UNIVERSITÀ DEGLI STUDI DI PALERMO

POLITECHNIC SCHOOL

DEIM - Dept. of Energy, Information Engineering and Mathematical Models

Ph.D course in Electric, Electronic and Telecommunication Engineering,
Mathematics and Automation

XXV cycle of study

ING-INF/03

Indoor Localization using Cognitive Radios and Software-defined Wireless Networks

PhD Candidate:
Ing. Pierluigi Gallo

Advisor:
Prof. Ilenia Tinnirello

PhD Course Coordinator:
Prof. Francesco Alonge

Year 2014

You can’t reach your destination
if you don’t know where you are
and where you’re going
— anonymous

To my parents, who taught me most,
to my lovely wife, who shares the load and the joy of life with me,
to my sons, who teach me something new every day.

Acknowledgements

This work has been supported in part by the EU FP7-258301 (CREW) and would not be possible without the precious suggestions from my advisor Ilenia Tinnirello and for the support and illuminating discussions with Giuseppe Bianchi, Domenico Garlisi, Francesco Gringoli, and Stefano Mangione. The author would like to thank Nicoló Facchi for his help in editing the appendix.

P. G.

Contents

List of Figures

Introduction

In recent years the impressive boost of location-based services have had a significant impact on people lifestyle, changing forever social and spatial interactions. Global Positioning System (GPS) is the dominant technology for localization and tracking outdoors, whereas positioning solutions indoors are not mature enough. A prevalent approach has not yet emerged from the plethora of research proposals, which are effective only in specific contexts or in niche scenarios. Nonetheless, standardization is still at a early stage and caution has been taken to avoid decisions impairing market adoption.

The market has remained open to innovation and current indoor localization proposals span different tradeoffs between performances, costs and scalability. Among these, Wi-Fi based localization technologies look promising because they are extremely pervasive both in terms of deployed infrastructures and available mobile terminals. This would permit to cover large indoor areas including airports, hospitals, schools, and warehouses with a reduced investment and exploiting existing infrastructures. However, a series of issues arises because Wi-Fi technology was originally designed to extend wired networks to the wireless domain, not taking into account localization goals.

The current wireless standardization process tries to converge standard extensions, aimed to face the myriad of possible operational conditions, in a single one-size-fits-all solution. The fragility of such approach has recently emerged in terms of complexity and backward compatibility. In the same way, the quest for a universal indoor localization standard is doomed to failure. In fact, the intrinsic limits of Wi-Fi technology are even worsened by overloading wireless devices with extra interoperability requirements and the strategies of coexistence between data transport and localization services.

Furthermore, radio-based indoor positioning is sensitive to the mutable channel conditions (multipath, spectrum utilization, AP density, channel usage, ...), to hardware capabilities and status (availability of inertial sensors, battery level, multiple-antennas, ...), to geometric constraints (AP alignment, algorithm-dependent special cases, ...), to knowledge about the context (AP positions, radio maps, floor plans, mobility model, ...) and application requirements (refresh rate, still localization / tracking, accuracy, ...).

Light, noise and radio propagation conditions are quite different during night and day in indoor environments, therefore multiple sets of training data have been used depending on the time of day. This parametric approach helps facing context-dependency but it is not flexible enough to adapt to mutable conditions.

Another approach to overcome context dependency is the introduction of programmability and adaptability in localization devices and mechanisms by the mean of open Application Programming Interface (API). A similar approach has been recently pursued in the wireless communication field to enable cognitive and self-adapting Wi-Fi networks.

Advanced sensing capabilities, fine-grained control over transceivers and computation of modular localization algorithms are the basic building blocks for flexible localization systems. The cognitive radio paradigm is a key enabler of advanced sensing, making devices capable to observe the environment and provide optimal adaptation. This is possible by means of flexible Medium Access Control (MAC) and Physical Layer Protocol (PHY) layers, obtained through clear abstractions and the decoupling between the platform and its behavior.

The resulting *distributed sensing intelligence* requires a centralized control to enforce the best strategies for data transfer, localization and their coexistence. Software-defined networking provides control messages able to configure Access Points (APs) and mobile devices.

Localization algorithms receive input observations from the radio and inertial/visual sensors and provide output estimated position. Well-defined APIs and an OSGi framework provide the required application-layer modularity, permitting to run different algorithms given a common set of input data.

This thesis focuses on flexible sensing and control capabilities for indoor positioning. These are based on few positioning principles, which are overviewed in chapter 1. A new flexible indoor Indoor Positioning System (IPS) architecture is described in chapter 2. It lies at the intersection of SDN, cognitive radio and Open Service Gateway initiative (OSGi). Next chapters present multiple use cases that validate the proposed architecture and provide new methodological findings for the following positioning principles:

- *Time of Arrival (ToA)* - Accurate time-of-flight measurements with nanosecond accuracy are obtained through a dedicated MAC policy which starts and stops timers and use low-level signals from the radio platform. This use case is therefore enabled by MAC-layer flexibility. Motivation, design and experimental results are described in chapter 3;
- *Differential Time of Arrival (DToA)* - Mimicking bistatic radars and exploiting the bidirectional flows between APs and their associated stations, the possible target positions are taken at the intersections between one ellipsis and one hyperbola. This use case is enabled by flexibility both at MAC and PHY layers obtained with a software-defined radio platform. Working principles, features and experimental results of this use case are drawn in chapter 4;
- *Angle of Arrival (AoA)* - Directions of arrival from the hearable APs are obtained by

analyzing angular power profiles at the target position. We named this methodology ‘on-demand localization‘ because it requires a little human intervention. This use case validates the sensing sub-system of the proposed architecture, which includes the magnetic sensor of user’s smartphone. Design, implementation and experimental validation are described in chapter 5;

Fingerprinting - The fingerprinting use case shows both static positioning and dynamic tracking. Panoramic radio maps and user rotations produce static positioning whereas user’s heading and directional radio maps can follow user’s movements. This use case is enabled by advanced control capabilities, which join the sensing and mobility data coming from mobile terminals with those from the access points. Motivations, system description and preliminary results are described in chapter 6;

- *Pattern matching* - A path-following mechanism uses the smartphone camera as a sensor that recognizes visual indications on the floor, which are used to navigate visually impaired people. The use case validates the cognitive cycle by means of an extended API for visual sensing, the effects of human in the (control) loop, the ability to control vibrational actuators. Motivation, design as well as simulation and experimental evaluation are given in chapter 7.

1 Wi-Fi indoor positioning and programmable wireless networks

1.1 Taxonomy of Wi-Fi based indoor localization systems

Indoor localization relies on several radio, inertial and vision-based techniques and a plethora of algorithms[1]. In this thesis we focus and provide contributions on Wi-Fi based indoor localization. We are motivated by the pervasive coverage of Wireless Local Area Networks (WLANs) and their massive diffusion.

When propagation of Wi-Fi signals is used to infer user's position, the localization process is named radiolocation and is generally based on well-known schemes such as trilateration, multi-lateration, triangulation, scene analysis, and proximity [2].

