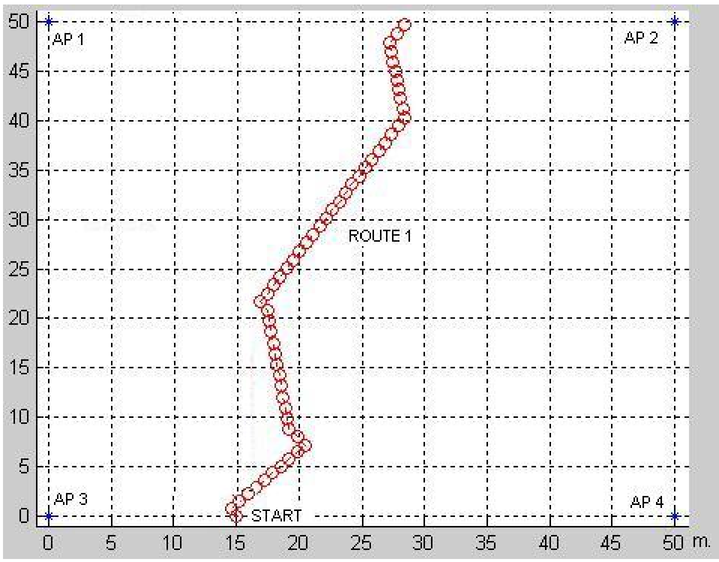

Figure 6.6: Example of generated route for simulation

Moreover, the Newton trilateration algorithm has also been simulated in order to evaluate the gain of tracking results versus a pure positioning technique.

2. Only two APs are in range of the MT. The APs are placed at the two upper corners of the layout. The algorithm specifically designed for such situations is employed to track the MT. The inverse and exponential functions proposed in Section 6.4.2 to obtain the weight coefficients C_o and C_p from ϵ_o and ϵ_p are evaluated.

For both situations, the tracking accuracy is tested along with the capacity of the tracking algorithm to react to changes in direction and its behavior with respect to the GDOP [80]. GDOP is a dimensionless parameter that relates the geometry of the MT and the reference points to the error of the position estimated with distance estimates from the MT to the mentioned references. The best case for tracking algorithms is a straight trajectory with APs placed in such a way that a good GDOP is available. However, in actual indoor situations, sub-optimal cases (i.e. change of direction or speed and poor GDOP) are likely to occur often. The generated routes ensure regions with bad GDOP due to the collinear situation between the MT and two of the APs involved in the position calculation.

6.6 Tests and results

6.6.1 At least three APs in range of the MT

In Figure 6.7, the performance of the tested algorithms can be appreciated while tracking a MT that follows the route described above. The figure shows the average and standard deviation of the position error in meters and GDOP for each point of the route; the points in which changes of direction occur are highlighted (asterisk in the figure). The position error corresponds to the Euclidean distance between the actual MT's position and the position estimate calculated by the algorithm. The GDOP is calculated using this expression:

$$GDOP = \frac{\sqrt{\sigma_x^2 + \sigma_y^2}}{\sigma_m}, \tag{6.4}$$

where σ_x and σ_y are the horizontal position estimation uncertainties, and σ_m is the uncertainty of the ranging error (for details about GDOP calculation see [23]). The average of the position error for each position of the route is calculated with:

$$average\_error = \frac{\sum_{i=1}^{N} e_i}{N}, \tag{6.5}$$

where e_i is the position error for the simulation run i and N is the number of performed simulation runs. The standard deviation of the position error is calculated with:

$$std\_error = \sqrt{\frac{\sum_{i=1}^{N}(e_i - \mu)^2}{N-1}}, \tag{6.6}$$

where μ is the average of the position error. The number of simulation runs for each trajectory is 500, a large enough amount to ensure the statistical reliability and accuracy of the obtained results.

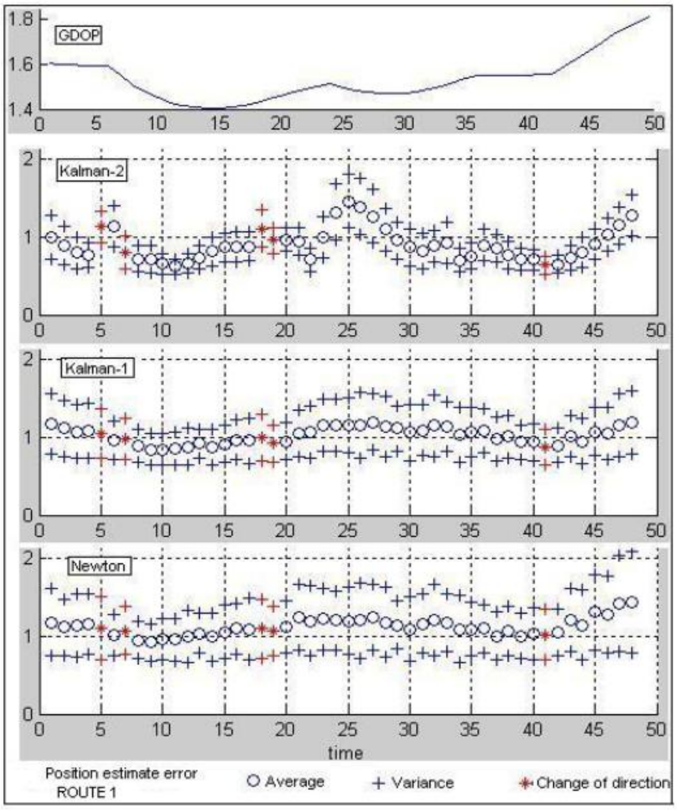

Figure 6.7: Average and standard deviation of the position estimate error and GDOP along the generated route for situations of at least three reference points

In order to avoid edge effects, the initialization period of the algorithms has not been taken into account, so the first position represented corresponds in fact to the sixth position of the route. In terms of accuracy, the Kalman-2 algorithm outperforms the other two because the average position estimate error remains lower than 1 m in most of the points, while Kalman-1 and especially the Newton trilateration algorithm provide errors greater than 1 m. As expected, the pure positioning algorithm without the inclusion of prediction and tracking performs the worst.

Analyzing the error figures after changes of direction, it can be observed that Kalman-1 performs better than Kalman-2 in situations of bad GDOP. This was expected due to the more constrained prediction step of Kalman-2, which forced the prediction on the straight line drawn by the past trajectory in addition to using more position memory than Kalman-1. However, in situations of good GDOP, both algorithms respond well to changes of direction. If we focus on the performance dependence with the GDOP when the actual trajectory describes an almost straight line, it can be seen that the performance degradation due to bad GDOP is a little bit higher in Kalman-2 than in Kalman-1. However, even in this case, the average error figures provided by Kalman-2 are not worse than the ones provided by Kalman-1. As expected from the discussed prediction steps, Kalman-2 provides the highest accuracy during straight intervals (around 0.7 m of average error). Furthermore, the standard deviation of the position estimate error obtained with Kalman-2 is always the lowest.

Figure 6.8 shows an interval of the actual trajectory and the estimated trajectories obtained with the Newton and Kalman-2 algorithms. The former provides an erratic path whereas the latter is able to achieve a smoother trajectory closer to the actual one.

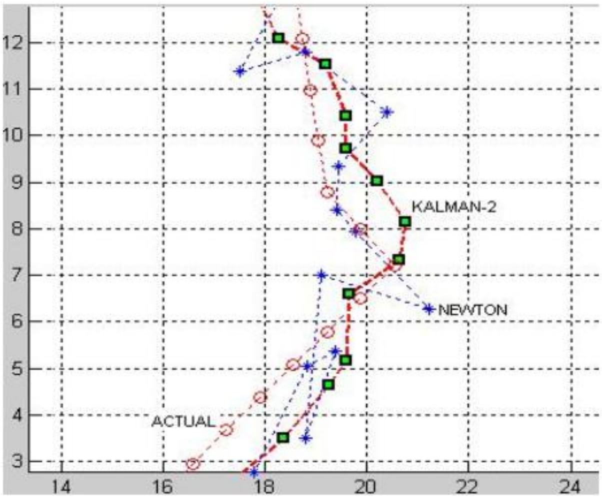

Figure 6.8: Actual and estimated trajectories for situations of at least three reference points

Finally, a large number of routes (5000) following the same mobility model are generated and tracked using the algorithms, in order to obtain statistically reliable accuracy figures. Figure 6.9 shows the Cumulative Distribution Function (CDF) of the absolute positioning error for the tested algorithms. The Kalman algorithm with the improved prediction step provides the best accuracy: less than 0.9 m of positioning error for the 66% of cases and less than 1.4 m for 90% while Newton provides 1.2 m and 1.8 m for the 66% and 90% benchmarks, respectively.

6.6.2 Only two APs in range of the MT

The generated routes cross the squared area following a trajectory governed by the described motion model and are estimated using the four-stage algorithm presented in Section 6.4.2. The two alternatives (i.e. the inverse and the exponential functions) to implement the combination process (Stage 4 of the algorithm) are tested. For every route tracking simulation using a specific alternative, a large number of runs are performed to obtain statistically representative results. For each alternative, the mean absolute positioning error in each step of the route is depicted.

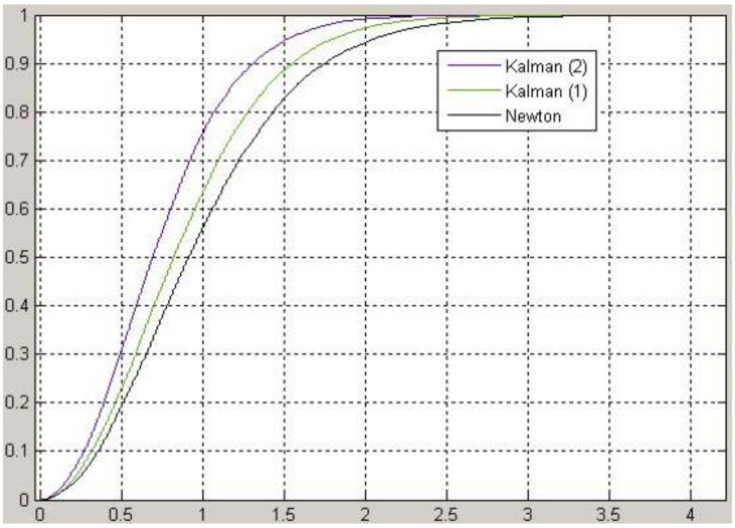

Figure 6.9: CDF of the absolute positioning error for situations of at least three reference points

6.6.3 Weight coefficients obtained with the inverse function

Figure 6.10 shows how the mean positioning errors are closely correlated to the ones obtained with just the prediction for all the steps in the route, and clearly differ from the ones obtained using the observables data. As discussed in Section 6.4.2, this behavior was expected because the inverse function keeps the prediction weight coefficient C_p very high while C_o was low. In addition, in many steps, the error of the final estimated position is higher than the error obtained just from the observables. These facts show that the weight coefficients are not properly adjusted with this function.

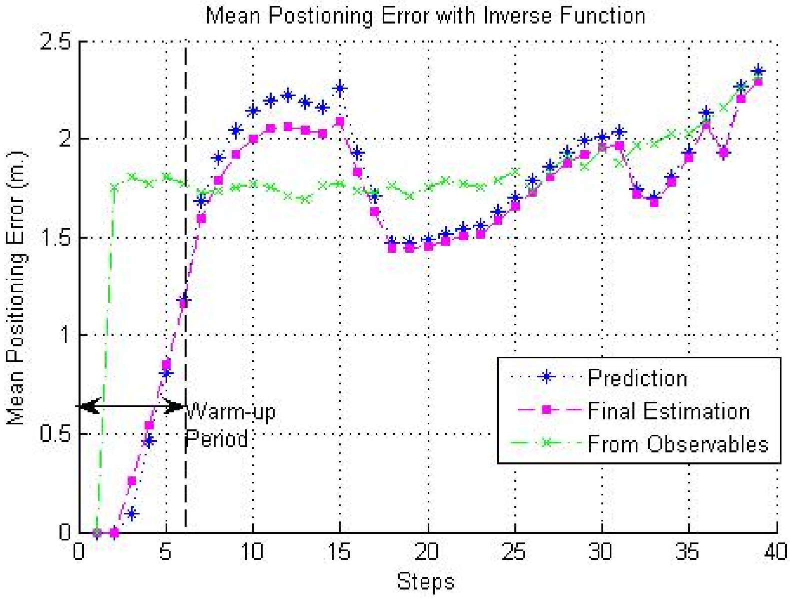

Figure 6.10: Mean positioning error with the inverse function for situations with two reference points

6.6.4 Weight coefficients obtained with the exponential function

The value of the scaling factor f for the exponential function in Equation 6.3 is tuned following a trial-and-error approach. The first attempt was with f=1, but it did not significantly improve the behavior with respect to the before tested inverse function. Then values smaller than 1 were tested in order to smooth the slope of the exponential function, being f=1/3 the value that provided best results for our mobility model. Figure 6.11 shows that the use of this function with this value of the scaling factor results in a more desirable behavior of the combination of the two sources of information (observables and prediction) with respect to the functions tested beforehand. In most of the steps, the error of the final estimation is lower than the one provided by the observables. However, as with the other functions, when the MT nears the line joining the two APs, the GDOP worsens and the accuracy decreases. It can also be observed that the prediction error is smaller with respect to the previously tested functions. This is because the prediction is obtained from the previous final position estimates and therefore, the achieved improvement on the combination process affects the prediction as well.

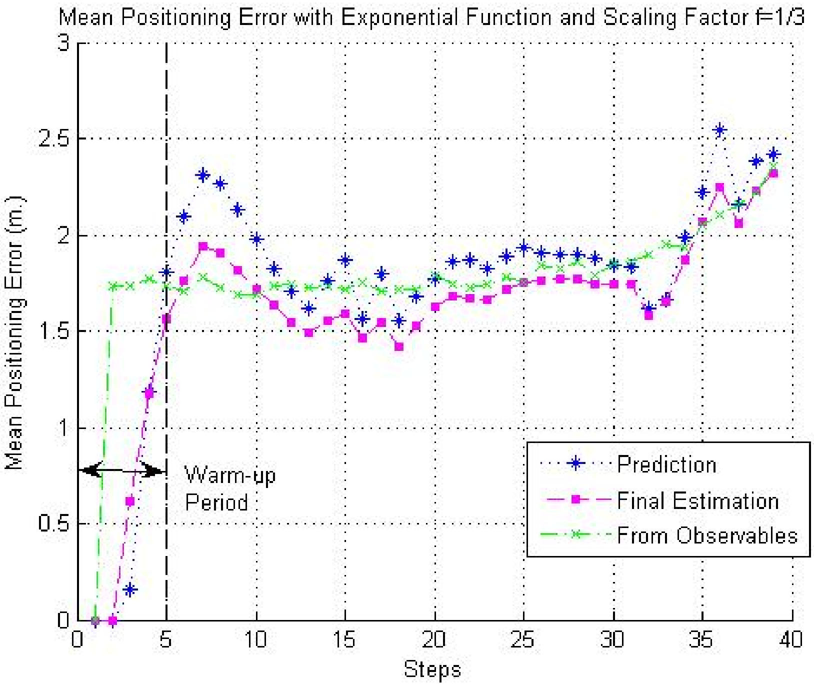

Figure 6.11: Mean positioning error with the exponential (f=1/3) function for situations with two reference points

In Figure 6.12, a representative portion of the estimated routes is depicted. As expected, the trajectory obtained from the observables shows erratic jumps. The predicted trajectory does not always react properly to changes in direction and the final estimated trajectory shows a combination of smoothness and accuracy in the areas in which there is a change in direction.

Figure 6.13 shows the CDF of the positioning error obtained with the proposed algorithm employing the inverse and the exponential (with f=1 and f=1/3) functions. This figure corroborates the qualitative observations and indicates that the exponential function with f=1/3 provides better results than the other functions because it allows an accuracy improvement of around 0.5 m for 80% of the cases.

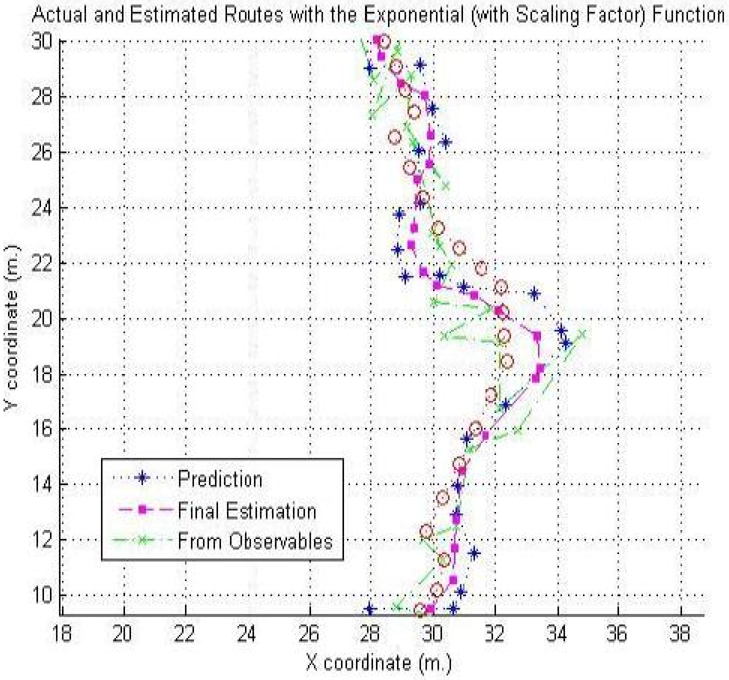

Figure 6.12: Routes estimated with the exponential function (f=1/3) for situations with two reference points

6.7 Conclusions

This chapter has presented my research work about how to track WLAN-enabled MTs in real-time once the distances from the MT to several APs have been estimated taking advantage of the WLAN infrastructure. The results of the previously performed study about trilateration (Section 5) is the start point of the current research, because the use of tracking algorithms has been considered in order to improve the previously achieved positioning performance with trilateration.