Localization techniques can be categorized in two types depending on the kind of the provided information: physical location and symbolic location. Physical location identifies user's position on a 2-D/3-D map by means of coordinates. This can be both absolute and relative in accordance to the reference coordinate system. Symbolic location, also known as logical location [3], indicates places accordingly to typical activities or context peculiarities: in the office, in the bedroom, in the market, etc.

Wi-Fi based indoor localization systems are composed by one or multiple transmitters and one or multiple receivers. These are usually the user's handset and the network infrastructure, which includes several APs. Depending on devices which take observations, run algorithms and consumes the localization service, these can be classified as: remote positioning, self-positioning, indirect remote positioning and indirect self-positioning.

In *remote positioning*, also known as *network-side localization*, the mobile device transmits signals while fixed network devices take care of the sensing and computational burden. Observations generally flow to a centralized station, where the localization algorithm is run.

In *self-positioning*, also referred as *handset-side localization*, the mobile device is the sensing unit while fixed transmitters are positioned in known locations.

These schemes have two corresponding counterparts that require a data transport between the mobile device and the network infrastructure in order to provide positioning results to the other side.

Indirect remote positioning and *indirect self-positioning* require a data transport between the mobile device and the network infrastructure to send periodically the estimated position. In fact, nodes that use the positioning information may be different than those who perform sensing and computation, as shown in table 1.1.

Table 1.1: Classification of Wi-Fi based indoor localization schemes

Positioning scheme		**sensing /computing platform**	
		handset	network
consuming platform	handset	self-positioning	indirect remote positioning
	network	indirect self-positioning	remote positioning

1.1.1 Positioning principles

Indoor positioning systems can be classified accordingly to their functioning principles. Lateration estimates the position of an object by measuring its distances from multiple reference points (circular lateration) or the difference of ranges between an object and the reference points (hyperbolic lateration). Determining the distance between an object and a reference point is called *ranging*. Distances are generally obtained by means of their direct proportionality to the propagation time. Angulation is based on angles measured between the direction of arrival of a signal and points at known positions.

Basic principles of the most used localization techniques are discussed in the rest of the present chapter. The advantages and disadvantages of these technologies suggest to use the one that works best in the specific context.

Circular lateration and time of arrival

Given n reference points at known positions, their ranging observations from the target permit to build a system of n nonlinear equations. The special case of three reference points is known as trilateration, and is depicted in fig. 1.1. Given the distance from an AP, the target lays on a circle centered in the AP. The intersection of three circles provide the target position.

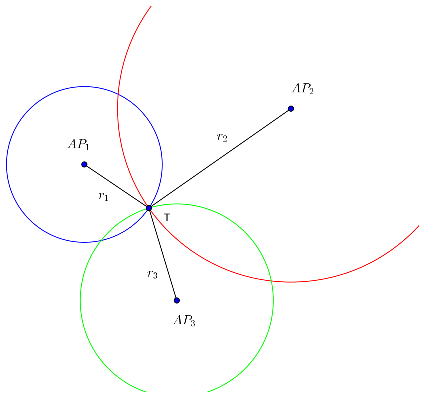


Figure 1.1: Circular trilateration based on ranging from APs at known positions.

Equations which describe circular trilateration are:

$$\begin{cases} \sqrt{(x_{AP_1} - x_T)^2 + (y_{AP_1} - y_T)^2} = r_1 \\ \sqrt{(x_{AP_2} - x_T)^2 + (y_{AP_2} - y_T)^2} = r_2 \\ \sqrt{(x_{AP_3} - x_T)^2 + (y_{AP_2} - y_T)^2} = r_3 \end{cases}$$

This method can be extended to 3D by considering three spheres rather than circles, which centers are the base stations. The intersection between two spheres is a circle and the intersection between this circle and the remaining sphere provides two points. To remove the ambiguity about these points, it is necessary to edit an extra equation (therefore an extra sphere) for another access point.

When ranging is obtained by measuring the absolute propagation time, the method is named *Time of Arrival (ToA)*. The name *direct Time of Arrival (ToA)* refers to the time necessary for a direct flight from the transmitter to the receiver. Measuring this time interval requires synchronization of all transmitters and receivers as well as timestamps on all the Wi-Fi frames. These requirements can be skipped if a single reference clock is used. Observing the Roundtrip Time of Flight (RToF), the time taken by the signal to travel back and forth between transmitter and receiver, the propagation time is computed at the same device, and therefore is affected only by the clock short-term stability.

The contribution provided to ToA and RToF is described in details in chapter 3.

Hyperbolic lateration and time difference of arrival

Hyperbolic lateration uses the difference between ranges, being the hyperbola is the set of all points for which the difference between two distances is constant. Given r_i, r_j the ranging of

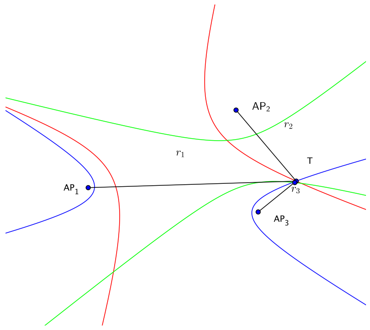


Figure 1.2: Hyperbolic trilateration based on ranging from APs at known positions.

the target from AP_i and the AP_j AP, the difference defines an hyperbola. The intersection of three hyperbolas define the target position. The special case of three hyperbolas is depicted in fig. 1.2 and is named hyperbolic trilateration.

Equations corresponding to fig. 1.2 are the following:

$$\begin{cases} \sqrt{(x_{AP_1}-x_T)^2+(y_{AP_1}-y_T)^2}-\sqrt{(x_{AP_2}-x_T)^2+(y_{AP_2}-y_T)^2}=r_1-r_2 \\ \sqrt{(x_{AP_1}-x_T)^2+(y_{AP_1}-y_T)^2}-\sqrt{(x_{AP_3}-x_T)^2+(y_{AP_3}-y_T)^2}=r_1-r_3 \\ \sqrt{(x_{AP_2}-x_T)^2+(y_{AP_2}-y_T)^2}-\sqrt{(x_{AP_3}-x_T)^2+(y_{AP_3}-y_T)^2}=r_2-r_3 \end{cases}$$

Equations can be extended in 3D, considering an hyperboloids rather than an hyperbolas and one extra equation. When difference between time propagation values are available, as depicted in figure, the method is named *Differential Time of Arrival (DToA)*.

Both circular trilateration and hyperbolic trilateration are prone to errors, which sum to true distances. The measures affected by errors are named pseudoranges. Circles and hyperbolas do not intersect in a point but define an area. When more than three access points are available, than three the system is over-determined because of , the process is called *multilateration* and is generally solved using the least square algorithm.

Our findings on a novel interpretation of hyperbolic lateration, using an analogy with bistatic radars is described in chapter 4.

Triangulation and Angle of Arrival

Estimation of the target's position having the angles towards points at known coordinates is an old problem. It was known among sailors as *resection* or, more generally, *angulation*.

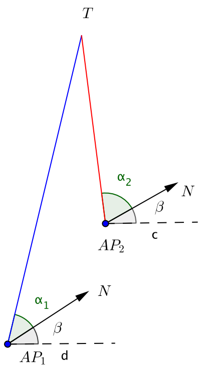


Figure 1.3: Angulation considering angles from a reference direction (e.g. magnetic North).

When angles define the direction of arrival of a signal it is referred as Angle of Arrival (AoA) or Direction of Arrival (DoA). It is a general opinion that base stations or mobile devices have to be equipped with antenna arrays or directional antennas oriented by motors, although we introduce a simple Angle of Arrival (AoA) technique relying on a little user's intervention, described in chapter 5.

The basic principle behind angulation in 2-D is described in fig. 1.3. The estimated target point is obtained by the intersection of two rays coming out from two access points at known positions. The magnetic North is generally used as the reference direction to measure angle. The angle β is known and takes care of the angular shift between the North direction and the x-axis of the coordinate system. The extension to 3-D can be obtained adding another access point and its angle.

$$\begin{cases} \alpha_1 + \beta = \arctan\left(\dfrac{y_{AP_1} - y_T}{x_{AP_1} - x_T}\right) \\ \alpha_2 + \beta = \arctan\left(\dfrac{y_{AP_2} - y_T}{x_{AP_2} - x_T}\right) \end{cases}$$

Angle measurements are affected by errors, therefore the intersection of two rays becomes the intersection of two circular sectors, which describes an area. Our contribution to AoA is reported in chapter 5.

Fingerprinting

Fingerprinting, also known as *pattern matching* or *scene analysis*, considers observations taken in a scene and compares them with previously collected measurements.

Scene analysis is the name used for optical pattern matching where images taken by the observer or by the surroundings (e.g. user's camera or surveillance cameras) are compared with previously collected images of the scene. Static scene analysis compares a snapshot of a scene with others previously recorded at different positions. Dynamic scene analysis considers the differences between images successively taken in time.

Fingerprinting techniques share a key basic principle: observations (e.g. signal strength values) are measured over a grid of known positions during an off-line phase (training data), then the measure is taken in the target point during the online phase (testing data) [4]. For each point of the grid the observer collects the signal strength from multiple base stations and obtains a RSS vector for each point. This vector is called a fingerprint. Fingerprint analysis is done through Euclidean distance, probabilistic Bayesian approach, best matching point, or the centroid of the best N points. In case of Wi-Fi power signals, the training data are called radio maps [5].

New findings on patter matching are presented in chapter 6. The usage of *panoramic fingerprinting* for static scene analysis and *angular fingerprinting* for dynamic scene analysis permits both static positioning and mobile object tracking. As for visual scene analysis, we navigate visually impaired people making them follow a path (see chapter 7).

1.1.2 Performance and coupling hurdles

Techniques that exploit distances or angles are more influenced by errors in comparison to the recognition of particular observations in the scene. This is caused by the difficulties of indoor propagation modeling, which is sensitive to context peculiarities and dynamics (site specifics, furniture, moving object and people, scatterers, etc). Furthermore, indoor propagation is generally affected by severe multipath [6, 7].

Another obstacle to a flexible indoor positioning is the association between sensing technologies and location algorithms. Despite some observations are traditionally tight to a specific class of algorithms, they can be used in not-ordinary ways. As for example, Received Signal Strength Indication (RSSI) values are generally coupled to fingerprinting algorithms or attenuation models, but as shown in chapter 5, they can be proficiently employed to estimate the AoA.

These two considerations shed light on two desirable requirements: (i) the adaptability of the Indoor Positioning System (IPS) to the context; (ii) a clear decoupling between sensing observations and the positioning algorithms. An inspiring approach to both directions has been identified in the wireless domain, where recent trends have boosted cognitive, self-managing and software-defined networks. Furthermore, the quest for flexibility in wireless communication provides direct effects also on Wi-Fi based indoor localization, which requires dynamic reconfiguration both at PHY and MAC layers.

Programmability at PHY layer, including selection of modulation and coding, has been provided by the mean of FPGA-based and/or Software Defined Radio platforms [8, 9, 10, 11, 12]. The diffusion of these platforms is generally limited to research projects or wireless network prototyping because of high costs, required programming skills, non-real-time operation, and lack of code reusability [13]. The need for flexibility and programmability at MAC-layer has been satisfied by several frameworks both for WLANs [14, 15, 16, 17] and wireless sensor networks [18]. These leverage the principle that network operations can be defined in software and therefore can be dynamically reprogrammed on the same hardware. Among those frameworks, we selected the Wireless MAC Processor (WMP) [17] as a cornerstone element in our architecture for cognitive wireless localization. For the sake of completeness, an overview of the Wireless MAC Processor (WMP) is drawn in the following paragraph.

1.2 Wireless MAC Processor overview

The WMP provides a clear decoupling between a general-purpose wireless platform and its behavioral model. The first is abstracted by means of a set of primitives, the latter is modeled using an high-level description language that uses those primitives as elementary building blocks. According to this vision, wireless protocols should not be designed once for all, but can be redefined 'on-the-fly', tailored to specific context. This would make protocols simpler, because they would not require including many operational conditions.

The WMP turns a non-programmable wireless node into a programmable platform. Initially designed for cognitive and self-managing wireless networks it is usable also beyond the boundaries of networking scenarios and is a strategic booster for indoor localization, as described in chapter 2.

The WMP uses *non-reprogrammable hardware capabilities* provided by the following subsystems:

- the *transceiver*, which deals with the reception and the transmission of the frames, uses modulation and coding schemes provided by the PHY;
- the *transmission queues*, in which traffic flows or control and management frames can be separately enqueued for achieving different MAC performance;
- the *reception queue*, where incoming packets can be stored before being forwarded to the host;
- the *memory blocks*, mostly available as configuration registers, able to store programs and variables.

These sub-systems can be controlled by simple and well-defined *actions* such as transmission, timer setting, reading or writing registers, etc. They can asynchronously provide *events* which occur when the channel becomes busy, a timer expires, a new frame is enqueued, etc. User's configuration parameters, values provided by actions and the persistent image of an event[1] are stored in memory blocks. These can be composed in logical expressions resulting in enabling *conditions*.

Actions, events and conditions represent the API. These are the building blocks available to the MAC programmer to define the operational behavior of the device by defining and arranging states and transitions, composing them in eXtended Finite State Machines (XFSMs). Rather than be controlled by standardized pre-defined protocols, these non-programmable hardware sub-systems are governed by a generic XFSM execution engine. It reacts to events from the internal system and those from the external channel depending on the state of the hardware, the current state of the XFSM and the values taken by global variables.

XFSMs define MAC programs, which are named MAClets to underline their ability to be injected in wireless nodes by authorized and centralized controllers. The control plan is in charge of exchanging MAClets and sending other control information among network devices, as detailed in [19].

The MAC engine does not need to know which MAC program a new fetched state belongs to and the definition of code switching transitions are logically independent of the MAC program. This allows the WMP architecture to support switching transitions into a second-level state machine, named meta-machine, whose states represent the current MAC program. The WMP is therefore able to execute multiple concurrent MACs.