The Kalman Filter has been taken as the basis for obtaining an algorithm that provides good accuracy when at least three APs are in sight of the MT. In addition, an algorithm designed for situations with only two APs has been proposed, which solves the ambiguity of raw trilateration by applying tracking principles that can utilize past trajectory information. Both algorithms have been optimized and tested using simulations fed again with actual ranging data from the hardware TOA-based lab prototype. The tested performance parameters include positioning accuracy and robustness to changes of direction and GDOP. The results show a substantial accuracy increase (half a meter) with respect to pure trilateration with normal AP availability and the feasibility of accurately estimating 2D positions using only two APs. The obtained results demonstrate the feasibility of achieving accurate and robust location of WLAN terminals by means of a reasonably simple and flexible technique, which in addition does not require major hardware modifications over a standard WLAN infrastructure.

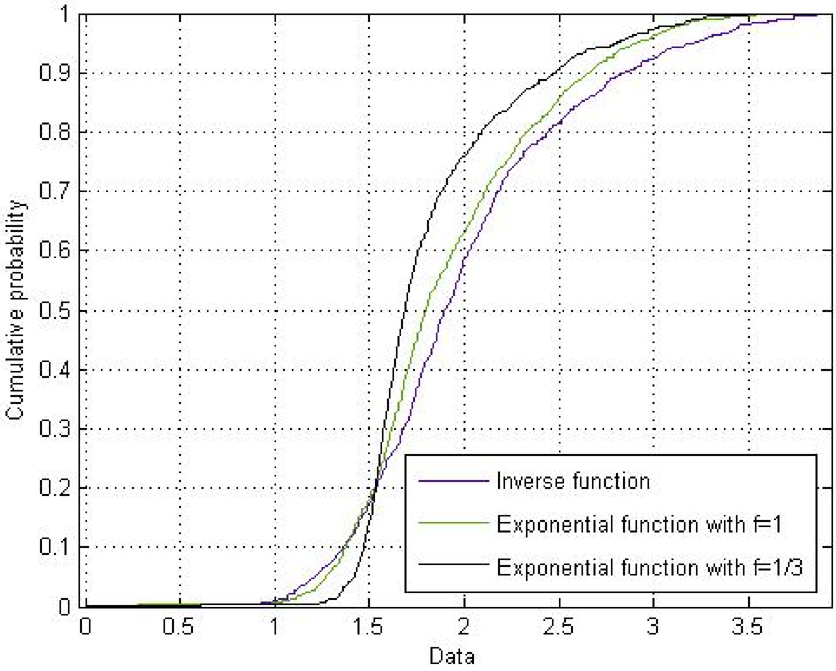

Figure 6.13: CDF of the absolute positioning error using some of the tested functions for situations with two reference points

Part IV

Studies on the location method

Chapter 7

Studies on the location method

7.1 Introduction

Once the different components of the TOA-based location technique (i.e. ranging and positioning) have been explored in Parts II and III and thus a complete location method can be proposed, I consider important facing the a priori major limitation of TOA-based location using wireless communications networks, which is its expected lack of scalability to large number of users. Therefore the first part of the current chapter corresponds to a study of the impact of this mentioned limitation on the proposed location method. After that, an approach to reduce the latency and improve the scalability is proposed in order to mitigate the noticed negative effect. On the other hand, this chapter also includes an evaluation of the new expected capabilities for TOA location of the upcoming IEEE 802.11v standard protocol, which can mitigate (among other issues) the expected limited scalability of TOA-based IEEE 802.11b/g location methods.

The scalability and the IEEE 802.11v capabilities are assessed on the location method that can be proposed from the results of the research presented in Parts II and III. More specifically, the method is composed by the hardware TOA-based technique to estimate the distance between WLAN nodes (Chapter 2) and the positioning module achieved in Chapter 6. Before starting with the first presented study and in order to understand the performed studies, I consider it necessary to summarize the complete process to estimate a position with this proposed method.

7.2 The positioning process

The process to calculate the MT's position employing the considered location technique can be summarized as follows:

1. The MT performs an active frequency scanning of all the IEEE 802.11 channels to select at least three available APs in range.

2. For each detected AP, the MT:

    (a) Associates and authenticates to the AP.

    (b) Performs the TOA-based ranging process to estimate the distance to the AP.

    (c) Disassociates from the AP.

3. The MT's coordinates are calculated by means of a trilateration or tracking algorithm using at least three distances as estimated in step 2 and the known coordinates of the APs.

The key step of TOA-based positioning using WLAN is the ranging process (i.e. step 2-b above) to estimate a distance with the TOA. In addition to the fact that ranging with WLAN entails the biggest challenges to reach good performance given the limited characteristics of the WLAN protocols, it has big and direct impact on the scalability and positioning latency of the achieved solution because it requires data communication between the WLAN nodes and a proper data post-processing.

The considered ranging method is the TOA-based one explained in Chapter 2. It mainly consists of performing RTT measurements by taking advantage of exchanges of IEEE 802.11 b/g MAC frames (Data-ACK) that can be induced in the MT. The transmission of each data frame to the AP is provoked by sending an ICMP Echo Request (i.e. Ping) from the MT to the AP. RTT measurements are performed synchronously: the transmitter (MT) waits for the reception of the ICMP Ping Echo Response from the AP before sending the next ICMP Ping Echo Request in order to assure that the received ACK corresponds to the previously sent data frame (as shown in Figure 7.1). At least one timeout has to be set up to avoid the blockage of the mechanism in case the MT does not receive the Echo Response from the AP after a certain amount of time, situation that is illustrated in the second RTT measurement of Figure 7.1 (the timeout expires and then the next data frame encapsulating an Echo request is sent). Please note that in this figure retransmissions are not depicted and that RTT would correspond to t_{TX_ACK} minus t_{TX_DATA}. Time is measured at hardware level adding a counter module to the WLAN card, employing the WLAN card clock at 44 MHz as time base.

In order to mitigate as much as possible the noise caused by the hostile indoor radio channel and the own RTT measurement system, 300 RTT measurements are performed to estimate a single RTT. Then these measurements are processed following a statistical method to accurately estimate the RTT (details can be found in Chapter 2). Once the RTT is estimated, the TOA and the distance between the MT and the AP can be easily calculated (again see Chapter 2 for details).

7.3 Study on scalability

Since TOA-based location techniques entail generating traffic to the network in each distance calculation process, frame collisions, busy medium detections and retransmissions occur when several MTs are simultaneously positioning using the same infrastructure (for details about the IEEE 802.11 medium access mechanism see [81]). These issues involve delays when a WLAN device accesses to the medium to transmit frames, thus that can degrade the positioning latency and then limit the number of MTs that can simultaneously calculate their position while maintaining a reasonable low latency. In addition, at the same time, since in such situations the achievement of necessary distances to calculate a position can be delayed, the positioning accuracy can also be degraded especially if the speed of the target is not low. In any case the result is that it can be expected that the performance of the indoor location method is degraded when the number of users increase. The goal of this study is assessing this performance degradation (taking the positioning latency as Quality of Service (QoS) parameter), in order to evaluate the scalability of the location system to large number of users. Since it is not feasible for me to build a large number of location prototypes, this assessment is carried out through simulations. To this end a simulator was build.

7.3.1 Description of the simulator

The simulator was built using OMNeT++ [82], a discrete event simulation system commonly used to simulate networks and protocol behaviour under multiple different situations, and the INET Framework package [83], which contains a wide range of libraries enabling us to use 802.11b components, in infrastructure or in ad-hoc mode. As the INET Framework is an open source code project, we modified its code in order to implement the parts needed for the TOA-based positioning process. The existing core of the package was not modified; hence timings, state machines, delays, packet messages and protocol layer behaviour remain unchanged. The IEEE 802.11b simulator runs in infrastructure mode. As shown in Figure 7.2, the simulator can be divided into three different parts: the environment, the IEEE 802.11b APs and the IEEE 802.11b MTs.

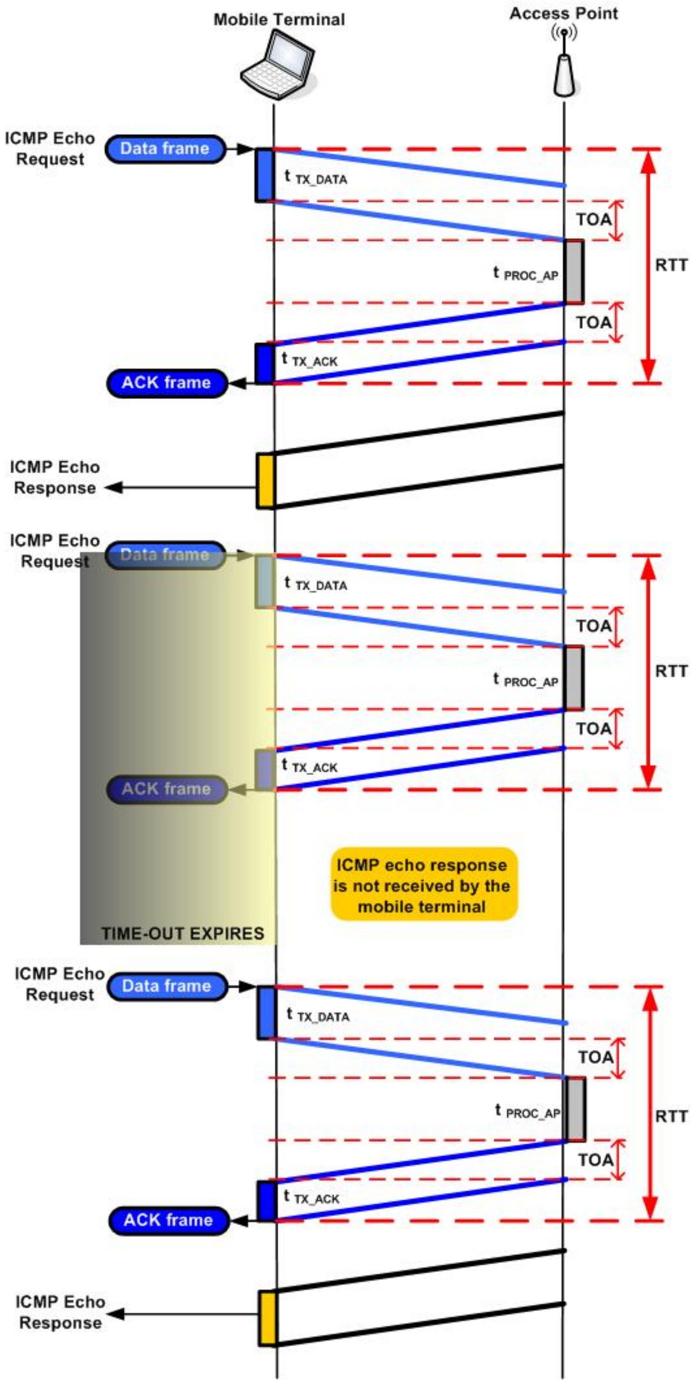

Figure 7.1: RTT measurements during the ranging process

The environment

The environment is always informed of the location and movement of the MTs, and determines which elements are within the communication range of the others using a propagation model, the frequency and the distance between them. The environment part is also in charge of assigning MAC and IP addresses to each element, reducing the complexity of the simulation by avoiding a dynamic set up

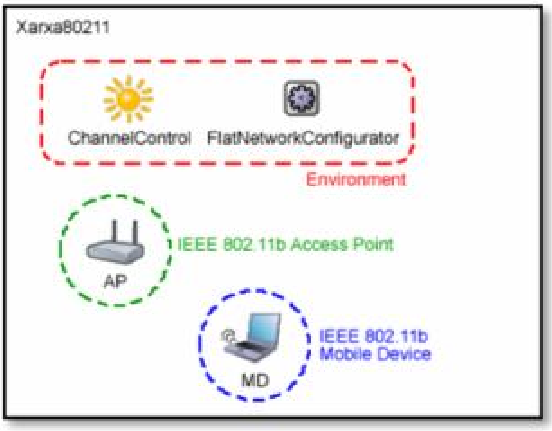

Figure 7.2: Parts of the TOA-WLAN location simulator

process of IP configuration. Furthermore, this part enables transmissions only if the involved MTs while AP are working under the same frequency or channel. In our simulations we use 4 radio channels, one for each AP. As this information is contained in a parameter list, when an MT performs an active scanning only scan these 4 channels instead of all 11 possible channels. This reduces the scanning process delay.

The wireless propagation model employed has a path loss parameter α=3 because of the indoor environment. This governs which devices are in range of each others, because it fixes the interference distance:

$$interference\_distance = \left(\frac{\lambda^2 \cdot P_{TX}}{4^2 \cdot \pi^2 \cdot P_{RX}} \right), \tag{7.1}$$

where PTX is the maximum transmission power, equal for both APs and MTs (set to 2mW) and PRX is the minimum power level to physically receive a signal, which has a value of -120dBm.

The APs and MTs

Regarding the APs and MTs, first of all we dealt with the problem of choosing which of these elements were going to be the passive ones, in other words: which elements start, control and oversee the positioning system and service. One option was to build a central system wired to the APs with the main task of discovering the position of the MTs, being the MTs the passive elements only responding the ICMP Ping echo requests frames sent by the APs. The other possibility was to build a distribute system in which MTs start, control and oversee the positioning service and APs -the passive elements in that case- only respond ICMP Ping echo requests frames sent by the MTs (and the ARP requests). It was decided to implement the latter option as it allows avoiding a complex and rambling programming, while preserving the functionality of the positioning process. Both APs and MTs have the same OSI structure until layer 3, but MTs have more functionality in layer 2: ability to send association packets and authentication packets, change the working frequency very easily with some command packets from upper layers, and work in infrastructure mode or in ad-hoc mode. Furthermore, there are upper layers in the MT case, because they are the active part of the system. Apart from this, there are some modules inside APs and MTs which do not correspond to an OSI structure, such as the interface table, which helps us to communicate all elements (see Figures 7.3 and 7.4)

In MTs and APs there is a Network Interface Card (NIC) which contains the standard IEEE 802.11b layer 1 and layer 2 modules (Radio, MAC and Management), as shown in Figure 7.5. Note

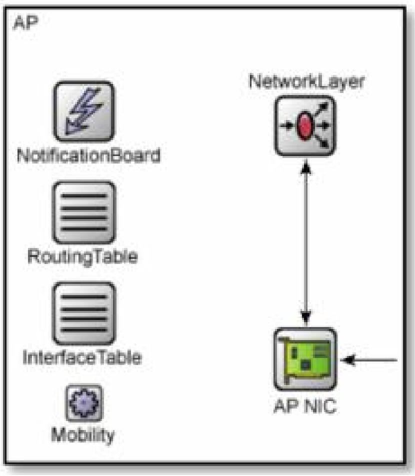

Figure 7.3: Internal modules of an AP

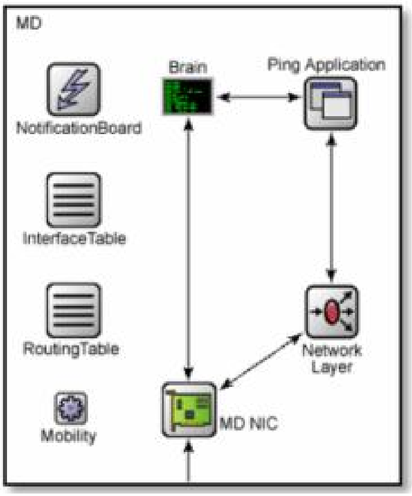

Figure 7.4: Internal modules of a MT

that in Figure 7.5 that there is an extra module only for MTs, the Agent, which will be detailed below. Immediately above the NIC module there is the Network layer, which is composed by several interconnected modules (IP, ARP and ICMP, as depicted in Figure 7.6) and shows a more automated behaviour.

Here finishes the common architecture for MTs and APs; however MTs are composed of additional modules (Ping Application and Brain) that allow implementing their commented proactive ranging functionality, as shown in Figure 7.7.