The WMP enables cognitive, self-managing and software-defined wireless networking. Next chapter provides a modular framework that extends this flexibility also to non-network capabilities, such as indoor localization.

[1] e.g. after a frame is received, a specific bit is set to 1 and advertises the occurrence of the event till it is read, then is reset to 0

2 A cognitive indoor positioning architecture

The goal of this chapter is to present a novel IPS architecture able to decouple localization functions and algorithms from the available radio and sensing platform. For this purpose, we exploit similarities with wireless architectures, both at node and network levels, and context-aware architectures.

As stated in the previous chapter, Wi-Fi based localization requires adaptation to the context and to the operating conditions. This goal perfectly matches the behavior of cognitive wireless networks, which are able to sense the environment and to adapt to it. The different perspective regards the final objective of these systems: localizing people and objects rather than providing optimal transmission strategies over the wireless channel. However, these two aspects coexist because the same hardware is used to obtain both networking and non-networking goals.

Our indoor positioning architecture is modular, extensible and relies on three main aspects:

- the definition of localization-specific abstractions (localization capabilities of the underlying radio platform);
- the cognitive cycle $measure \rightarrow decide \rightarrow act$;
- the ability to recognize and analyze the context.

2.1 Motivation

Current localization solutions work well only in specific contexts, under specific conditions or with certain hardware configurations, despite they claim to be general. The quest for a universal indoor localization system has not yet provided a single solution because current approaches have both pros and cons depending on the context and the operating conditions.

We clarify the influence of the context by providing a simple explanation which consider different topological ways to face the localization problem. As stated in the previous chapter, positioning can be categorized in network-side and handset-side, depending on which nodes take observations and run localization algorithms. This simple preliminary choice has a significant impact on the following aspects related to the availability of localization services.

Localization at the handset side provides better privacy because the user is the only one who knows about his position. Another advantage is the portability. In principle, a location-capable user's device can work in different indoors with various infrastructures, eventually requiring an infrastructure-dependent reconfiguration. Handset-side localization may require knowledge on the infrastructure (e.g. positions and MAC address of the APs, TX power of transmitters, radio maps, etc.) but these information can be easily broadcasted by APs inside beacon frames.

Localization at the network side exploits better understanding of the infrastructure and can be ideally applied to any handset device that comes inside the network coverage. In case of passive sensing, the network can localize even unaware or uncooperative users, which is particularly interesting in monitoring, rescue and military applications. This approach provides requirements to the network infrastructure, ideally any device can be located as long as it keeps within the instrumented area.

Hybrid localization schemes take observations and decisions both on network and client devices. Such systems generally provide good performances due to redundant sensing devices (e.g. multi-antenna systems at the network side and inertial sensors on the handset). The cooperation between network and mobile devices requires their interoperability both in hardware and software. Therefore, the network can localize only previously configured specific devices, and only as long as they are within the instrumented area.

A possible solution would consist in *building-specific configuration* for the localization application. Configuration parameters include, among others, APs location, the propagation model, eventual radio maps and building plans. Although this solution provides a partial flexibility, it is not optimal because pre-defined settings cannot provide adaptation and therefore are not able to follow context dynamics.

Four different approaches listed above can be summarized in terms of their limiting aspects: in the first case the system is tight to specific devices, in the second to specific areas, then to specific devices *and* specific areas, and finally, limitation consists in specific operating conditions.

To overcome the limits of current indoor localization systems, we propose Modular Architecture for FLexible Wi-Fi Networking and Indoor Positioning systems (MAFLIP), a flexible localization architecture, which guarantees both networking and non-networking functionality.

2.2 The positioning system architecture

Our positining architecture is inspired by three emerging technology trends in networking: (i) SDN for wireless MAC programmability, recently introduced by the Wireless MAC Processor

(WMP) [17, 19, 20]; (ii) OpenDaylight controller architecture [21], which lies at the intersection of SDN, Model Driven Software Engineering (MDSE) and Model Driven Network Management (MDNM); (iii) Service-Oriented Context-Aware Middleware (SOCAM) architecture, which provides reasoning about contexts [22].

The high-level definition of our architecture is depicted in figure 2.1. This envisions cognitive IPSs based on context estimation. Available observations are analyzed to estimate surrounding context, then the system takes decisions about the most appropriate localization functions and algorithms. This is closely related to the cognitive cycle in new generation networks and can be indefinitely reiterated in a closed-loop that is able to follow the context dynamics. The enabler for this approach is a clear decoupling between the sensing platform and localization functions and algorithms. Unlike the inspiring architectural frameworks, we provide the following functions:

- coexistence between network and positioning functions;
- coexistence between network applications and positioning algorithms;
- the possibility to configure MAC and PHY depending on non-network sensing capabilities (e.g. inertial, optical, magnetic, ...) and data they provide;
- dedicated positioning controllers.

For this purpose, ordinary networking hardware is reused for different from data transfer purposes. This requires to extend current network controlling paradigms in order to include also non-network operations.

Modules that perform networking goals are represented in fig. 2.1 as white boxes whereas non-network modules, including inertial sensors, positioning functions and algorithms, are depicted in gray. Among possible non-network controllers we take into account Positioning and Tracking Controller (PTC)s. Network controllers and Positioning and Tracking Controllers (PTCs) implement policies that are inherently dynamic and depend on temporal conditions and external events.

Our architecture is based on few enabling technologies: software defined radio platforms, the WMP implementation and a modular Java framework. Full flexibility is obtained when all these architectural elements are present, but in general legacy nodes do not provide neither software-defined radios nor WMP. The capabilities of the underlying platform can be very heterogeneous as well as the the low-level API provided by the vendor can be very poor or extremely rich. We face such heterogeneity by means of pluggable interfaces, introduced in paragraph 2.4.