Above the Network module in MTs there is the Ping Application module. This module is very important as it is the one liable for managing the pseudo distance request command from the uppermost module by generating Ping Requests and calculating the packet loss. In addition, this module records statistics about the time spent in carrying out correctly the necessary number of Ping Requests involved in the pseudo distance estimate and also records statistics about the total amount of Ping Requests per pseudo distance request sent. These statistics are collected each time this application is called by the uppermost module and saved in an external file for post-processing. The configurable parameters of this module are: the packet size in bytes, the number of Ping Requests necessary for

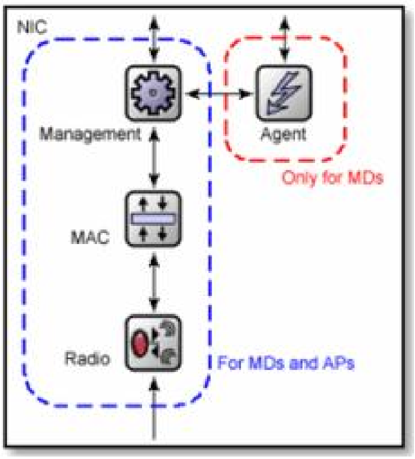

Figure 7.5: NIC of the APs and MTs

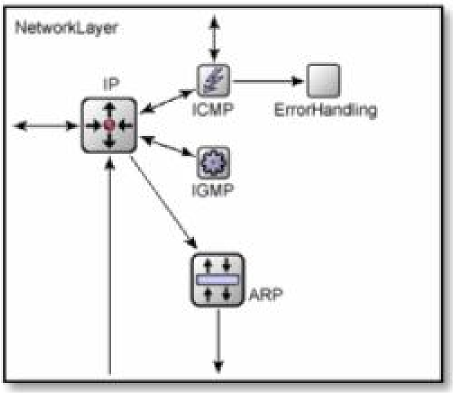

Figure 7.6: Network Layer of the APs and MTs

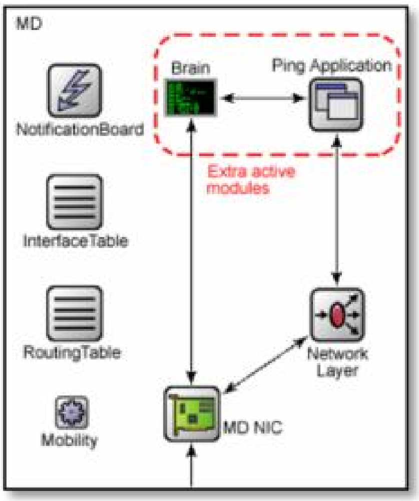

Figure 7.7: Extra active modules for MTs

each pseudo distance calculation, the timeout in seconds and the hop limit, which is the maximum number of jumps an ICMP Echo Request can do, not very useful in our simulation but implemented as well.

The uppermost module in MTs is the Brain module and as its name indicates it is the intelligent and the core module that manages and processes all the information received from packets, as well as it starts the positioning process. The MAC layer's Agent module is directly connected to this module, so that the Brain module can create commands, such as active scanning or association/disassociation requests, in order to discover APs in the communication range and connect to them.

The most important mentioned internal modules of the NIC and Network Layer modules are explained in the following paragraphs.

The Radio module The Radio module is the physical layer for the IEEE 802.11b models (both MTs and APs), and is basically made up by a state machine which controls transmissions and reception, a Signal to Noise Ratio (SNR) evaluator and a decider. The configurable parameters that govern this module are: the bitrate used to transmit or receive data in bits per second (bps), the transmitter power in milliwatts, the carrier working frequency in hertz, the thermal noise in dBms, the receiver sensitivity in dBms, the path loss coefficient, the Signal to Noise plus Interference Ratio (SNIR) threshold in dBs and in AP cases the channel number. All these values have to be set up in the simulator configuration file according to the ChannelControl values. APs and MTs use the same values for its radio modules: 11Mbps bit rate, transmit power of 2mW, SNIR threshold is set up in 4dB, sensitivity of -85dB and thermal noise of -120dBm.

When a frame is being received the module calculates the received power using the next expression:

$$P_{RX} = \frac{\lambda^2 \cdot P_{TX}}{4^2 \cdot \pi^2 \cdot distance\alpha}, \tag{7.2}$$

If the received power is greater than the sensitivity threshold and no other frame is colliding while receiving, then the frame is sent to the decider. Then, the decider checks if the SNIR of the frame is enough to recognize the data and if the data has any bit errors. In the favorable case the frame is sent to the upper layer (MAC layer). Otherwise, the frame is discarded. When the Radio module has to transmit a frame, the ChannelControl is notified and allows the Radio module to send only if the receiver Radio module is within communication range with the other device. Otherwise, that frame is discarded.

The MAC layer module The MAC layer module implements the IEEE 802.11b MAC protocol. Although it is a very realistic implementation, there are some features (like fragmentation, power management and polling) not supported, but the lack of these functions does not affect either this study or the LBS. When the MAC module receives a frame from the Radio module, it decapsulates the frame and analyses whether the packet is for that device or not, checking for this purpose the MAC final destination field of the received frame. In the affirmative case, the packet is sent to the first upper layer and in the other case it is discarded.

Another task of this layer is to set up automatically the MAC address of each device in a way that each device has a unique MAC address, notifying the FlatNetworkConfigurator about it and contributing with the shared and public IP/MAC list. Furthermore, like in other modules, there are some configurable parameters involved in this layer such as the queue size, the bit rate (that should be the same as in the Radio module), the retry limit (used when not receiving the corresponding acknowledgment or response) and the minimum contention window for data and broadcast packets (necessary for the back-off calculus). This layer functionality changes as these parameters are modified. MAC layer has also the same parameters for both APs and MTs, and the most important settable value is the sending packet retry limit, set up to 7 packets. The MAC frames corresponding to the ICMP Ping echo requests/replies are formed by 86 Bytes.

The Management module The Management module handles management frames and routes packets if the device is not the final addressee using a bridging mechanism. Otherwise, data packets are sent to network layer. This module is also in charge of controlling the association, disassociation, authentication and de-authentication processes, their respective timeouts and steps, the channel scanning in the MT case and the hosts list in the AP case. Furthermore, AP's Management modules set the SSID of each AP and the beacon interval. SSIDs must be introduced manually in the simulator configuration file, as well as the beacon interval in seconds and the number of associate/authenticate steps necessary to link a MT with an AP, being 2 for simulating non-encrypted links and 4 for WEP encrypted links.

The Agent module The Agent module, the extra internal module of the NIC module for the MTs, is attached to the Management module. This module is used only in 802.11b infrastructure mode and creates scanning, association and handover commands to send to the Management module. This module is governed and managed by another upper layer module.

The ICMP module The ICMP module is the Internet Control Message Protocol (ICMP) implementation. AP's ICMP module only receives ICMP Echo Requests, so this module creates the corresponding ICMP Echo Reply and returns it to IP module. In the MTs case there are 2 possibilities: receive an ICMP Echo Reply from IP module (in this case the packet's header is stripped and sent up to an upper module) or receive a Ping Request from an upper layer (forming the appropriate header and attaching it to the packet for IP delivering).

Implementation of the positioning process

Since the goal is studying the scalability of the location system, it is not necessary to implement and simulate all the steps of the explained positioning process, but only the steps in which interaction with other WiFi nodes (it is access to the shared medium) exist. For instance, it has no sense to include the time measuring mechanism for a RTT, the statistical processing of the RTT measurements, the algorithm to obtain the distance, and the trilateration or tracking algorithm.

The MTs core module starts (i.e. the Brain module) the process when a position has to be calculated. The MTs have access to a table with IP and MAC addresses of the APs located in the room. When the simulation starts, each MT waits for a delay and then starts an active scan in order to find which APs are in the range. Once the active scan finishes, for each detected AP the MT calls the scenario management module in order to calculate the power attenuation between them using a propagation model. After that, the list of detected APs is sent to the MT core module and it chooses 3 APs from the list using a power criterion. If there are less than the necessary 3 APs detected, it sends again a scan request and repeats this process until getting a list with at least 3 APs.

Once 3 APs in range are achieved, an association request is sent to the first AP and the management of this request falls in layer two. It starts the authentication process with the involved AP and straight afterwards the association process. Once the MT is connected with the AP, it immediately starts the ranging process to obtain the distance estimation. As explained in Section 2, this process mainly consists of synchronously sending 300 ICMP Ping echo requests to the AP in order to obtain the 300 RTTs. After MT core module sends an ICMP Ping echo request a timer starts counting. In the case MT receives the corresponding ACK and the ICMP Ping echo response, the timer stops and the pending RTTs counter decreases in one unit. In the case of several collisions, maximum number of retransmissions exceeded or any other kind of phenomena that can cause delay in the reception of the frame (e.g. large amount of frames in the AP's transmission queue) the timer can reach the timeout limit, then the MT core module starts the next RTT measurement (sends another ICMP Ping echo request to the AP). If the previous ACK is later received the MT core module discards it and does not compute the RTT.

Once 300 valid RTT measurements are collected, the distance can be estimated (not implemented in the simulator), MT switches off the link with AP according with 802.11b protocol (first disassociation and then deauthentication) and tries to connect with the second AP in the list in the same way mentioned before. When MT has obtained the three required pseudo distances, it can compute the own coordinates by means of a trilateration or tracking algorithm (not implemented in the simulator). MT then waits until the next positioning process, which starts immediately in the case of a tracking service activated.

7.3.2 Description of the simulations

Considered scenarios

The intention is to study the scalability of the system when operating in several representative LBS. Each considered service means a specific scenario in terms of simulation. The main difference between these services/scenarios is the elapsed time between positioning requests. For each scenario, simulations have been carried out for a range of number of MTs, in order to evaluate the possible degradation of the quality of service (positioning latency in our case) depending on the number of users. Given a specific scenario and a specific number of users in the area, 10 runs of the simulations are executed, in order to obtain reliable output data.

Scenario A In the first scenario (A), the time between position requests follows a uniform random variable from 0 to 10 seconds. The duration of each simulation run is 200 seconds. Simulations include situations from one to 30 MT in the area. This scenario models a positioning service in which pedestrians have to be located frequently but with no requirements of pure tracking. The traffic load generated in the network due to the positioning processes is expected to be heavy when the number of users is close to 30 taking into account that the room is not big.

Scenario B A second scenario (B) poses more stringent situations, because all the MTs are going to be continuously tracked with no time elapsed between the position calculations in each MT. This entails generating heavier traffic load than in scenario A. This kind of situations can occur in emergency services, in which firemen or policemen have to be always monitored from a remote control centre. It is expected that extreme network saturation will occur when having close to 30 MTs in the room.

Scenario C The last scenario (C) models pure positioning and the time between requests is governed by a negative exponential random variable (i.e. memory-less) with an average of 90 seconds. Each simulation run lasts 2400 seconds (40 minutes). This is representative of LBS such as search of lost people in events or department stores. As the positioning traffic in this scenario is lighter than the previous ones, we simulated it until reaching the amount of 50 MTs

Simulation area and trajectories for the MTs

The simulation area is a square room of 50x50 meters, with four APs placed in the corners, each of them at a distance of 15cm from the border of the room (i.e. coordinates in meters (0.15, 0.15), (0.15, 49.85), (49.85, 49.85) and (49.85, 0.15) respectively). Each MT starts its trajectory according to a uniform distribution in both axes of the room. The MT mobility model is basically the Mass Mobility of the INET Package of the OMNeT++ simulator, with some modifications that allow simulating a realistic pedestrian mobility model, as can be seen in Figure 7.8. The speed of the MTs is set up to 1 m/s.

7.3.3 Results of the simulations

Since the scalability is aimed to be evaluated the two main performance metrics that are studied are the mean time spent by a MT to estimate a pseudo distance and the mean time spent by the MT in the

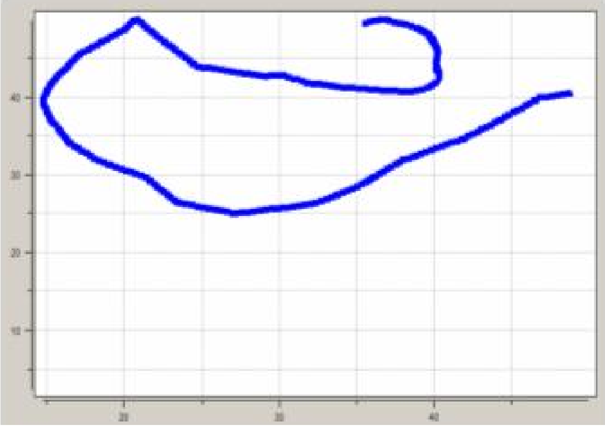

Figure 7.8: Trajectory of a MT generated with the motion model

association and disassociation processes with one AP. It has to be noticed that the mean positioning latency can be obtained as three times the time spent to get a pseudo distance plus the mean association and disassociation times. In addition, the mean number of ICMP Ping Echo Requests retransmissions and the mean number of distances estimations that are performed by a MT during the simulation time are also displayed in some cases, in order to endorse the main presented results and to better understand the behaviour of the positioning system.

Results are presented by means of graphs in which the x axis corresponds to the number of MTs using the positioning service in the room and the y axis to the value of the performance metric extracted. This way it is easy to observe the behaviour of the metric when increasing the number of MTs. As for a given scenario and fixed number of MTs the simulation is run 10 times, graphs draw the mean and the standard deviation values. When having few MTs a low standard deviation is expected as not so many collisions and retries are likely to occur. However, when reaching a certain number of MTs, standard deviation is expected to scatter around the mean because some MTs will have difficulty to carry out the RTT measurements and at the same time others may not.

Scenario A

In Figure 7.9 it can be appreciated a linear relation between the time spent to obtain a pseudo distance and the number of MTs in the system, with a low rhythm of growth in situations of less than 20 users. Numerically, in this interval of number of users the mean duration of distance estimation oscillates between 400 ms (only one user in the room) and one second, which mean that the positioning latency is not greater than 3 seconds; in fact it is possible to obtain a position in less than 1.5 seconds for situations of less than 7 MTs. However, in situations of more than 20 devices the slope increases, reaching positioning latencies longer than 10 seconds for 30 devices. This inflexion in the slope corresponding to 20 users in the system denotes a traffic saturation point, in which the system starts degrading performance faster as there are more collisions, retries and difficulty to access the medium.

This is corroborated with the graph in Figure 7.10, in which the tendency of ICMP Ping Echo Requests retransmissions as the number of MTs grows is depicted. Notice how from 20 MTs the channel is congested and the number of retransmissions cannot increase; the consequence is an unacceptable growth of the latency. In addition, in Figure 5 it can be appreciated the already explained behavior of standard deviation when increasing the numbers of MTs. This can be explained because with a large number of MTs we encounter situations in which some APs serve several MTs and other APs only a few of them in a certain period of time.

With more detail, the observed general tendency of the mean time spent in getting a pseudo distance (Figure 7.9) with respect to the number of MTs is mainly due to two facts. The first one

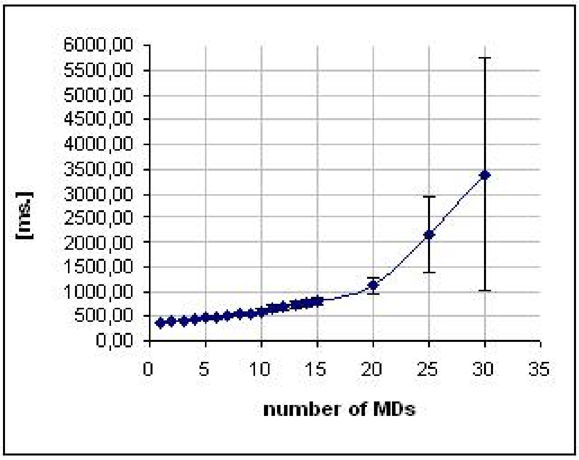

Figure 7.9: Time spent to obtain a pseudo-distance (Scenario A)

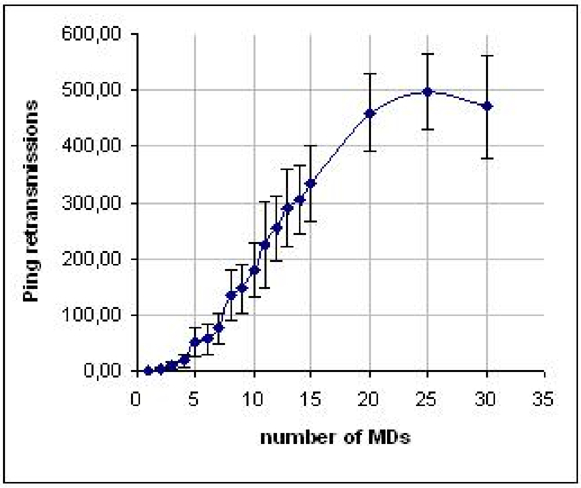

Figure 7.10: Echo Requests retransmissions (Scenario A)

is that MAC queue of APs receives in some cases ICMP Ping Echo Request from more than one MT at the same time, so there is a delay time until it can send the reply. The second fact involved in this degradation is the detection of channel activity following CSMA/CA medium access mechanism, with the consequent application of a back-off, a delay that multiplied by at least 300 times increases gradually the time needed to obtain a pseudo distance. The general tendencies observed are coherent with presented evaluations of the IEEE 802.11 CSMA/CA mechanism [84].

Figure 7.11 shows that, also as expected, the time spent in association and disassociation increases when increasing the number of MTs. It follows a logarithmic trend and the obtained values fit with empirical ones presented in some contributions (e.g. [85]). Despite the increasing delay in the latency also due to this process, it is negligible with respect to the delay we get in a pseudo distance measurement. Hence as expected it can be assumed that the positioning latency mainly corresponds to the sum of the processes of obtaining the pseudo distances.