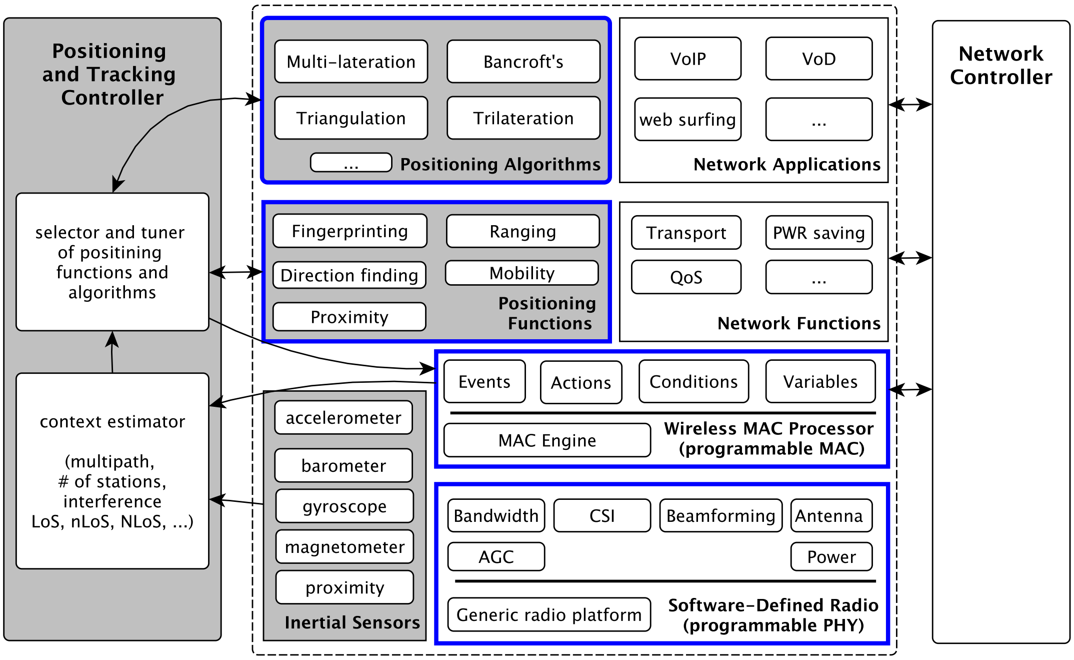


Figure 2.1: Modular Architecture for flexible Wi-Fi networking and Indoor Positioning Systems

2.3 Multi-tenancy and multi-application localization Control

In our vision, both network and localization services are provided in a multi-tenancy scenario. Multiple controllers operated by different providers act on the same device. These opportunely select, install and configure high-level functions and applications, depending on measurements coming from low-level sub-modules. Two kinds of controllers are envisioned: (i) *Network Controllers*, which apply rules that are typical for SDN [19]; (ii) *Non-network Controllers*, which also include sensing capabilities of smartphone devices and consider them as nodes of a mobile Wireless Sensor Network (WSN) [23].

Controllers separately apply policies to their own functions and applications, which are completely isolated from each other. However, they both work on the same WMP, thanks to its virtualization capabilities. Virtualization exploits frame classifiers for managing multiple virtual interfaces, as reported in figure 2.2. Multiple operators coexist on the top of the same hardware and provide both network and localization services.

Programmable networks are enabled by an efficient control plane. However, this is not enough to ensure multi-tenancy and multi-application scenarios, therefore programmable network control planes are required. These are obtained through recent extensions of the SDN approach, again using pluggable interfaces.

2.4 Software defined networking and the control plane

Software-defined Network (SDN) was originally proposed in [24] as a paradigm to enable network programmability. It decouples network control (control plane) from packet forwarding (data plane). Software-based SDN controllers provide a logically centralized network intelligence where the underlying network infrastructure is abstracted from applications. The first commercial evolution was OpenFlow, enabling network controllers to remotely program packet forwarding rules.

SDN is based on a 3-tier architecture composed by: applications, one control plane and one data plane. These tiers are respectively separated by northbound and southbound interfaces. The northbound interface is generally implemented as a REpresentational State Transfer (REST) API and permits applications to use control plane functions. The southbound interface provides a well-defined forwarding instruction set to control the behavior of networking devices.

Several controllers have been developed in recent years, all having OpenFlow as the southbound protocol [25, 26, 27, 28, 29]. First generation of network controllers are usable only for programmable forwarding or for a limited set of network functions. They do no take into account the available flexibility coming from the wireless networks both with Software-defined Radio (SDR) and programmable MACs.

Recently, pluggable interfaces has been introduced making controllers able to support multiple southbound control protocols. In addition to OpenFlow, the ONOS controller supports also NetConf [30] whereas Extensible Network Controller (XNC) supports also Cisco OnePK [31]. The main innovation offered by XNC is the introduction of the Service Adaptation Layer (SAL), which separates southbound and northbound protocols, both implemented as service/application plugins.

This pluggable vision is enabled by OpenDaylight, an extensible software architecture for controllers, which is able to support existing and future applications, protocols and interfaces [21]. OpenDaylight is based on OSGi, a Java technology for plug-ins that we choose to enable our modular vision for positioning functions an algorithms, as depicted in figure 2.1. This pluggable architecture helps building primitives that serve not only for controlling network functions but also for inertial sensors and cameras.

2.4.1 Controlling the MAC behavior

Effective indoor localization requires flexibility at MAC layer to guarantee an intelligent sensing including accurate ToA measuring, forging frames to include coordinates and timestamps, etc. This requires both MAC flexibility and a logically centralized controller.

Multiple frameworks for programmable MAC layer have been proposed in literature [15, 32, 33]. However, the lack of effective MAC-layer abstractions and programming languages for medium

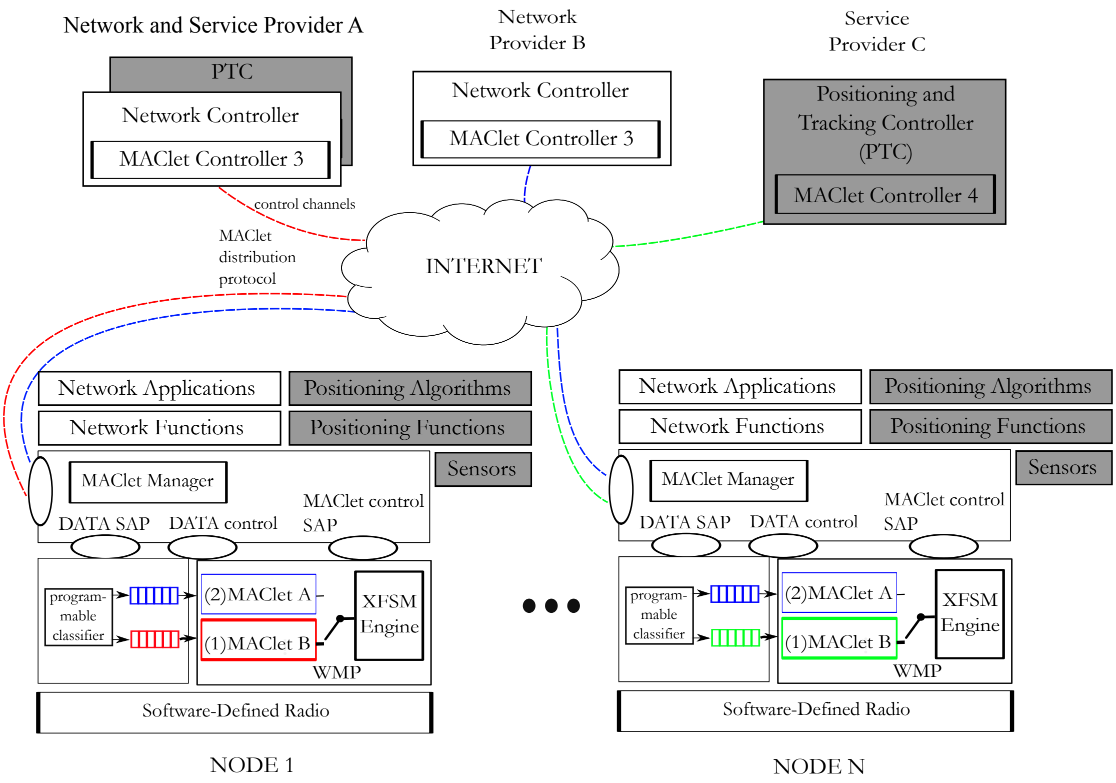


Figure 2.2: Nodes controlled by multiple network and location based service providers.

access rules has obstacled the definition of control frameworks which are dedicated to wireless MAC. Recently, a control architecture for WMP-enabled devices appeared in [34]. This control architecture requires some adaptations to be included into a wider framework for MAC, PHY and applications both related to networking and positioning.