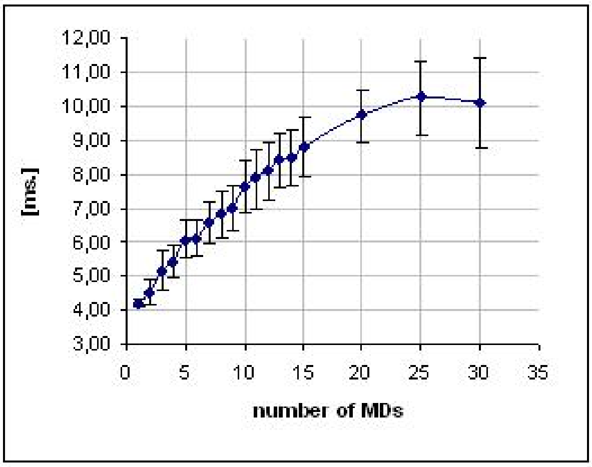

Figure 7.11: Time spent with association / disassociation process (Scenario A)

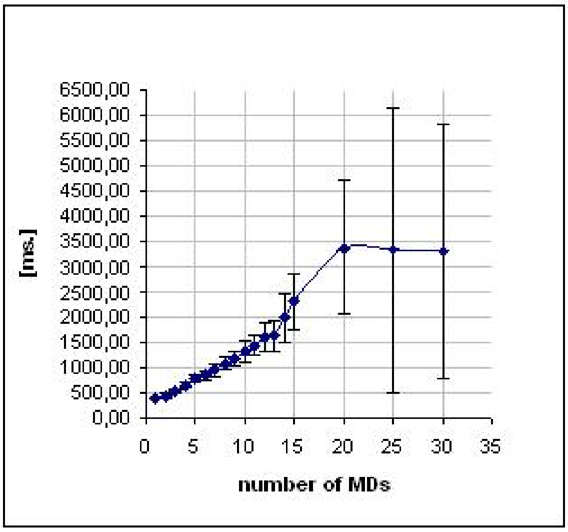

Figure 7.12: Time spent to obtain a pseudo-distance (Scenario B)

Scenario B

In Figure 7.12 we appreciate a similar linear tendency to the one obtained for scenario A, but now the inflexion point is reached for a lower number of MTs (13 in front of 20 MTs in A scenario). This earlier saturation was expected because the current scenario models a tracking service, in which the positioning requests rate is set to the maximum, in other words it is the most stringent service. Numerically, in this case we can obtain a position in approximately 1.5 seconds only if we have 3 or less MTs in the service. With more than 10 users, unacceptable latencies (for most of tracking-based applications) of more than 5 seconds are obtained. On the right part of Figure 7.12 it can be observed that the values decrease.

The system performance in this region can be better understood if we also take a look at Figure 7.13, which gives the amount of pseudo distances obtained by one MT during the simulation time as a function of the number of MTs: it can be seen that with more than 20 MTs the number of obtained pseudo distances drastically decreases, so that it can be thought that the time between distance esti-

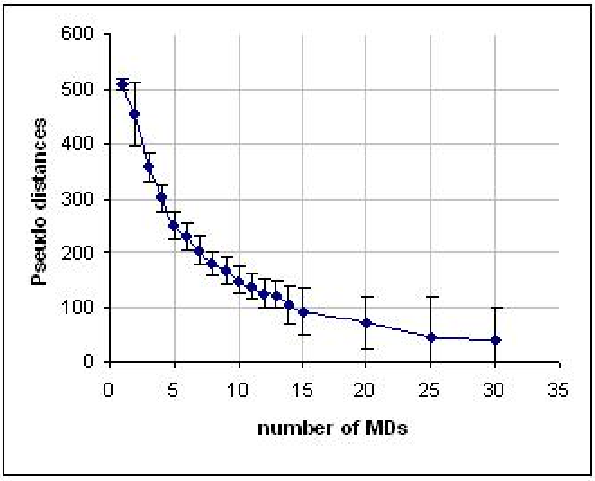

Figure 7.13: Number of pseudo distances per MT (Scenario B)

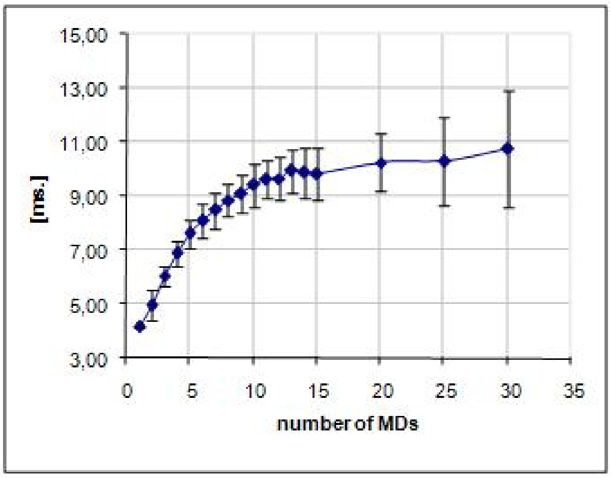

Figure 7.14: Time spent with association / disassociation process (Scenario B)

mations drastically increases, due to the impossibility to access to the medium in the first attempts of measuring the RTT. This behavior indicates a situation of total saturation of the system, state that was not reached in Scenario A in the same interval of number of users. On the other hand, observing Figure 7.14 (time spent for the switching between APs) it can be noticed a parallel behaviour with respect to the time spent for distance estimation: the values degrade faster than in Scenario A due to the more stringent traffic conditions.

Scenario C

As mentioned, the traffic conditions in this scenario are lighter than in the previous ones, therefore it is expected that the system can admit a higher number of users. Figure 7.15 clearly shows that now the linear tendency of latency as the number of MTs increases remains invariant in the studied interval (from 1 to 50 users in the system) and as expected the system is capable of assuming a larger number of MTs without degrading the latency. The key point is that the rhythm of latency increasing is noticeably lower than in previous scenarios. Numerically, the positioning latency is not bigger than

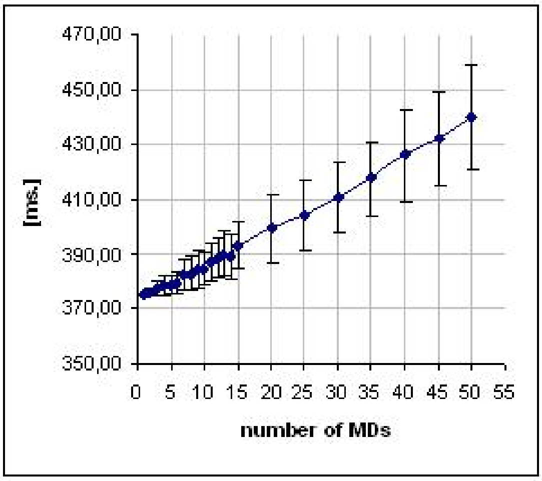

Figure 7.15: Time spent to obtain a pseudo-distance (Scenario B)

1.5 seconds even if 50 users are using the system in the room. In other words, this kind of positioning service is actually much more scalable than the previous ones. Other graphs are not depicted, but they corroborate the explained behavior of the system: the amount of packet retries performed by MTs and APs is smaller than 100 even with 50 users; all the values of duration of switching between APs belong to the [4ms., 6ms.] interval, noticeably smaller than in the other modelled services.

7.3.4 Conclusions of the study on scalability

A scalability study of the proposed TOA-based location system has been carried out by means of simulations, considering different types of LBS taking profit of the system. The main metric evaluated to analyze the scalability is the positioning latency, because is the main quality of service indicator that can be affected when increasing the number of users in the system. Obtained results allow to state that the TOA-based positioning method scales well when used in positioning services with long elapsed times (e.g. not lower than a few seconds) between positioning requests, because good positioning latencies of less than 1.5 seconds can be provided even in situations of 50 users in a room of 50x50 meters. However, in pure tracking services -which represent the most constrained scenarios since generated traffic in the network is maximum- only few users can simultaneously be tracked without degrading the quality of service under unacceptable values for most of applications. In addition, the system gets totally saturated if more than 20 users are accepted in the system for the specified scenario area. For pure positioning services (it is not tracking) with small times between positioning requests the system tolerates greater number of users without degrading latency but, as in the case of tracking, implementing a control admission mechanism would be necessary for most of applications. An important general observation after analyzing the results is that it is essential to take into account the strong impact of the service request rate on the quality of service in order to avoid reaching congestion.

7.4 Method to improve the performance of the TOA-based location method

The performed scalability study explained in Section 7.3 shows that positioning latency increases over unacceptable figures when more than few MTs are simultaneously tracked sharing the same WLAN infrastructure. Given that for some applications this observed phenomena can be a crucial limitation, now the objective is the study of strategies to overcome this negative effect. It has to be noticed that in most wireless networks pseudo-distances to the reference nodes must be estimated sequentially, and the localization latency is at least three times the time required to obtain a pseudo-distance. Note that this is not the case with GPS because in that system the receiver can simultaneously calculate the pseudo-distances to the GPS satellites by employing different signal reception channels. The first proposed strategies focus in reducing the tracking latency in TOA-based location techniques while keeping the accuracy degradation as low as possible. The idea is that if the time needed to obtain a pseudo-distance is reduced by decreasing the number of RTTs used to estimate it, the accuracy is degraded, but at the same time, the reduction in latency contributes to a better tracking and therefore a better accuracy. This important trade-off is analyzed during the design of the strategy. On the other hand, once these strategies to reduce the latency are proposed, it is possible to refine them in order to allow for better scalability. This improvement is also addressed during this research. This research is performed taking as basis again the same TOA-based location technique resulting from the ranging and positioning of Chapters 2 and 6 (for useful details affecting the current study see the positioning process that has been described in Section 7.2 of the current chapter 7).

7.4.1 Strategies to reduce the latency

In a TOA-based ranging method, the number of necessary RTTs performed for each pseudo-distance calculation is usually determined by the targeted TOA accuracy: the more RTTs averaged, the better the accuracy of the estimate. For instance, in the location method taken as basis for the current study, the RTT statistics showed that 300 RTT measurements were needed to estimate a single TOA with an error smaller than 6 nsec in 95% of the cases (see Chapter 2 for details). If the target is static, the targeted TOA accuracy is directly related to the targeted positioning accuracy through the trilateration algorithm employed to calculate the position (GDOP issues are out of the scope of this paper). Thus, in a static scenario, reducing the number of RTTs in order to improve the latency means worsening the accuracy of the target position estimate; an obvious trade-off between latency and accuracy exists.

If by the contrary the MT follows a motion pattern, the positioning error is compounded by two factors: the TOA estimation error (i.e. that explained for the static scenario) and the error caused by the mobility of the MT. The mobility of the MT affects the position estimation accuracy because the MT undergoes a displacement during the time elapsed between pseudo-distance calculations (the calculation of the pseudo-distances is sequential); hence, the trilateration or tracking algorithm that estimates the MT position takes pseudo-distances obtained at different physical locations of the MT. In this motion scenario, reducing the time needed to obtain a pseudo-distance, or reducing the number of RTTs, means reducing this second factor of the positioning error because then the MT displacement between pseudo-distances is smaller. However, at the same time, reducing the number of RTTs means worsening of the TOA estimation as in the static case. By reducing the number of RTTs to an adequate figure, mitigation of the second error component can compensate for the increment of the first one, so that the goal of reducing the positioning latency without degrading the positioning accuracy can become feasible.

In order to evaluate the described strategy, it is applied to the commented TOA-based localization technique. Since the original number of RTTs is 300, two approaches are considered in order to find the proper number of RTTs that can allow achievement of the desired effect: reducing the number of RTTs to 100 (strategy A) and to 30 (strategy B). Figure A depicts a simple scheme of the process needed to obtain a 2D MT position estimate for each strategy. The MT performs the fixed number of RTT measurements used to estimate the pseudo-distance to each one of the needed three reference points and then estimates its own position. The expected positioning latency reduction with respect to the original approach can be clearly observed.

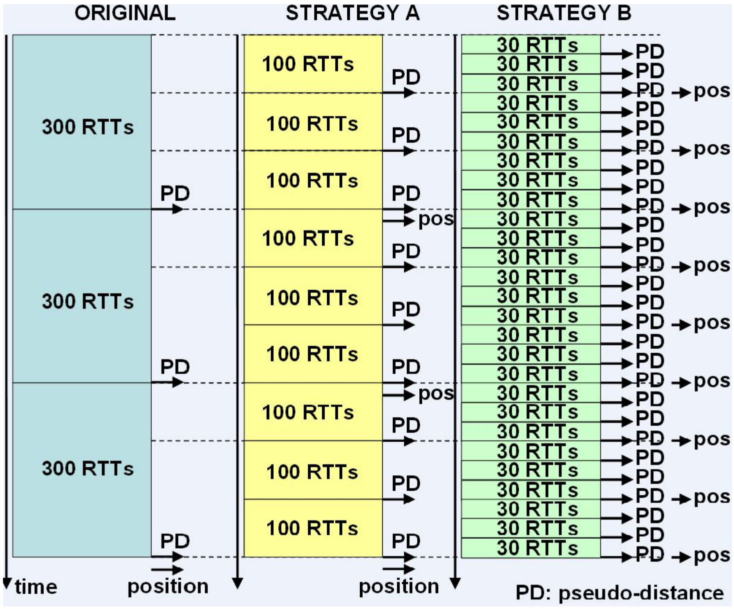

Figure 7.16: Schemes of the original process and the proposed A and B strategies used to reduce the latency

7.4.2 Evaluation of the strategies to reduce the latency

Simulation tool and setup

The validity of the described strategies is checked by evaluating the positioning accuracy provided when applied to the previously described TOA-based tracking technique over WLAN. To this end, simulations are carried out. The employed simulator is the one employed for the scalability study (Section 7.3), with the addition of some newly developed modules: a module implementing the EKF based tracking algorithm (according to the positioning/tracking technique specification given in Chapter 6) to calculate the position given a configured pseudo-distance statistical model, and a module that processes the position estimations to provide tracking accuracy statistics. Since the simulator employs WLAN as the communications technology, the results are especially valid when employing such wireless networks. However, the proposed strategies, ideas, and qualitative results can be also considered in the generic scope of TOA-based tracking methods.

As in the simulations performed during the scalability study, the simulated area is a square room with dimensions 50x50 m, with four APs placed in the corners, each of them at a distance of 15 cm from the border of the room (i.e., coordinates in m (0.15, 0.15), (0.15, 49.85), (49.85, 49.85), and (49.85, 0.15), respectively). The MT starts its random trajectory according to a uniform distribution on both axes of the room. The MT mobility model is again the Mass Mobility of the INET Package of the OMNeT++ simulator, with some modifications that allow simulation of a realistic pedestrian mobility model. The speed of the MT is set to 1 m/s. The MT position is continuously estimated by employing the TOA-based tracking technique with a specific strategy. Each strategy is evaluated for 30 different trajectories of the MT, and for each trajectory, 10 runs of the simulations are executed in order to obtain reliable output data. The duration of each simulation run is 200 seconds.

Statistical model of the pseudo-distances

Calculation of the pseudo-distances between an MT and a reference point is simulated according to distance estimation models obtained after real distance measurements using TOA from the hardware lab prototype (described in Chapter 3) to a commercial WLAN AP. In order to allow simulation of

Number of RTTs to estimate a pseudo-distance	300	100	30
Average of the pseudo-distance pdf (m.)	9.4	9.59	9.9
Standard deviation of the pseudo-distance pdf (m.)	1.41	2.68	3.34

Table 7.1: Average and standard deviation of the distance estimation pdfs obtained with 300, 100, and 30 RTTs at a real distance of 9 m.

Strategy	Original	A	B
Average absolute positioning error (m.)	1.46	1.63	1.42
Time to obtain a pseudo-distance (ms.)	379	129	46
Positioning latency (ms.)	1150	400	150

Table 7.2: Average absolute positioning errors and latencies achieved with the original and latency reduction strategies

the different strategies presented in this paper, statistical models of the pseudo-distance calculation employing 300, 100, and 30 RTT measurements were obtained. The first step was carrying out an extensive RTT measurements campaign in an indoor environment, considering different real distances between the prototype acting as MT and the AP. Then, for each real distance and for each different number of RTTs (300, 100, and 30), more than 500 pseudo-distances were estimated with the collected RTTs. Once achieved, this pseudo-distance database and the probability density function (pdf) of the distance estimation models for 300, 100, and 30 RTTs were obtained. These pdfs follow a Gaussian distribution; Table 7.1 shows the average and standard deviation of the obtained pdfs at a real distance of 9 m. As expected, the distance estimation accuracy decreases with the number of RTTs employed to perform the estimation (the standard deviation increases and the average becomes farther with respect to the real distance value).

Results

For each strategy, the average absolute positioning error is obtained, considering all of the trajectories and simulation runs. Table 7.2 summarizes the obtained results (the time needed to obtain a pseudo-distance and the positioning latency are also included).

The positioning latency is slightly longer than three times the time required to obtain a pseudo-distance, since some time is consumed to switch from one AP to the next. It can be appreciated that reducing the number of RTTs to 100 entails a slight degradation of the accuracy with respect to the original approach. This means that the reduction of the elapsed time between pseudo-distances and the consequent lessening of the positioning error component associated with the MT mobility is not enough to compensate for the worsening of the TOA estimate. However, if the number of RTTs needed to estimate a distance is reduced to 30, not only is the latency reduced more drastically, but in addition, the accuracy does not worsen with respect to the original approach.