In our vision, MAC controllers are based on the WMP API for collecting channel signals and statistics. They exploits frame classifiers for managing multiple virtual interfaces, which are dynamically configured by positioning and tracking controllers and by network controllers through a unique interface, depicted in the left side of the wireless nodes in fig. 2.1.

The MAC Engine is able to switch from one MAC program to another, multi-threading can be supported by opportunistically programming the switching events (e.g. at regular timer expirations) in the meta state machine. This feature allows to run simultaneously multiple access schemes over the same hardware (as multiple virtual interfaces with different behaviors). A frame classifier is then required for multiplexing the traffic between the available access schemes. The classifier can work on several frame parameters, such as the QoS class, the source and destination MAC addresses, the frame size, the frame type, the events occurring when processing the frame, etc.

The MAClet Manager and MAClet Controller are the main components of the control plane. The MAClet Manager handles MAClets at node-level and provides the node-level intelligence.

It transmits and receives MAClet protocol messages to/from MAClet Controllers and Managers. It upgrades the local repository and loads, runs, configures MAC programs over the WMP. The MAClet Controller provides the network-level intelligence on the basis of low-level data received from MAClet Managers; it commits locally computed best response strategies or those decided by the operator. Different controllers can work simultaneously on the same physical network. Our goals is to extend this concept to the simultaneous control by different *kinds* of controllers. A MAClet implementing Distributed Coordination Function (DCF) for web traffic operated by Internet Service Provider (ISP) A, a Time Division Multiplexing (TDM) MAClet dealing with Voice over Internet Protocol (VoIP) traffic handled by operator B and a IPS MAClet dealing location based services offered by operator C coexist in the same hardware.

Control messages are classified in Management, Action, Information, and Flow Control which are common to network and non-network controllers. Management Messages allow registration of the MAClet Managers to a given controller. Action Messages are used to send, load, activate, configure MAClets and their parameters, Information Messages carry on low-level statistics from managers to controllers, whereas flow control messages are used by the Controller to create, remove, and configure queues. MAClets are stored in MAClet repositories, grouped by kind and final goal.

These new concepts would not be possible without flexible MAC and PHY layers because localization functions need low-level information (RSSI, fine-grained timestamps, CSI, ...) and low-level actions (forge frames, emit pilot signals, change frequency and bandwidths, ...) for localization.

2.4.2 Controlling the PHY behavior

The radio platform acts as sensor and an actuator. Both aspects are used when active localization requires sending frames or piloting signals which are dedicated to localization. As previously stated, the focus on Wi-Fi based indoor localization is motivated by the diffusion of IEEE 802.11 technologies. However, wireless cards and wireless protocols were originally designed to fulfill different goals: improve transmission rates, coverage, coexistence, power reduction, but not with localization purposes. Despite we provide several new localization methodologies, the main goal of the proposed architecture is to take advantage of flexible platforms at MAC and PHY layers, respectively described in [17] and in Appendix A.

CR permits devices to dynamically negotiate spectrum use and to choose appropriate frequencies, protocols and modulations to coexist with other devices. CR gives flexibility of operation that goes way beyond that of SDR. However, since SDR enables wireless devices to switch dynamically between protocols and frequencies, it is a key enabler for cognitive networks. The platform we selected to guarantee flexibility at PHY layer is the Wireless Open-Access Research Platform (WARP) card. The flexibility at PHY layer has been contributed by porting the WMP framework on WARP in order to experimentally validate our approach.

2.4.3 Controlling Functions and Applications

Localization services require both high-level and low-level context awareness. High-level awareness of the context include, for example, the information about user's activity, if he/she is sleeping, showering or watching TV. It is used to adapt to changing contexts seamlessly. On the other hand, low-level context awareness recognizes, for example, how many wireless devices are contending the channel, the kind of signal propagation (e.g. Line of Sight (LoS),near Line of Sight (nLoS),Non Line of Sight (NLoS)), the user's mobility pattern (still, walking, running), etc.

Managing context-aware services and efficiently supports context acquisition, discovery, and reasoning has been enabled by a clear formalization of the context model [22]. The authors present the SOCAM architecture, which provides reasoning about contexts. This permits to deduce high-level contexts from low-level ones, and implicit contexts from explicit ones.

A key enabler for modularity and context-awareness is OSGi, and we use it for controlling functions and applications.

OSGi

Functions and Applications have to be modular, extensible, reusable, maintainable, and adaptable, similarly to what the WMP and MAClets provides at MAC layer. These are implemented as components and services that can be dynamically installed, activated, de-activated, updated and de-installed at execution time.

Components and services act at the application layer and are the basic building blocks of a modular framework. OSGi is a specification whose main goal is to define components and service models for Java [35]. In OSGi terminology software components are called bundles, which are the smallest units of modularity. Bundles are cohesive, self-contained units of code, analogously to what MAClets do at the application layer.

Bundles explicitly define their dependencies to other modules and services and their external API, therefore their API can be finely controlled.

OSGi bundles are jar files with additional meta information, stored in the META-INF/MANIFEST.MF file, which is part of the standard Java specification. This permits backward compatibility because non-OSGi runtimes ignore the OSGi metadata. Therefore OSGi bundles can be used without restrictions in non-OSGi Java environments.

When installed in a OSGi runtime, OSGi bundles have a lifeclycle as depicted in fig. 2.3. During the installation, a bundle receives a unique install ID, its status changes to $INSTALLED$ and it is stored in a local bundle cache[1]. The OSGi runtime then tries to resolve all dependencies of the bundle. If this process succeeds and all are dependencies are resolved, the bundle shifts

[1] this cache can be considered as the corresponding element of a MAClet repository

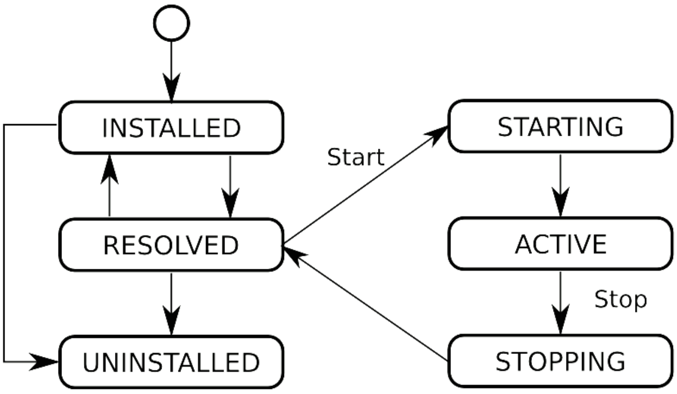


Figure 2.3: OSGi bundle lifecycle.

to the *RESOLVED* status otherwise it persists in the *INSTALLED* status.