7.4.3 Strategies to improve the scalability

The main idea of the strategy applied to improve the scalability of the technique to large number of MTs is to guarantee a more uniformly distributed occupancy of the medium between all MTs that are being tracked simultaneously. This can be achieved by properly reducing the time that each MT occupies the medium in order to allow other MTs to perform the RTT measurements during these idle periods. This entails reducing the number of RTTs to be performed for one pseudo-distance and then leaving an idle time until the theoretical time required to perform 300 RTTs is reached. After this idle time, the next RTT measurement series with the next reference point begins. This is shown in Figure 7.17, in which the number of RTTs has been reduced to 100 in strategy C. Since each MT starts being tracked at a random instant in time, with this approach, it is expected that on average all of the MTs

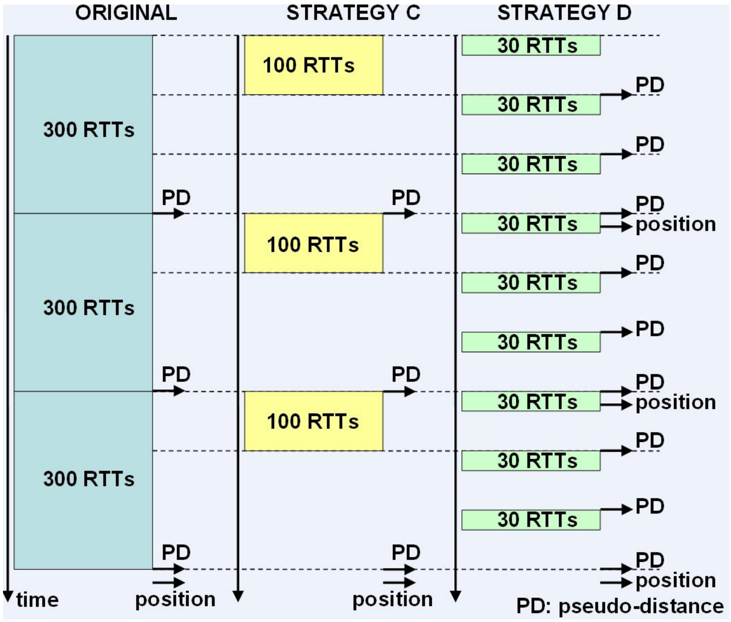

Figure 7.17: Schemes of the original process and the proposed C and D strategies for improvement of scalability

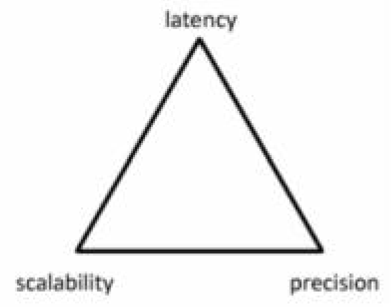

Figure 7.18: Trade-off in tracking service

can profit from the idle periods left by other MTs, so that the smallest possible number of terminals perform RTT measurements at the same time.

It must be pointed out that considering these kinds of strategies raises a trade-off situation between three tracking merit figures: scalability, accuracy, and latency (see Figure 7.18), so that varying one of them surely means altering at least one of the others. In strategy C, the positioning latency remains the same as in the original case (around 1.2 seconds), so that the trade-off only affects scalability and accuracy. Notice that although the number of RTTs is reduced with respect to the original approach, the elapsed time between pseudo-distances does not vary, and then the error component due to the MT mobility (explained before in Section 7.4.1) is not lessened. Hence, it is expected that the positioning accuracy degradation resulting from reducing the number of RTTs would be worse than for strategy A as presented in Section 7.4.1.

The idea behind strategy D in Figure 7.17 is improving the scalability to the same degree as with strategy C (the total amount of time that each MT occupies the shared medium does not vary) but taking advantage of the results obtained from the evaluation of the strategies to reduce the latency. Thus, the number of RTTs is reduced to 30 (with the corresponding idle periods) so that the elapsed

Strategy	Original	C	D
Average absolute positioning error (m.)	1.46	2.3	1.95

Table 7.3: Average absolute positioning errors achieved with the scalability improvement and original strategies

time between pseudo-distance calculations is reduced with respect to the original and C strategies in order to achieve less accuracy degradation than with strategy C. In addition, the approach reduces the positioning latency by a factor of three. In this case, the trade-off situation is the following: the scalability and latency are expected to improve with respect to the original approach, and then the price paid is a worsening of the accuracy. The expected positive outcome of this strategy is that it allows improvement in two performance figures with only slight accuracy degradation.

7.4.4 Evaluation of the strategies to improve the scalability

Evaluation of both presented strategies consists of quantification of the expected scalability improvement and the accuracy degradation that each strategy provides with respect to the original approach when applied to a TOA-based technique to track MTs. Results allow checking the validity of the given conjectures and assessing the validity of the proposed strategies. As in the scalability study presented in Section 7.3, the scalability is quantified by considering the positioning latency as a quality of service indicator. Simulations are carried out employing again the same explained simulator and set-up. The three tracking strategies are separately simulated in order to allow the performance evaluation. For each strategy, simulations have been carried out for a number of MTs from 1 to 60 in order to evaluate the scalability. Given a specific strategy and a specific number of users in the simulation area, 10 runs of the simulations are executed in order to obtain reliable output data.

Accuracy

The average absolute positioning error is calculated in order to assess the tracking accuracy (see Table 7.3). Obtained results corroborate the expected positive outcome of strategy D because its accuracy degradation is less than when strategy C is employed. Since strategy D lessens the latency with respect to C's reduction of the number of RTTs, this confirms the validity of the presented approach applied to reduce the latency while keeping the accuracy degradation as low as possible in dynamic scenarios and it also shows its applicability when the scalability is to be improved in addition to the positioning latency. Also, according to the results obtained when reducing the latency, the current obtained results are coherent with respect to the latter: strategies A and C fix the same number of RTTs, but A produces noticeably better accuracy because the elapsed time between pseudo-distances is kept smaller; the same comparative analysis can be applied to the results of strategies B and D. Finally, if strategies A and D are compared, one can see that although both allow the same time between pseudo-distances, as expected, the former performs better in terms of accuracy because the TOA estimation is more precise (the number of RTTs is bigger). In that case, again, a clear trade-off between accuracy and scalability exists.

Scalability

The performance metric studied to evaluate the scalability is the mean time spent by a MT to estimate a pseudo-distance for a certain number of MTs tracked in the simulation area because it is directly correlated with the positioning latency. Figure 7.19 depicts the value of this performance metric (y axis) depending on the number of MTs tracked in the simulation area (x axis) when the original tracking approach is employed. It can be seen that, in this case, an MT can obtain a position in less than approximately 1.5 seconds (500 ms to estimate a pseudo-distance) only if we have 3 or fewer MTs being tracked, and with less than 3 seconds only for fewer than 8 users. In addition, with more

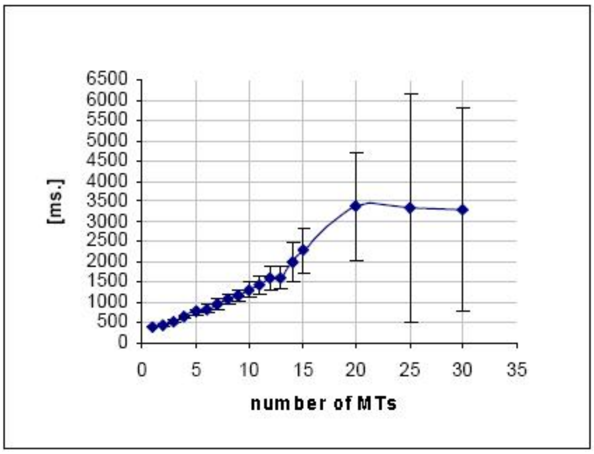

Figure 7.19: Time spent to obtain a pseudo-distance with the original tracking approach

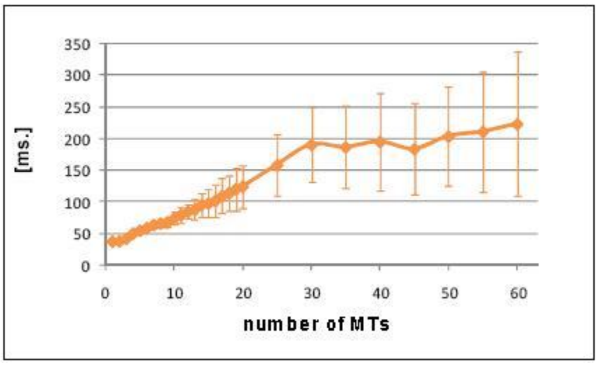

Figure 7.20: Time spent to obtain a pseudo-distance with strategy D

than 12 users, unacceptable latencies (for most of tracking-based applications) of more than 5 seconds are obtained. As concluded after the scalability study (Section 7.3), it can be stated that this approach does not scale well for a large number of users.

Obtained results for strategy C show that it allows the achievement of positioning latencies under 1.5 seconds for less than 25 users in the service area and below 3 seconds in any case for less than 60 users. These results demonstrate an important scalability improvement with respect to the original strategy. It can be also observed that, with strategy C, the MTs apply the idle time after the 100 RTTs only with fewer than 16 MTs because with a larger number of users, the time spent to perform the 100 RTTs (obtaining a pseudo-distance) exceeds the theoretical time that it takes to perform 300 RTTs. With more than 16 MTs, the scalability is also improved with respect to the original approach, but only because of the reduction in the number of performed RTTs. Figure 6 depicts the results for strategy D. Although the shared medium occupancy for each MT is the same as in strategy C, the latency reduction provided by D allows better results regarding scalability while considering latency as the quality of service parameter; positioning latency remains below one second in any case, even with 60 simultaneous users in the tracking service. Therefore, it can be stated that strategy D allows the tracking method to scale very well to a large number of users, providing a drastic improvement with respect to the original TOA-based tracking approach.

7.4.5 Conclusions of the studies to improve the performance of the TOA-based location method

Since TOA-based tracking techniques suffer from a lack of scalability mainly due to the big tracking latencies when the number of users increases, strategies to mitigate this problem have been proposed and evaluated. First, strategies to reduce the positioning latency without causing severe degradation of the accuracy have been analyzed. We have to be aware that most of TOA-based positioning and tracking methods propose performing a number of RTT measurements to estimate a single pseudo-distance, especially when operating indoors; in addition the use of wireless communications networks requires that the pseudo-distances to the required reference points are obtained sequentially. These two facts cause long positioning latencies, especially for tracking services. Simulation results presented show that this first group of proposed strategies can be applied to a TOA-based tracking technique over WLAN to reduce the number of RTTs from 300 to 30 (reducing the latency by a factor of 10) without worsening the positioning accuracy.

On the other hand, these achieved strategies can be refined in order to directly improve scalability. It has been shown how adequate strategies allow reduction of the scalability problem: it is possible to reduce the periods of time in which each MT injects traffic into the network in order to achieve a more uniformly distributed occupancy of the shared medium among all of the MTs. Simulations show how the scalability can be improved with only a slight accuracy degradation if a proper strategy that takes the MTs' mobility pattern into account is adopted. Although all the proposed strategies have been evaluated taking as tracking method the one proposed in the first chapters of this PhD thesis document, the results are valid in the scope of TOA-based tracking over wireless networks.

7.5 Study of IEEE 802.11v for TOA-based location

Since existing WLAN protocols (IEEE 802.11 b/g and others) were not devised for positioning purposes, it has been seen that despite the progress provided by research TOA-based location with current WLAN hardware and protocols can be considered still too immature for the commercial world. More specifically, implementing TOA-based systems with off-the-shelf hardware compliant with these protocols pose some limitations related to the fact of using standard IEEE 802.11:

Lack of scalability: As shown in Section 7.3, scalability to large number of users is poor in some situations. Since traffic is injected to the network in order to estimate the distances, frame collisions, busy medium detections and retransmissions occur when several MTs are simultaneously positioning using the same infrastructure. This involves delays that can degrade the positioning latency and limit the number of MTs that can simultaneously calculate their position while maintaining a reasonable latency.

Pre-calibration of the processing time at the AP: The needed manual pre-calibration is time consuming. In addition, since an accurate estimation of the data frame processing time at the AP (t_{PROC_AP}) is needed in order to properly estimate the TOA from the performed RTT measurements, variations of the processing time during the operation of the system can have a negative impact on the estimated TOA's accuracy.

Accuracy: Since IEEE 802.11b/g does not provide accurate time-stamps in the transmission and reception of frames, most of existing TOA-based ranging approaches propose the use of extra hardware added to the off-the-shelf WLAN hardware in the MT in order to obtain accurate RTT measurements.

However, the increasing interest in this kind of methods led to the inclusion of specific capabilities to ease the implementation and enhance the performance of TOA-based positioning in the upcoming IEEE 802.11v protocol standard [44], fact that could contribute to enhance the maturity of the promising TOA-based location technique. Given this situation, I considered interesting studying the impact of IEEE 802.11v on TOA-based location. More specifically, the goal of the current study is the analysis of the improvements of the TOA-based positioning technique performance that can be expected in practice with the upcoming IEEE 802.11v with respect to the use of the current WLAN protocols IEEE 802.11b/g. To this end, the performance of the proposed WLAN location approach based on TOA is studied through simulation with the two standards. First, it is implemented and simulated with IEEE 802.11 b/g as originally proposed. Afterwards the same approach with some modifications that take advantage of the upcoming capabilities of IEEE 802.11v is studied. A comparative performance evaluation between both is carried out in order to show the enhancements produced by the upcoming standard.

7.5.1 TOA-based location with IEEE 802.11v

New capabilities for TOA-based location

IEEE 802.11v includes an important number of new features [44], but in this study we focus on the ones affecting the TOA-based position calculation because this is our objective. Two main specific capabilities for TOA-based positioning can be found in the IEEE 802.11v standard draft [44]:

No need for association: Exchange of frames related with positioning (e.g. presence frames) between the MT and the AP without the need for being authenticated or associated to the AP. This allows the MT performing the RTT measurements to the required APs more easily and faster, hence reducing the positioning latencies, so that the scalability could be improved.

Measurement of the processing time: Measurement of the MAC processing time t_{PROC_AP} at the AP with a resolution up to tenths of nanosec. and inclusion of this figure as a field in the presence response in order to ease the TOA estimation in the MT. Thus the processing time is now calculated dynamically for each RTT measurement in real time so that the TOA estimation accuracy is not affected by variations of the t_{PROC_AP} value during the system operation, and on the other hand the manual pre-calibration is not necessary anymore.

Positioning process

The TOA-based positioning process described in Section 7.2, i.e. employing IEEE 802.11b/g, can be adapted to the IEEE 802.11v in order to gain advantage of the new features. Assuming that the MT to be located supports presence capability the modified process is detailed next. The MT (i.e. the client device in terms of the draft) sends a *Presence Request* frame (this is a new frame type defined by the upcoming standard) to an AP indicating its purpose to perform an RTT measurement and starts counting the time (RTT) until the corresponding ACK frame from the AP is received. The AP sends to the MT -just after sending the mentioned ACK- a *Presence Response* (another new frame type) including a *timestamp difference* field (i.e. the t_{PROC_AP}) in the presence parameters. This procedure has to be repeated for at least three APs in order to obtain the minimum needed distance estimations for trilaterating to calculate the MT's position. The entire process can be implemented in the MT as follows:

1. The MT identifies three channels belonging to three APs to perform the measurements by means of the Presence Parameters information element in the Beacon, Probe Response or Presence Configuration Request frame.

2. For each one of the three detected APs the MT must:

   (a) Tune the AP channel frequency.

   (b) Perform the TOA-based ranging process to estimate the distance to the AP.

3. Estimate its coordinates by means of a trilateration or tracking algorithm using the three calculated distances and the known coordinates of the APs.

It is important to observe that now the MT switches from one AP to another by means of a simple frequency channel change, without associating and authenticating with the AP, so that the delay of AP switching is expected to be smaller. It has to be pointed out that unfortunately IEEE 802.11v does not provide any new mechanism to enhance the accuracy of RTT measurements as a requirement. The lack of accurate of accurate time-stamps is kept as in IEEE 802.11b/g.

7.5.2 Simulation environment

Extensive simulations of both positioning approaches have been performed in order to comparatively evaluate their performance. The employed WLAN location simulator is the same explained in Section 7.3, built using OMNeT++ [82] including the INET Framework package for the IEEE 802.11 components. Extra components presented in Section 7.3 that implement the TOA-based positioning process are used together with new implementations to integrate the TOA capabilities of IEEE 802.11v. The number of RTT used to estimate a distance is set to 300 for both tested location approaches. The simulations will allow assessing the improvement that can be expected form the two explained capabilities of IEEE 802.11v for TOA-based positioning. Each capability is treated separately.

No need for association

The considered figures of merit to evaluate this new capability are the positioning latency and the system scalability, because as explained before this capability of IEEE 802.11v could smooth the problem of the lack of scalability of IEEE 802.11b/g. Several representative LBS, taken as different scenarios, are considered for simulation. For each scenario, simulations have been carried out for a wide range of number of MTs in order to evaluate the positioning latency depending on the number of users. Given a specific scenario and a specific number of users in the area, ten runs of the simulation are executed in order to obtain reliable output data.

The simulated area is a square room of 50x50 meters. The simulator runs in infrastructure mode, with four APs placed at the corners. This physical environment has been chosen so that it is simple enough to avoid extra variables not under study such as Non Line of Sight (NLOS) or 3D positioning. Each MT starts its trajectory according to a uniform distribution in both axes of the room. The mobility model of a MT is basically the Mass Mobility of the INET Package of the OMNeT++ simulator, with some modifications that allow simulating a realistic pedestrian mobility model. The average speed of the MTs is set to 1 m/s.