If several bundles exist which would satisfy the dependency, then the bundle with the highest version is used. In case these bundles have also the same version ID, then the install ID is used and the lowest one is chosen. The bundle can be started (explicitly or automatically) and its status becomes *STARTING*, then it shifts to *ACTIVE*.

2.4.4 Non-networking service example

Exploiting the same network infrastructure for data transfer and localization permit a joint optimization. In this thesis we do not study networking strategies that get benefits by knowledge about position of nodes. On the other hand, a Wi-Fi network infrastructure helps providing non-networking services able to help context estimation and localization.

In this paragraph we provide an exemplary use case, useful to envision the potentials of the proposed architecture. In large buildings such as schools, universities and hospitals, the Wi-Fi coverage is obtained with multiple overlapping glsplap that are able to listen to each other. A centralized controller sets advanced sensing in turn on each APs for a short time. It permits to reveal mutated propagation conditions due to the presence of moving people. This research branch is named Device-free Passive Localization (DfPL) and uses wireless infrastructures to detect changes in the environment and track the location of obstructing entities [36, 37].

Deploying dedicated listening APs wastes energy and money, but the flexible architecture depicted in fig. 2.1 is able to instruct the WMP to operate standard DCF except for a dedicate time slice during any Target Beacon Transmission Time (TBTT). During this slice, the AP runs advanced sensing, providing measures of the RSSI from any other AP and eventually from any hearable station. Even in case where the number of APs is not sufficient to accurately localize the target, this process helps estimating the context and its changes due to moving people.

This then provides hints choosing the right localization algorithm.

2.4.5 Coexistence

Analogously to the WMP approach, we identified the atomic primitives to get information from the radio card and set configuration parameters, as shown in fig. 2.1, which lists channel usage, channel state, received power, etc. The same approach is applied to ambient and inertial sensors, from which we gain heading information, step detection, 3D orientation of the device, etc.

The decoupling between the platform and its behavioral model is obtained through an open API with primitives able to read operational or configuration data (e.g. get the channel state information, the received power, the Automatic Gain Control (AGC) settings, etc.) and to set configuration parameters (e.g. set TX power, the inter-frame space, the channel bandwidth etc.).

All primitives able to get low-level data from the card are timestamped in order to permit deferred or off-line analysis. Not surprisingly, the low-level primitives defined at MAC layer in [38] are usable also for localization, although location-specific primitives can be defined (e.g. transmit a pilot signal, transmit n frames back to back at regular intervals, etc).

The WMP clearly decouples the role of *manufactures*, which are in charge of providing hardware signals as well as radio primitives and *programmers*, which are free to define the protocol states and relevant transitions. They orchestrate provided primitives according to their desired logic.

Manufactures can differentiate their products by defining and implementing the radio primitives, eventually providing richer API to the programmer, including for example also dedicated actions for positioning (e.g. send a pilot signal). Furthermore, they can supply and support new MAClets (e.g. dedicated MAClets for ToA-based localization).

2.5 Maturity of enabling technologies

Technologies and paradigms considered in this thesis have different levels of maturity and adoption. Emerging technologies are periodically represented in Gartner's hype cycle graphs, which describe their level of maturity. An hype cycle distinguishes five phases of the technological life cycle: technology trigger, peak of inflated expectations, trough of disillusionment, slope of enlightenment and plateau of productivity. Such phases are reported in fig. 2.4 where CR is indicated beyond the peak of inflated expectation and SDR is raising the slope of enlightenment. Indoor localization and Cognitive Positioning Systemss (CPSs) in particular were recently introduced in [39], therefore the indication of location-aware technologies as very mature, has to be referred to outdoor technologies.

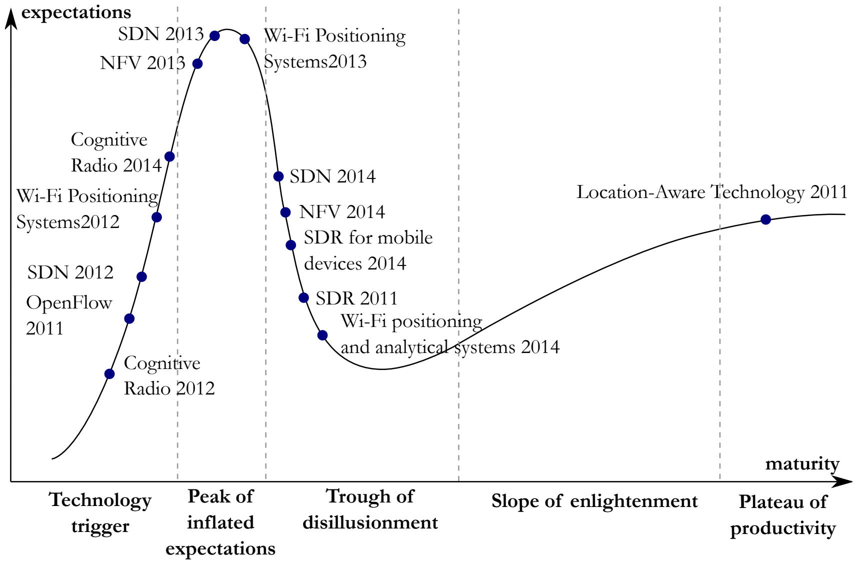


Figure 2.4: The maturity of location-aware technology, CR and SDN paradigms in a Gartner's hype cycle view.

3 MAC-based localization - ToA and RTT

In this section we deal with Time of arrival techniques and we propose the Wireless MAC Processor Positioning System (WMPS). This runs on off-the-shelf 802.11 Access Points and is based on the time-of-flight ranging. We prove, through extensive experiments, that the propagation delays can be measured with the accuracy required by indoor applications despite the different noise components that can affect the result: latencies of the hardware transceivers, multipath, ACK jitters and timer quantization. Key to this solution is the choice of the Wireless MAC Processor architecture, which enables a straightforward implementation of the ranging subsystem directly inside the commercial cards without affecting the basic DCF channel access algorithm.

In addition to the proposed measurement framework, this study developed a simple and effective localization algorithm that can work without requiring any preliminary calibration or device characterization. Finally, the architecture allows the measurement methodology to be adjusted as a function of the network load or propagation environments at the run time, without requiring any firmware update.

3.1 Introduction

The proliferation of location-aware applications is prompting hand-set manufacturers and mobile operators to develop new mechanisms to extend or even outperform the Global Position System (GPS). Assisted-GPS, which is a technique for speeding up the detection of the satellite constellation by retrieving fresh information from 802.11 or 3G infrastructures [40, 41], has paved the way for new systems where the terminal receives so much information from the Network that it can determine its position without the need for GPS. In this context, large players, such as Google and Apple, have recently started mapping Base Station IDs and WiFi network names (beacons) to their corresponding geographical position, allowing localization in urban canyons and in GPS hostile places, such as airports, museums and most indoor environments. Unlike GPS, however, the accuracy of such systems is low [42], and the possibility of refining the positioning using inexpensive and widespread technology, such as

802.11, is very interesting.