In the first scenario (A), the time between position requests is modeled by a negative exponential distributed random variable (i.e. memoryless, the request arrival process is Poisson) with an average of 90 seconds. The duration of each simulation run is 2400 seconds. Simulations include situations from one to fifty MTs in the area. This scenario models a positioning service in which pedestrians do not have to be located frequently (e.g. search of lost people in events or department stores). The traffic load generated in the network due to the positioning processes is not expected to be heavy. The second scenario (B) poses more stringent conditions, because all MTs are going to be continuously tracked with no time elapsed between the position calculations in each MT. This entails generating heavier traffic load than in scenario A. This kind of situations is likely to occur in emergency services, in which firemen or policemen have to be always monitored from a remote control centre. Each simulation run lasts 200 seconds and the maximum number of MTs is set to 30.

The considered positioning services through these scenarios (light positioning and tracking) cover the most representative traffic conditions that directly affect the scalability of the service. This way we guarantee that the comparative evaluation between 802.11b/g and 802.11v based location is fair.

Measurement of the processing time

The considered figure of merit is the ranging error caused by the use of a pre-calibrated value of t_{PROC_AP} when employing IEEE 802.11b/g network infrastructure. Thus first the manual pre-calibration to obtain t_{PROC_AP} is simulated and the obtained t_{PROC_AP} value is stored. Then the ranging process between a MT and an AP employing the before obtained t_{PROC_AP} value is simulated and the ranging error measured, inducing at the same time as much as possible variation of t_{PROC_AP} at the AP. Since t_{PROC_AP} could increase when the traffic load on the AP increases (due to the major simultaneity of received frames in the AP), the way to induce the variation in the simulation is adding MTs performing ranging (i.e. transmitting data) to the AP. IEEE 802.11v is not simulated because as explained before its capability of measuring the processing time at the AP in real-time guarantees no accuracy degradation due to t_{PROC_AP} fluctuations.

Simulations are carried out for a wide range of number of MTs in order to evaluate the ranging accuracy depending on the traffic load on the AP. For each number of users in the area, ten runs of the simulation are executed in order to obtain reliable output data. It is important to leave clear that for these simulations the simulator is configured for not taking into account the ranging error due to the radio channel and the RTT measurement mechanism, so that the phenomena aimed to be studied can be isolated. The simulated area is again a square room of 50x50 meters but now only one AP is placed in the room, because the study focuses on the ranging process to this AP.

7.5.3 Performance evaluation

No need for association

Some specific metrics that allow evaluating the positioning latency and the system scalability have been analyzed in the simulations: mean time spent by the MT to obtain a distance estimation (which can be easily translated into positioning latency), mean time spent by the MT in the association and disassociation processes with one AP (for IEEE 802.11b/g), mean number of distance estimations that are performed by a MT during the simulation time, and MAC frame retransmissions per MT. Results are presented by means of graphs in which the x axis corresponds to the number of MTs using the positioning service in the room and the y axis to the performance metric. This way can be easily observed the behavior of the metric when increasing the number of MTs. Since for a given scenario and fixed number of MTs the simulation is run 10 times, graphs draw the average values. In all cases the standard deviation along the 10 runs is small enough to guarantee the statistical reliability of the obtained averages.

Scenario A: light positioning service Figure 7.21 shows the time spent to obtain a pseudo distance following both approaches when the number of MTs increases. In both cases it can be appreciated a linear behavior between this time and the number of MTs, even with the same slope. The only appreciable difference between them is that using IEEE 802.11v a slight offset of 25 ms exists. The explanation is that in our implementation the time between consecutive RTT measurements is bigger for the 802.11v approach, due to the extra delay of the presence response reception event handling from MAC layer to application level (for 802.11b the ICMP Echo Request reception can be more directly handled at application level, avoiding this delay). This time difference between both approaches depends on the implementation of the ranging process at application level and is not caused by factors related with the IEEE 802.11 protocol. This fact, added to the small value of the offset, makes it not relevant for our study.

The relevant issue to be observed is the growing rhythm of the time spent to obtain a pseudo-distance with the number of MTs (i.e. the slope of the represented functions). The time increment basically corresponds to the increment of the channel access delay that suffers each MT, which is determined by the backoff procedures of the IEEE 802.11 Distributed Coordination Function (DCF) to access the medium. When the number of simultaneous MTs trying to gain the shared medium to transmit increases, the probability of frame collisions (with the consequent retransmissions) and busy medium detections gets higher, fact that directly impacts on bigger backoff times. Since in scenario A the traffic that each MT injects to the network is low, the channel access delay hardly increases with the number of MTs and therefore the time to obtain a pseudo-distance does not noticeably vary: it starts around 400 ms for one single MT and remains below 500 ms in both approaches even though

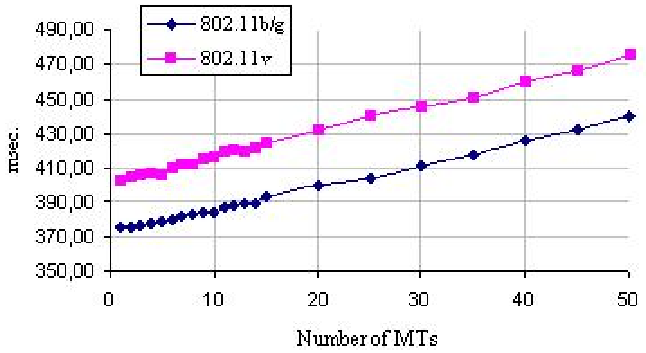

Figure 7.21: Time spent to obtain a pseudo distance in scenario A

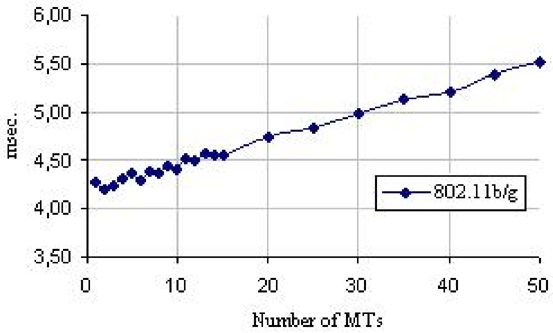

Figure 7.22: Time spent with the association/disassociation process in scenario A (IEEE 802.11b/g)

the number of MTs grows until 50.

On the other hand, Figure 7.22 indicates that the additional time spent in the IEEE 802.11b/g approach with the association-disassociation process to the APs can be considered negligible in terms of positioning latency. The most important point is that obtained figures indicate that both systems can scale well with a positioning latency of less than 1.5 seconds, therefore it can be stated that in terms of latency and scalability both approaches behave very similar with light positioning traffic.

Scenario B: tracking service In Figure 7.23 can be seen that, as expected, in scenario B the time spent to obtain a pseudo distance increases much more fast than in scenario A, because MTs are continuously tracked with no time elapsed between the position calculations in each MT. Although this means a heavy traffic load in the network, it is still far from reaching the saturation traffic condition (situation in which each MT always has frames waiting for transmission). This condition is not reached because, as explained before, each MT waits for the finalization of a single RTT measurement before transmitting the frame corresponding to the next measurement.

Observing Figure 7.23, the time that takes performing 300 RTT measurements increases in almost 2 seconds when the number of MTs is increased from one to 20 using 802.11v, which means an increment of the channel access delay for each frame transmission of more than 3 msec. (each measurement entails two frame transmissions, one from the MT and another from the AP). With 802.11b/g the increment of the channel access delay for each frame transmission is of around 5 msec., so that the pseudo distance latency worsens faster than with 802.11v. This different behavior can be due to the bigger length of the ICMP Echo frames with respect to the Presence ones; because as shown in [86] bigger frame lengths entail more frame collisions and therefore an increment of the channel access

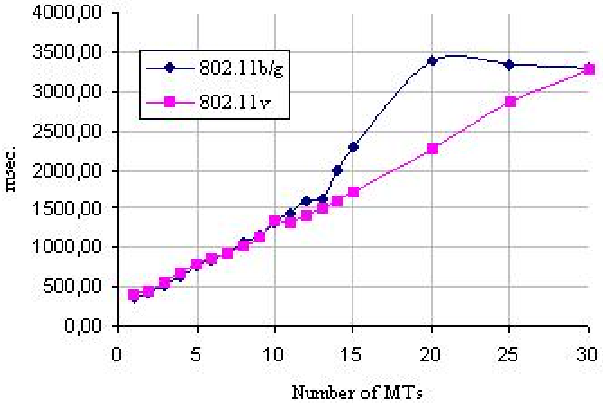

Figure 7.23: Time spent to obtain a pseudo distance in scenario B

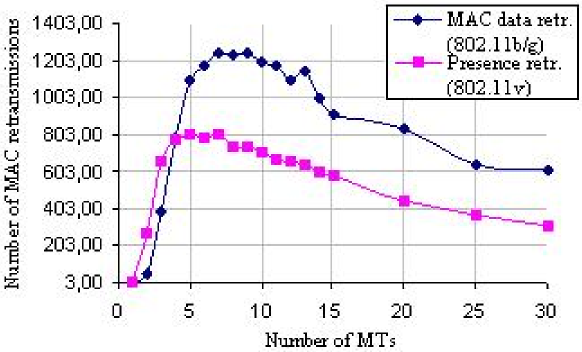

Figure 7.24: MAC retransmissions per MT in scenario B

delay. Recently in [86] the average channel access delay for a network in saturation traffic condition has been accurately modeled; presented results show an average access delay between 10 and 15 msec. for transmissions of frames of similar size to the ICMP Echo ones. Since as explained above the simulated tracking service does not make the network reach this saturation, it is coherent that in our simulations the obtained average channel access delays remain always below this mentioned upper limit value.

The explained worse behavior of the 802.11b/g approach due to the bigger number of frame collisions is corroborated observing Figure 7.24, which illustrates that the number of MAC data frames retransmissions is much bigger than the number of Presence frames retransmissions. Figure 7.23 shows that in practice for a range of 15-30 MTs the time to obtain a distance would be up to more than one second longer for 802.11b/g than for 802.11v, which means noticeably poorer positioning latency of the former approach (at least 3 more seconds with respect to 802.11v since three distances are needed). This is corroborated by Figure 7.25 because for more than 15 MTs in the service the number of estimated distances per MT is bigger using 802.11v. Results obtained for more than 30 MTs have not been included in the graphs because they contain large standard deviations and therefore they do not have statistical relevance for our study. On the other hand, Figure 7.26 illustrates that, as for the previous scenario, the time for switching between APs is negligible and does not affect the latency and scalability.

After these observations, it can be stated that 802.11v scales better (for certain ranges of number of users) than 802.11b/g when continuous tracking is performed, but in both cases a control access

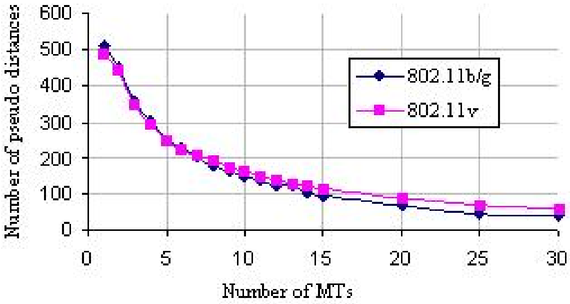

Figure 7.25: Number of obtained pseudo distances per MT in scenario B

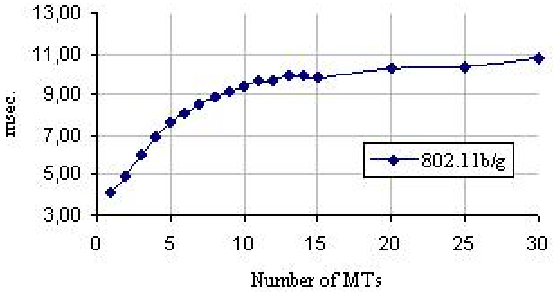

Figure 7.26: Time spent with the association/disassociation process in scenario B (IEEE 802.11b/g)

mechanism would be needed because unacceptable positioning latencies for most of applications are finally reached for more than 15-20 MTs. However IEEE 802.11v provides the important advantage of avoiding the authentication and the network layer traffic to the AP, fact that allows higher accessibility to existing network infrastructures.

Measurement of the processing time

Results are presented by means of a graph (Figure 7.27), in which the x axis corresponds to the number of MTs performing ranging with the AP (i.e. generating traffic to the AP) and the y axis to the ranging error due to the use of the pre-calibrated t_{PROC_AP}. This way it is easy to observe the behavior of the analyzed metric when increasing the traffic to the AP. Since for a given scenario and fixed number of MTs the simulation is run 10 times, graphs draw the average values.

Figure 7.27 shows that, despite the error increases with the traffic as expected, it always remains negligible (below the micrometer). This means that t_{PROC_AP} it is hardly affected by the volume of traffic handled in the AP and therefore the pre-calibrated value of t_{PROC_AP} can be considered valid for accurate ranging during all the positioning system's operation. This observed phenomenon could be explained because when a MT gains the access to the medium for transmitting to the AP does not allow the other MTs transmitting their data (they suffer frame collisions and busy medium detections), so that the traffic handled by the AP at a instant of time can not hardly increase.

These obtained results mean that this new capability of IEEE 802.11v does not contribute to enhance the robustness in terms of ranging accuracy with respect to IEEE 802.11b/g. Despite this fact, the capability is important because it avoids the manual pre-calibration process.

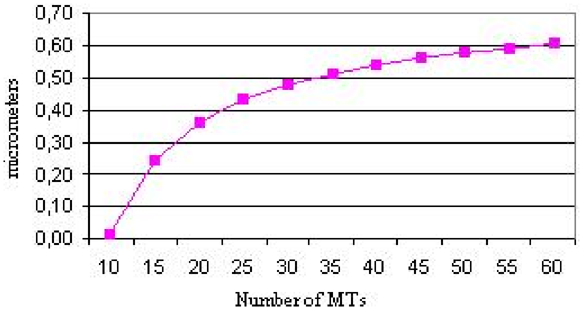

Figure 7.27: Ranging error due to the pre-calibration

7.5.4 Conclusions of the study on IEEE 802.11v for TOA-based location

This study has explored the benefits that the upcoming IEEE 802.11v standard can provide to soften the current limitations of the promising TOA-based positioning technique. To this end the performance of a commonly adopted TOA-based positioning approach that uses RTT measurements with IEEE 802.11b/g and the same approach incorporating the IEEE 802.11v capabilities has been comparatively evaluated.

Results have shown that IEEE 802.11v can provide a slight improvement of the scalability to large number of users in the case of stringent tracking services, but in a minor degree than expected with the new "no need for association" explained capability. Therefore a control access mechanism would be still needed as with IEEE 802.11b/g. However, an important advantage of 802.11v is that authentication to the APs is avoided. On the other hand, the real-time calculation of the processing time in the AP that will include IEEE 802.11v does not provide an increment of ranging accuracy but avoids the manual pre-calibration now needed in TOA-based techniques that use RTTs. In our opinion the major disappointment regarding IEEE 802.11v is not providing any mechanism to ease reaching accurate RTT measurements with off-the-shelf WLAN hardware, which is one of the main limitations of TOA with WLAN. However the overall conclusion is that IEEE 802.11v can provide important benefits to TOA-based positioning in the next future.

Part V

Conclusions

Conclusions

Research work on TOA-based location methods over WLAN has been reported in this PhD Thesis document. An accurate study of the state-of-the-art about positioning with WLAN has lead to know the limitations of existing methods and in addition has allowed identifying TOA as the most promising alternative to go a step forward overcoming these limitations. TOA-based location works in two stages: ranging and positioning. The ranging consists of estimating the distances between the targeted terminal and several APs, each distance obtained measuring the TOA and then multiplying it by the speed of the WLAN signal. After that, the positioning takes as inputs the estimated distances and the known coordinates of the involved APs and calculates the position of the terminal by means of a trilateration or tracking algorithm.

The ranging is the most challenging part because accurate TOA measurements between WLAN nodes are difficult to achieve. Two different ranging approaches have been explored. Both approaches share the same basic design principles, which are performing RTT measurements at the mobile device using standard data/control frames at the IEEE 802.11 MAC layer to estimate the TOA between transmitter (the mobile device) and receiver (the AP), but completely differ in their implementation. In the first method a slight hardware enhancement of the client's WLAN interface is required, because the transmissions and receptions of the MAC frames are time-stamped at hardware level using available signals (the available clock signal among them) in the chipset of the client's WLAN interface. A lab prototype has been implemented, extensive RTT measurements campaigns indoors have been carried out, and the statistical properties of the measurements have been analyzed. It has been seen that the measurements are not deterministic but noticeably contaminated by different sources of noise related with the own measurement system that provoke time dispersion and a positive error bias. RTT is then dealt as a random variable and estimated by means of a proper simple statistical processing that palliates as much as possible the existing noise. The achieved ranging accuracy is around one meter of error; this fact proves that direct time stamping of IEEE 802.11 standard frames using the available clock in the WLAN card would suffice to achieve accurate ranging capabilities.

The second ranging method is conceived to avoid any hardware manipulation over WLAN standard consumer equipment with the challenge of keeping the good ranging performance of the first method. The approach mainly consists of time-stamping the MAC events at OS level and measuring the RTT with the clock signal of the mobile device's CPU. The driver of the WLAN client's interface is modified and the OS configured to minimize the effect of the WLAN interrupt latency at the OS. A lab prototype has been implemented and extensive RTT measurements campaigns have been carried out with different OS configurations. As expected, the time dispersion and inaccuracy of the RTT measurements is noticeably bigger with respect to the hardware-based method, mainly due to the interrupt latency and the impact of the OS mechanisms on the measurements. For that reason the adopted statistical processing of the RTT samples is more stringent and the number of needed samples is bigger. Distances estimates indoors show good stability and average errors around 2 meters, a good result that allows demonstrating the feasibility of achieving accurate ranging by means of a pure-software solution using WLAN standard equipment.

The work on the ranging part is closed with a method to mitigate the effect of the multi-path and obstructed path problem on the researched TOA-based ranging techniques. First it is proved that this phenomenon has a negative impact on the ranging accuracy, especially when severe obstructions exist between the WLAN nodes. Then an approach to detect and overcome this problem is presented. This method identifies in real time the degree of obstruction and accordingly applies a multipath-sensitive ranging algorithm. The method has been tested with the hardware-based ranging technique and it has been demonstrated that it provides robustness enough to preserve the good ranging accuracy regardless the situation.

Regarding the positioning, first some trilateration algorithms are assessed as the simplest way to calculate the target's position once the ranging is completed. Positioning errors smaller than 2m. for the 90$\%$ of the cases are obtained. Then those results are taken as start point to research tracking algorithms in order to improve the achieved positioning performance. In this way, Kalman filtering has been taken as the basis for obtaining an algorithm that provides good accuracy when at least three APs are in sight of the MT. In addition, an algorithm designed for situations with only two APs has been proposed, which solves the ambiguity of raw trilateration by applying tracking principles that can utilize past trajectory information. Both algorithms have been optimized and tested using simulations fed with actual ranging data from the hardware TOA-based lab prototype. The results show a substantial accuracy increase (half a meter) with respect to pure trilateration with normal AP availability and the feasibility of accurately estimating 2D positions using only two APs.

The obtained results about ranging and positioning demonstrate the feasibility of achieving accurate and robust location of mobile terminals by means of a technique that can be easily and rapidly deployed over standard WLAN infrastructures. This technique does not require pre-calibration and totally adapts to changes in the environment.

Since the scalability to large number of users can be a weakness of TOA-based methods due to the traffic generated in the ranging process, a study to assess the scalability of the proposed TOA-based location methods has been performed. The analysis of the simulations results has shown that the scalability is good when the time between position calculations is considerably long, but at the contrary it is bad for tracking services. Given this, some strategies to improve the scalability without degrading the positioning accuracy in tracking services are then researched. The adopted approach mainly consists of reducing the number of RTT measurements to estimate a single distance, leaving idle periods in which the devices do not generate traffic to the network; this way a more uniformly distributed occupancy of the shared medium among all of the mobile devices is achieved. The idea behind takes advantage of the mobility of the devices: the reduction in latency contributes to a better tracking accuracy and then compensates the degradation of ranging accuracy caused by the reduction of the number of RTTs. Simulations with real ranging data shows the important achieved scalability improvement with only a slight accuracy degradation.

Part VI

Appendices

Appendix A

Acronyms in the text

GPS	Global Positioning System
GIS	Geographical Information Systems
LBS	Location-Based Services
GSM	Global System for Mobile Communications
GPRS	General Packet Radio Service
UMTS	Universal Mobile Telecommunications System
AHLoS	Ad Hoc Location System
WLAN	Wireless Local Area Networks
UWB	Ultra Wide Band
WPAN	Wireless Personal Area Networks
RSSI	Received Signal Strength Indicator
AOA	Angle Of Arrival
3GPP	3rd Generation Partnership Project
AP	Access Point
RADIUS	Remote Authentication Dial-In User Server/Service
SNMP	Simple Network Management Protocol
WiFi	Wireless Fidelity
API	Application Programming Interface
RFID	Radio Frequency Identification
EKF	Extended Kalman Filter
LOS	Line Of Sight
PLMN	Public Land Mobile Network
TOA	Time Of Arrival

TDOA Time Difference Of Arrival

DTDOA Differential Time Difference of Arrival

VOR VHF Omnidirectional Ranging

NLOS Non Line Of Sight

MIMO Multiple-Input Multiple-Output

RTLS Real Time Location System

PDA Personal Digital Assistant

HTTP Hyperlink Transport Protocol

XML eXtensible Markup Language

SDK Software Development Kit

NMEA Nation Marine Electronics Association

SUPL Secure User Plane Location

SOAP Simple Object Access Protocol

DLOS Direct Line Of Sight

RTT Round Trip Time

RV Random Variable

ESPRIT Estimation of Signal Parameters via Rotational Invariance Techniques

MUSIC Multiple Signal Classification

RTS Ready To Send

CTS Clear To Send

DCF Distributed Coordination Function

FEP Frame Exchange Protocol

NAV Network Allocation Vector

SIFS Short Inter Frame Space

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

PIFS Priority Interframe Space

DIFS Distributed Interframe Space

EIFS Extended Interframe Space

BBP Baseband Processor

PLCP Physical Layer Convergence Procedure

PDF Probability Density Function

CDF Cumulative Distribution Function

OS	Operating System
TSC	Time Stamp Counter
ACPI	Advanced Configuration and Power Interface
OSPM	Operating System-directed configuration and Power Management
ICMP	Internet Control Message Protocol
MT	Mobile Terminal
DDP	Dominant Direct Path
NDDP	Non Dominant Direct Path
UDP	Undetectable Direct Path
CV	Coefficient of Variation
GDOP	Geometric Dilution Of Precision
QoS	Quality of Service
NIC	Network Interface Card
SNR	Signal to Noise Ratio
SNIR	Signal to Noise plus Interference Ratio
SME	Station Management Entity
CIR	Channel Impulse Response

Appendix B

Technical contribution to the IEEE 802.11v standard

B.1 Introduction

Recommendations of IEEE 802.11 for WLAN were thought with the main goal of providing all features of existing LAN to mobile users through a wireless access. In conventional LAN the location of devices connected to the LAN is not a problem since it is not difficult to identify the plug where a device is connected. Changes in the location of plugs are carefully tracked by network managers. When the access is wireless, it is not difficult to identify the AP that hosts a specific device, normally the one with the best radio path to the device. This gives a rough estimate of the placement of the device with a possible error similar in magnitude to the radio of the AP coverage. For some applications this error is too large, this fact leading to research in the area of more accurate location of WLAN devices.

Among the techniques that can be used to improve the accuracy of the location estimate, TOA shows interesting advantages. If the RTT of the signal from the WiFi client device to three APs could be measured, an accurate estimate of the location based on trilateration could be provided after easily converting RTT into distances to the AP (see Section 5). This would avoid the necessary synchronization of APs in TDOA and the limitations of the popular fingerprinting technique (building the fingerprints database, vulnerability to environmental changes).

Nowadays, obtaining the RTT at the WiFi device can be done with high accuracy by introducing changes in the hardware of the device and AP (PHY layer) in order to estimate the echo of the physical signal. This would require a non-negligible increase in the cost, so that it seems to be not suitable for commercial oriented solutions. However, the RTT can also be estimated through measuring the time between a MAC data frame is sent and its ACK is received. In such a solution, the clock of the WLAN card (which has a resolution of several nanoseconds) can be used to measure the RTT and then accurate ranging is totally feasible (see [1] and [2]) with minor hardware modifications. This last approach could be implemented by means of a pure software solution if the IEEE 802.11 MAC frames would be time-stamped in the transmission and reception using the WLAN card clock. Currently clocks in WLAN cards work at rates from tens of MHz on. If the number of clock cycles is used to stamp the transmitted packets, the RTT can achieve a precision of tens of nanoseconds. As an example, a commonly used clock of 44 MHz could estimate the RTT with an error of ±11.4 nanoseconds that leads to an error in estimated distance of 1.7 m. This estimate is subject to other error sources such as multipath and time consumed to detect the packet at the AP. These errors must be palliated with adequate filtering (see Sections 6 and 4).

B.2 Current standard

The widely deployed IEEE 802.11 standards (a,b,g) do not include ranging nor positioning capabilities. Currently, the IEEE 802.11 Working Group (WG) is working towards the definition of the IEEE 802.11v standard, which is going to support as a requirement some ranging and positioning capabilities. However, in the available IEEE 802.11v draft of March 2006 ([44]), the capability of time-stamping the transmission and reception of the MAC frames is not included as a requirement (this issue is left as open to the system's designer). This means that a pure software solution to easily achieve accurate TOA-based ranging (and then positioning) would not be totally feasible, so that hardware modifications in the client device would be still needed as nowadays occurs. The main capability which is included in this draft regarding TOA-based ranging is the estimation of the MAC processing time in the AP (time difference between the reception of the presence request in the AP and the transmission of the ACK) with a maximum resolution of nanoseconds and its inclusion as a field (time measurements field) in certain frames, capability which is also needed for the pursued objective.

B.3 Proposal

This note proposes the addition (as a requirement in the IEEE 802.11v standard) of the capability of timestamping the transmission and reception of certain MAC IEEE 802.11v frames (e.g. probe request, presence request) in the client device, with the maximum timing resolution (several nanoseconds) that can be provided by the available clock in the hardware of the WLAN device. This capability, added to the proposed "time measurements" field mentioned before included in the IEEE 802.11 draft, would allow to achieve accurate ranging (and then positioning) through a pure software solution. If this feature remains open (as in the draft) it would be very difficult to achieve this goal without hardware modifications, issue that we consider essential because directly affects the feasibility of the final solution.

Specifically, the instant to time stamp the end of the transmission of the frame that is sent from the client device to estimate the RTT (probe request, presence request...) occurs when the PHY layer issues the PHY-TXEND.confirm to the MAC sublayer. At that instant, the MAC sublayer could record the time using the WLAN card clock as time base. This value could be added as a new parameter in the MLME-PRESENCEREQUEST.confirm primitive the IEEE 802.11v draft, in order to provide it to the Station Management Entity (SME). On the other hand, the instant when the reception of the corresponding ACK frame is received in the device is also needed to calculate the RTT. This occurs when the PHY layer issues the PHY-RXSTART.indication of the ACK frame to the MAC sublayer. This time can be recorded by the MAC sublayer in the same way mentioned before and then stored to be reported later (not in that precise instant) to the SME as a new parameter of the MLME-PRESENCERESPONSE.indication primitive of the IEEE 802.11v draft, when the corresponding presence response is received in the device.

This way the client device knows the following instants: end of the transmission of the frame to the AP, start of the reception of the corresponding ACK and MAC processing time in the AP (this last one already defined as requirement in the IEEE 802.11v draft) with a resolution of nanoseconds, so that the TOA (and then the distance between both devices) could be accurately estimated by means of a pure software solution with a simple implementation.

Appendix C

Modified code of the Madwifi driver

First of all we declare the required variables in the *ath_stats* structure, which is defined in the *if_athioctl.h* file.

```
struct ath_stats {
...
u_int32_t ast_ant_rx[8]; /* rx frames with antenna */
u_int32_t ast_ant_tx[8]; /* tx frames with antenna */

u_int32_t ast_cpuclock;
u_int32_t ast_cpuclock_old;
u_int16_t ast_macretry;
};
```

Now we can see the code of the file *if_ath.c*, where the rest of the modified code is placed. First the needed libraries for the time captures are included and the needed variables declared.

```
#include <linux/smp.h>
#include <linux/timex.h>
#include <linux/cpufreq.h>
#include "reg.h"
...
static u_int32_t clocks=0;
```

Then the time-stamp is captured with the *rdtsc*() call at the interrupt handler, as explained in Section 3.3.

```
/*
 * Interrupt handler.  Most of the actual processing is deferred.
 */
irqreturn_t
#if LINUX_VERSION_CODE >= KERNEL_VERSION(2,6,19)
ath_intr(int irq, void *dev_id)
#else
ath_intr(int irq, void *dev_id, struct pt_regs *regs)
#endif
{
struct net_device *dev = dev_id;
struct ath_softc *sc = dev->priv;
struct ath_hal *ah = sc->sc_ah;
HAL_INT status;
int needmark;
```

```
rdtscl(clocks);
```

Inside the same interrupt handler function the captured time-stamp (CPU clock) is stored. This is not done until it is not known that the interruption was caused by the transmission of a frame.

```
if (status & HAL_INT_TX) {
#ifdef ATH_SUPERG_DYNTURBO

if (sc->sc_dturbo_switch) {
u_int32_t txqs = (1 << sc->sc_bhalq);
ath_hal_gettxintrtxqs(ah, &txqs);
if(txqs & (1 << sc->sc_bhalq)) {
sc->sc_dturbo_switch = 0;
/*
 * Hack: defer switch for 10ms to permit slow
 * clients time to track us.  This especially
 * noticeable with Windows clients.
 */
mod_timer(&sc->sc_dturbo_switch_mode,
  jiffies + msecs_to_jiffies(10));
}
}
#endif
sc->sc_stats.ast_cpuclock_old=sc->sc_stats.ast_cpuclock;
sc->sc_stats.ast_cpuclock=clocks;
```

The data processing to calculate the RTT is performed inside the $ath\_rx\_tasklet$ function, as shown below.

```
static void
ath_rx_tasklet(TQUEUE_ARG data)
{
...
int len, type;
u_int phyerr;
struct ieee80211_frame *wh;
u_int32_t diff;

... // after rx_accept

len = ds->ds_rxstat.rs_datalen;
bus_dma_sync_single(sc->sc_bdev,
bf->bf_skbaddr, len, BUS_DMA_FROMDEVICE);
bus_unmap_single(sc->sc_bdev, bf->bf_skbaddr,
sc->sc_rxbufsize, BUS_DMA_FROMDEVICE);

wh=(struct ieee80211_frame *) bf->bf_skb->data;

if( ((wh->i_fc[0] & IEEE80211_FC0_TYPE_MASK) == IEEE80211_FC0_TYPE_CTL) && ((wh->i_fc[0] & IEEE
diff=sc->sc_stats.ast_cpuclock-sc->sc_stats.ast_cpuclock_old;
printk(KERN_EMERG "CPU clocks=%lu *\n",(uintmax_t)diff);
}
```

The next step is notifying the driver that it must sent a transmission interrupt each time a frame is received. This way the time-stamp is always captured at the transmission frame handler. The mentioned notification is achieved by means of the following call inside the ath_tx_start function.

```
DPRINTF(sc, ATH_DEBUG_XMIT, "%s: set up txdesc: pktlen %d hdrlen %d "
"atype %d txpower %d txrate %d try0 %d keyix %d ant %d flags %x "
"ctsrate %d ctsdur %d icvlen %d ivlen %d comp %d\n",
__func__, pktlen, hdrlen, atype, MIN(ni->ni_txpower, 60), txrate,
try0, keyix, antenna, flags, ctsrate, ctsduration, icvlen, ivlen,
comp);

flags|=HAL_TXDESC_INTREQ;//immediate H/W interrupt
```

The last step consists of discarding the frame for time-stamp if retransmissions have occured. This is performed inside the function $ath\_tx\_processq$.

```
if(ni->ni_vap->iv_fixed_rate==-1)
if ((ds->ds_txstat.ts_status & HAL_TXERR_FILT) == 0 &&
    (bf->bf_flags & HAL_TXDESC_NOACK) == 0)
sc->sc_rc->ops->tx_complete(sc, an, ds);
sc->sc_stats.ast_macretry=ds->ds_txstat.ts_longretry;
```

Appendix D

List of publications

This appendix presents the publications associated with this PhD Thesis, which includes book chapters, international journals and conferences, and national conferences.

D.1 Book chapters

[1] M. Ciurana, I. Martin-Escalona, and F. Barcelo-Arroyo. *Location Based Services Handbook: Applications, Technologies and Security*, chapter Location in Wireless LAN. Taylor and Francis Group. ISBN: 978-1420071962, 2010.

[2] I. Martin-Escalona, M. Ciurana, and F. Barcelo-Arroyo. *Radio Communications*, chapter Location in ad hoc networks. Intechweb. ISBN: 978-9537619-X-X, 2010.

D.2 Journals

[1] M. Ciurana, F. Barcelo-Arroyo, and S. Cugno. A robust to multi-path ranging technique over IEEE 802.11 networks. *Wireless Networks (Springer Netherlands)*, Volume 15(number 3, April 2009).

[2] M. Ciurana, F. Barcelo-Arroyo, and S. Cugno. Tracking mobile targets indoors using WLAN and Time of Arrival. *Computer Communications (Elsevier)*, Volume 32(issue 13-14, August 2009).

D.3 International conferences

[1] M. Ciurana, F. Barcelo-Arroyo, and M. Llombart. Improving the performance of TOA over wireless systems to track mobile targets. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 1–6, 2009.

[2] M. Ciurana, D. Lopez, and F. Barcelo-Arroyo. SofTOA: software ranging for TOA-based positioning of WLAN terminals. In *Proceedings of International Symposium of Location and Context Awareness*, pages 207–221, 2009.

[3] M. Ciurana and F. Barcelo-Arroyo. Facing the obstructed path problem in indoor TOA-based ranging between IEEE 802.11 nodes. In *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2008.

[4] M. Ciurana, F. Barcelo-Arroyo, and S. Cugno. Positioning Uncertainty when Trilaterating Mobile Targets in 2D using Two Reference. In *Proceedings of ICT Mobile Summit*, 2008.

[5] M. Llombart, M. Ciurana, and F. Barcelo-Arroyo. On the scalability of a novel WLAN positioning system based on time of arrival measurements. In *Proceedings of Workshop on Positioning, Navigation and Communication*, pages 15–21, 2008.

[6] M. Ciurana, F. Barcelo-Arroyo, and S. Cugno. A novel TOA-based indoor tracking system over IEEE 802.11 networks. In *Proceedings of IST Mobile and Wireless Communications Summit*, 2007.

[7] F. Barcelo-Arroyo, M. Ciurana, I. Watt, F. Evenou, L. De Nardis, and P.Tome. Indoor location for safety applications using wireless networks. In *Proceedings of ERCIM Workshop on eMobilityn*, 2007.

[8] M. Ciurana, S. Cugno, and F. Barcelo-Arroyo. WLAN indoor positioning based on TOA with two reference points. In *Proceedings of Workshop on Positioning, Navigation and Communication*, pages 23–28, 2007.

[9] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo. A ranging method with IEEE 802.11 data frames for indoor localizations. In *Proceedings of IEEE Wireless Communications and Networking Conference*, pages 2094–2098, 2007.

[10] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo. A ranging method with IEEE 802.11 data frames. In *Proceedings of IEEE Radio and Wireless Symposium*, pages 133–136, 2007.

[11] M. Ciurana, F. Barcelo-Arroyo, and S. Cugno. Indoor Tracking in WLAN Location with TOA Measurements. In *Proceedings of ACM International Workshop on Mobility Management and Wireless Access*, pages 121–125, 2006.

[12] M. Ciurana, F. Barcelo-Arroyo, and S. Cugno. Multipath Profile Discrimination in TOA-based WLAN Ranging with Link Layer Frames. In *Proceedings of ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization*, pages 73–79, 2006.

[13] F. Izquierdo, M. Ciurana, F. Barcelo-Arroyo, and J. Paradells. Performance of a time-of-arrival technique for positioning WLAN terminals. In *Proceedings of IST Mobile and Wireless Communications Summit*, pages 25–30, 2006.

[14] F. Izquierdo, M. Ciurana, F. Barcelo-Arroyo, J. Paradells, and E. Zola. Performance evaluation of a TOA-based trilateration method to locate terminals in WLAN. In *Proceedings of IEEE International Symposium on Wireless Pervasive Computing*, pages 217–222, 2006.

D.4 National conferences

[1] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo. Estimación de distancias en redes IEEE 802.11 para localización indoor. In *Proceedings of Jornadas de Ingeniería Telemática*, pages 199–204, 2007.

[2] F. Izquierdo, M. Ciurana, F. Barcelo-Arroyo, and J. Paradells. Técnica de localización de terminales Wireless LAN basada en medidas Round Trip Time. In *Proceedings of Jornadas Telecom I+D*, pages 25–30, 2005.

Bibliography

[1] A. Kupper. *Location-based services: fundamentals and operation.* John Wiley and sons, LTD., 2005.

[2] J.A. Tauber. Indoor Location Systems for Pervasive Computing. Technical report, 2002.

[3] 3GPP. TS 03.71 Functional Stage 2 Description of Location Services (LCS). http://www.3gpp.org, 2002.

[4] 3GPP. TS 23.271 Functional Stage 2 Description of Location Services (LCS) R6. http://www.3gpp.org, 2004.

[5] RFC. 2138 RADIUS: Remote Authentication Dial In User Service. ftp://ftp.ietf.org/rfc/rfc2138.txt, 1997.

[6] RFC. 1157 SNMP: Simple Network Management Protocol. ftp://ftp.ietf.org/rfc/rfc1157.txt, 1990.

[7] Y. Chan Y. Chen and C. She. Enabling Location-Based services on Wireless LANs. In *Proceedings of IEEE International Conference on Networks (ICON)*, pages 567– 572, 2003.

[8] P. Bahl and V. Padmanabhan. Radar: An In-Building RF-based User Location and Tracking System. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pages 775–784, 2000.

[9] D. Sanghi S. Saha, K. Chaudhuri and P. Bhagwat. Location Determination of a Mobile Device using IEEE 802.11 Access Point Signals. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1987–1992, 2003.

[10] T. Kremenek P. Castro, P. Chiu and R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. In *Proceedings of Ubiquitous Computing (UbiComp)*, pages 18–34, 2001.

[11] A. Rudys L. Kavraki A. Ladd, K. Bekris and D. Wallach. Robotics-based location sensing using wireless ethernet. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pages 227–238, 2002.

[12] M. Youssef. *Horus: A WLAN-Based Indoor Location Determination System.* PhD thesis, University of Maryland at College Park, 2004.

[13] X. Chai and Q. Yang. Reducing the Calibration Effort for Location Estimation Using Unlabeled Samples. In *Proceedings of IEEE Pervasive Computing and Communications*, pages 95–104, 2005.

[14] H. Chu P. Huang Y. Chen, J. Chiang and A. Tsui. Sensor-assisted wi-fi indoor location system for adapting to environmental dynamics. In *Proceedings of ACM International Symposium on Modeling, Analysisand Simulation of Wireless and Mobile Systems (MSWIN)*, pages 118–125, 2005.

[15] F. Marx F. Evennou and E. Novakov. Map-aided indoor mobile positioning system using particle filter. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, volume 4, pages 2490–2494, 2005.

[16] S. Y. Seidel and T. S. Rapport. 914 MHz path loss prediction Model for Indoor Wireless Communications in Multi-floored buildings. *IEEE Transactions on Antennas and Propagation*, 40(2):207–217, 1992.

[17] X. Jia Y. Wang and H.K. Lee. An Indoors Wireless Positioning System Based on Wireless Local Area Network Infrastructure. In *Proceedings of International Symposium on Satellite Navigation*, 2003.

[18] P. Chatonnay F. Lassabe, P. Canalda and F. Spies. A Friis-based calibrated model for WiFi terminals positioning. In *Proceedings of IEEE World of Wireless Mobile and Multimedia Networks*, pages 382–387, 2005.

[19] H. Leppakoski A. Kotanen, M. Hannikainen and T.D. Hamalainen. Positioning with IEEE 802.11b wireless LAN. In *Proceedings of IEEE International Sympossium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, volume 3, pages 2218–2222, 2003.

[20] S. Ali and P. Nobles. A Novel Indoor Location Sensing Mechanism for IEEE 802.11 b/g Wireless LAN. In *Proceedings of Workshop on Positioning, Navigation and Communication (WPNC)*, pages 9–15, 2007.

[21] J.C. Hou H. Lim, L-C. Kung and H. Luo. Zero-Configuration, Robust Indoor Localization: Theory and Experimentation. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pages 1–12, 2006.

[22] T. Tamaki T. Uta N. Matsuzawa R. Yamasaki, A. Ogino and T. Kato. TDOA location system for IEEE 802.11b WLAN. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, volume 4, pages 2338–2343, 2005.

[23] E. Grab F. Winkler, E. Fischer and G. Fischer. A 60 GHz OFDM Indoor Localization System Based on DTDOA. In *Proceedings of IST Mobile and Wireless Communications Summit*, 2005.

[24] D. Niculescu and B. Nath. VOR base stations for indoor 802.11 positioning. In *Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pages 58–69, 2004.

[25] V. Lang and C. Gu. A locating method for WLAN based location service. In *Proceedings of IEEE International Conference on e-Business Engineering (ICEBE)*, pages 427–431, 2005.

[26] J. Austen-Francisco E. Elnahrawy and R.P. Martin. Adding Angle of Arrival Modality to Basic RSS Location Management Techniques. In *Proceedings of IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, 2007.

[27] S. Al-Jazzar and M. Ghogho. A Joint TOA/AOA Constrained Minimization Method for locating Wireless devices in Non-Line-of-Sight Environment. In *Proceedings of IEEE Vehicular Technology Conference Fall (VTC)*, pages 496–500, 2007.

[28] S. Venkatraman and J. Caffery. Hybrid TOA/AOA techniques for mobile location in non-line-of-sight environments. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, volume 1, pages 274–278, 2004.

[29] Ekahau real time location system. http://www.ekahau.com.

[30] AeroScout visibility system. http://www.aeroscout.com.

[31] Skyhook wireless WiFi positioning system. http://www.skyhookwireless.com.

[32] X. Li and K. Pahlavan. Super-Resolution TOA Estimation With Diversity for Indoor Geolocation. *IEEE Transactions on Wireless Communications*, 3(1):224–234, 2004.

[33] A. Aassie and A.S. Omar. Time of Arrival estimation for WLAN indoor positioning systems using Matrix Pencil Super Resolution Algorithm. In *Proceedings of Workshop on Positioning, Navigation and Communication (WPNC)*, pages 11–20, 2005.

[34] A. Ibraheem and J.Schoebel. Time of Arrival Prediction for WLAN Systems Using Prony Algorithm. In *Proceedings of Workshop on Positioning, Navigation and Communication (WPNC)*, pages 29–32, 2007.

[35] H. Reddy and G. Chandra. An improved Time-of-Arrival estimation for WLAN-based local positioning. In *Proceedings of International Conference on Communication Systems software and middleware (COMSWARE)*, 2007.

[36] H. Forstrom T. Dempsey D.D. McCrady, L. Doyle and M. Martorana. Mobile ranging using low-accuracy clocks. *IEEE Transactions on Microwave Theory and Techniques*, 48(6):951–958, 2000.

[37] S.A. Golden and S.S. Bateman. Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging. *IEEE Transactions on Mobile Computing*, 6(10):1185–1198, 2007.

[38] A. Günther and C. Hoene. Measuring Round Trip Times to Determine the Distance Between WLAN Nodes. *Lecture Notes in Computer Science, Networking*, pages 768–779, 2005.

[39] C. Hoene and J. Willmann. Four-way TOA and Software-Based Trilateration of IEEE 802.11 Devices. In *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2008.

[40] M. Klepal S. Basuki and D. Pesch. Time of Flight Ranging using Off-the-self IEEE802.11 WiFi Tags. In *Proceedings of International Conference on Positioning and Context-Awareness (POCA)*, 2009.

[41] S. Mazuelas J. Blas P. Fernández J. Prieto, A. Bahillo and R. M. Lorenzo. RTS/CTS mechanism with IEEE 802.11 for indoor location. In *Proceedings of NAV08/ILA37, The Navigation Conference and Exhibition*, 2008.

[42] S. Mazuelas R.M. Lorenzo J. Blas J. Prieto, A. Bahillo and P. Fernández. Adding indoor location capabilities to an IEEE 802.11 wlan using real-time RTT measurements. In *Proceedings of IEEE Wireless Telecommunications Symposium*, 2009.

[43] B. Rathke M. Abusubaih and A. Wolisz. A Dual Distance Measurement Scheme for Indoor IEEE 802.11 Wireless Local Area Networks. In *Proceedings of IFIP/IEEE International Conference on Mobile and Wireless Communication Networks (MWCN)*, 2007.

[44] The Institute of Electrical and Electronics Engineers. IEEE 802.11 WG. Draft Amendment to Standard for Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements-Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment v: Wireless Network management. IEEE P802.11v/D0.02, march 2006.

[45] Intersil. Information and datasheets available at:. http://www.intersil.com/.

[46] F. Izquierdo. Wireless LAN location technique based on Round-Trip-Time measurements. Master's thesis, Universitat Politècnica de Catalunya, 2005.

[47] H. Hashemi. The indoor radio propagation channel. *Proceedings of the IEEE*, 81(7):943–968, 1993.

[48] F. Lo Piccolo D. Giustiniano and N.B. Melazzi. Relative localization in 802.11/GPS systems. In *Proceedings of International Workshop on Satellite and Space Communications*, pages 289–293, 2007.

[49] P. Saxholm E. Corell and D. Veitch. A user friendly TSC clock. In *Proceedings of Passive and Active Measurement Conference*, 2006.

[50] Madwifi. http://madwifi-project.org.

[51] D.P. Bovet and M. Cesati. *Understanding the Linux Kernel*. O'Reilly, 2005.

[52] A. Rubini and J. Corbet. *Linux Device Drivers*. O'Reilly, 2001.

[53] Syslog. http://www.ibiblio.org/pub/Linux/docs/LuCaS/Manuales-LuCAS/doc-unixsec/unixsec-html/node86.html.

[54] Ubuntu. http://www.ubuntu.com.

[55] Real-time linux. http://rt.wiki.kernel.org/index.php/Main_Page.

[56] X. Li N. Alsindi and K. Pahlavan. Performance of TOA Estimation Algorithms in Different Indoor Multipath Conditions. In *Proceedings of Wireless Communications and Networking Computing*, pages 496–500, 2004.

[57] J.Caffery S.Venkatraman and H.-R. You. Location using LOS range estimation in NLOS environments. In *Proceedings of IEEE Vehicular Technology Conference*, volume 2, pages 856–860, 2002.

[58] P. Krishnamurthy K. Pahlavan and J. Beneat. Wideband radio propagation modeling for indoor geolocation applications. *IEEE Communications Magazine*, 36:60–65, April 1998.

[59] W.K.Tam and V.N.Tran. Propagation modelling for indoor wireless communication. *Electronics and Communications Engineering Journal*, 7(5):221–228, 1995.

[60] K.Pahlavan P.Krishnamurthy and J.Beneat. Radio propagation modelling for indoor geolocation applications. In *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 1, pages 446–450, 1998.

[61] M. Latva-aho X. Li; K. Pahlavan and M. Ylianttila. Comparison of Indoor Geolocation Methods in DSSS and OFDM Wireless LAN Systems. In *Proceedings of IEEE Vehicular Technology Conference*, volume 6, pages 3015–3020, 2000.

[62] K. Pahlavan X. Li and J. Beneat. Performance of TOA Estimation Techniques in Indoor Multipath Channels. In *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2002.

[63] S.J. Howard and K. Pahlavan. Measurement and Analysis of the Indoor Radio Channel in the Frequency Domain. *IEEE Transactions on Instrumentation and Measurements*, 39(35), October 1990.

[64] H. Kobayashi S. Gezici and H.V. Poor. Non-Parametric Non-line-of-Sight Identification. In *Proceedings of IEEE Vehicular Technology Conference*, volume 4, pages 2544–2548, 2003.

[65] M.P. Wylie and J. Holtzman. The Non-Line-Of-Sight problem in mobile location estimation. In *Proceedings of IEEE International Conference on Universal Personal Communications*, 1996.

[66] J. Riba and A. Urruela. A robust multipath mitigation technique for time-of-arrival estimation. In *Proceedings of IEEE Vehicular Technology Conference*, volume 4, pages 2263–2267, 2002.

[67] J. Vidal-J.A.Castro M. Nájar, J.M. Huerta. Mobile Location with bias tracking in Non-Line-Of-Sight. In *Proceedings of IEEE Internationnal Confernce on Acoustics, Speech, and Signal Processing*, pages 1–4, 2004.

[68] G. Strang. The Mathematics of GPS. *SIAM News*, 30(5), 1997.

[69] N. Levanon. Lowest GDOP in 2-D scenarios. *IEE Proceedings on Radar, Sonar and Navigation*, 147(3), 2000.

[70] J. Vidal M. Najar and A. Kjellstrom. Kalman Tracking for UMTS Mobile Location. In *Proceedings of IST Mobile and Wireless Communications Summit*, pages 230–235, 2001.

[71] H.Tsuji B.Long Le, K.Ahmed. Mobile Location Estimator with NLOS Mitigation Using Kalman Filtering. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1969–1973, 2003.

[72] A. Mitiche Q. Cai and J.K. Aggarwal. Tracking human motion in an indoor environment. In *Proceedings of International Conference on Image Processing*, pages 215–218, 1995.

[73] S. Niemi J. Latvala, J. Syrjärinne and J. Niittylahti. Patient Tracking in a Hospital Environment Using Wireless Stations and Extended Kalman Filtering. In *Proceedings of Middle East Conference on Networking*, 1999.

[74] Chao-Lin Chen Po-Hsuan Tseng and Kai-Ten Feng. An Unified Kalman Tracking Technique for Wireless Location Systems. In *Proceedings of IEEE International Symposium on Wireless Pervasive Computing*, pages 5–7, 2007.

[75] R. Quiros E. Pulido and H. Kaufmann. Analysis of a Kalman Approach for a Pedestrian Positioning System in Indoor Environments. In *Proceedings of International Euro-Par Conference*, pages 931–940, 2007.

[76] F.Evennou and F. Marx. Improving Positioning capabilities for indoor environments with WiFin. In *Proceedings of IST Mobile and Wireless Communications Summit*, 2005.

[77] R. Jordan I. Guvenc, C. T. Abdallah and O. Dedeoglu. Enhancements to RSS based indoor tracking systems using kalman filters. In *Proceedings of International Signal Processing Conference and Global Signal Processing*, 2003.

[78] D.Macagnano and G.Abreu. Tracking multiple dynamic targets with multidimensional scaling. In *Proceedings of IEEE Personal Indoor Mobile Radio Communication*, pages 1–5, 2007.

[79] R. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Eng. 82D*, pages 35–45, 1960.

[80] B.P.Burke P.Misra and M.M. Pratt. GPS performance in navigation. *Proceedings of the IEEE (Special Issue on GPS)*, 87(1):65–85, 1999.

[81] The Institute of Electrical and Electronics Engineers. IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications P802.11, november 1997.

[82] OMNeT++. http://www.omnetpp.org/.

[83] Inet. http://www.omnetpp.org/staticpages/index.php?page=20041019113420757.

[84] L. Fratta G. Bianchi and M. Oliveri. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In *Proceedings of International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 392–396, 1996.

[85] T. Kwon S. Pack, J. Choi and Y. Choi. Fast-Handoff Support in IEEE 802.11 Wireless Networks. *IEEE Communications Surveys and Tutorials*, 9(1), 2007.

[86] K. Long Y. Li, C. Wang and W. Zhao. Modeling Channel Access Delay and Jitter of IEEE 802.11 DCF. *Wireless Personal Communications*, 47(3):417–440, 2008.