Although several commercial systems based on WiFi have been proposed [43, 44], one of the most critical aspects affecting their accuracy depends on the limitations of current implementations in collecting physical parameters, such as the received signal strength or the propagation delay, capabilities that were not included in the original 802.11 Standard. For this reason, recent studies [45, 46] reported ways of improving the localization accuracy by relying on customized measurement instruments (e.g., using FPGA and SDR boards for acquiring the 802.11 signals), and/or complex statistical analysis of long measurement runs [47, 48]. Unfortunately, such customizations offer solutions that are not flexible enough for supporting alternative measurement methodologies that might be required by new applications. This situation can be changed without the need for hardware refinement simply by reconsidering the software that drives these systems. Modern cards are equipped with very powerful logics and sensing mechanisms, and current issues in collecting accurate localization information are due mostly to the interface available at the driver level rather than to the hardware itself. For example, on one side, some hardware signals are not exposed directly to the driver, whereas the access rate to the exposed signals depends on the host operating system. Starting from these considerations, we define a Positioning System that is *flexible* in terms of the measurement methodology and data processing, and can be *deployed* in off-the-shelf network adapters due to the Wireless MAC Processor [38] architecture. This approach to wireless card programmability has recently illustrated how to use programmable state machines profitably to implement complex MAC protocols on top of a MAC Engine, where the underlying machine code implements only basic operations, such as frame transmission and reception. By adding some measurement primitives triggered by specific events to the legacy DCF state machine, this chapter proves the effectiveness of a localization solution based on time-of-flight (i.e. propagation delay) ranging.

The chapter provides a twofold contribution. First, the different noise components of time-of-flight measurements (latencies of the hardware transceivers, multipath, ACK jitters, timer quantization, etc.) and their effects on the ranging accuracy are analyzed in detail. Time-of-flight ranging was also used in reference [49], but the present study is able to perform the measurements by exploiting different triggering events (frame start, frame end, preamble start, etc.) internal to the card and by avoiding preliminary characterization of the devices. Second, an overall localization system was designed, in which an high-level service deployed on the 802.11 infrastructure (namely, the APs) can run-time exploit the low-level measurement functionalities. This chapter also proposes a simple and effective adaptation of the Bancroft's localization scheme [50], which is a well known solution for the localization problem devised to avoid any preliminary calibration of both the anchor nodes and target devices. These results show absolute positioning errors lower than 2.5 m in outdoors (and lower than 5 m in 80% of the tested indoor positions), as well as the capability to track the targets moving at pedestrian speeds.

3.2 Related Work

A number of technologies and approaches have been proposed to solve the critical issues of ranging in indoor environments. Among these, the solutions based on 802.11 appear quite promising because they can rely on pervasive Access Point deployments and user interfaces. Although the initial proposals mainly considered the power received as the main ranging parameter [5], the use of propagation delays was recently proposed [47, 48, 49]. Propagation delays are linearly dependent on the distance, whereas the relationship between the received power and the distance is often quite specific to the propagation environment, thus requiring long calibration phases. On the other hand, propagation delays track the distance of the radio signal, which can be different from the actual distance because of reflections. Moreover, they are normally difficult to measure because off-the-shelf 802.11 cards can only measure the time of arrivals at the driver level with a resolution of 1 μs, which corresponds to a distance quantization error of 300 m (unsuitable for indoor localization). The clocks used in commercial cards are also of limited quality and their clock drifts cause significant errors. Two-way measurements are often used to solve this issue because they do not need to synchronize the clocks (used for timestamps) of independent nodes [51].

In references [47, 48], it was shown that the limited resolution of the delay measurements can be overcome by statistical means, such as stochastic resonance. Unfortunately, statistical techniques require the observation of a large number of samples, which prevent the tracking of moving objects. A different approach was proposed in [45, 46], where dedicated mechanisms for measuring the propagation delay were designed using either external hardware [45] or software-radio [46]. The solutions are quite expensive because these measurements need to be taken at each anchor in an indoor environment. To reduce the number of independent measurements required for localization, [52] proposed to perform indirect measurements of four-way propagation delays. An alternative solution based on cheap Atheros cards was described in [49]. Owing to the availability of a well documented open-source driver, the measurement resolution was improved using the host CPU for opportunistically polling some card registers. In particular, these registers account for the cumulative time intervals in which the medium has been sensed idle and busy, and are updated at the card internal clock, i.e., at 44 MHz. Reference [53] reported that for typical indoor environments, even in presence of a strong multipath (without line of sight propagation), the average error on the propagation delay is 1 clock cycle at most (at 44 MHz). Such an error can be compensated for if multiple reference points with independent propagation conditions are available for trilateration.

Apart from the measurement resolution, another critical aspect of delay-based ranging solutions is the need to estimate and compensate for hardware-dependent measurement delays, which are added to the propagation delays due to clock drifts [54], transceiver latencies and jitters in the ACK frame scheduling, i.e. in the SIFS interval. References [55, 56] and [57] proposed a characterization of the hardware additional delay to identify the fingerprinting of different cards. In reference [49], such a hardware additional delay was assumed to be bi-modal and related to the quality of the received signal. By jointly measuring received power levels and propagation delays, the ranging was then improved by compensating for the estimated hardware delay.

3.3 Ranging

This chapter proposes a solution for performing propagation delay measurements with a resolution of 1/88 MHz using ultra-cheap 802.11 cards. Rather than modifying the card driver and polling the card registers as proposed in reference [49], the solution is completely *internal* to the card, thereby avoiding the uncertain delays introduced by the host operating systems. The approach was enabled by the Wireless MAC Processor architecture recently proposed in [38], whose implementation has also been released in [58] for the Broadcom AirForce54G cards. As described in the next subsection, this architecture can integrate a (programmable) measurement framework to the MAC operations without any firmware update.

3.3.1 Platform

The Wireless MAC Processor (WMP) architecture enables the execution of different MAC schemes, which are specified in terms of programmable state machines, on the same hardware. Indeed, MAC protocols can be described effectively in terms of the state machines provided that a complete list of actions (such as transmit a frame, set a timer, build an header field, switch to a different frequency channel, etc.), events (channel up/down signals, indication of reception of specific frame types, expiration of timers, enqueueing of a new packet, etc.) and verifiable conditions (frame address, medium status, RTS thresholds) are available.

The proof-of-concept of such a vision was obtained by replacing the firmware of a commercial card (for which an open firmware was available) with a new firmware implementing an executor of generic state machines and a core list of events, conditions and actions [38]. Because the events exposed as a programming interface also include the hardware signals involved in a propagation delay measurement, this chapter proposes adding a few more actions (devised to perform ranging estimates) to the MAC state machine. Apart from the solution described in 3.3.2, the approach is intrinsically programmable and several methodologies, such as multiple handshake delays, can be supported by specifying different events for activating and stopping the time measurements.

3.3.2 Methodology

In this solution, the propagation delay measurements are carried out during the medium access operations and can work on normal DATA/ACK frame handshakes between a target station, i.e. a station to be localized, and an anchor station, i.e. a reference station whose location is known.

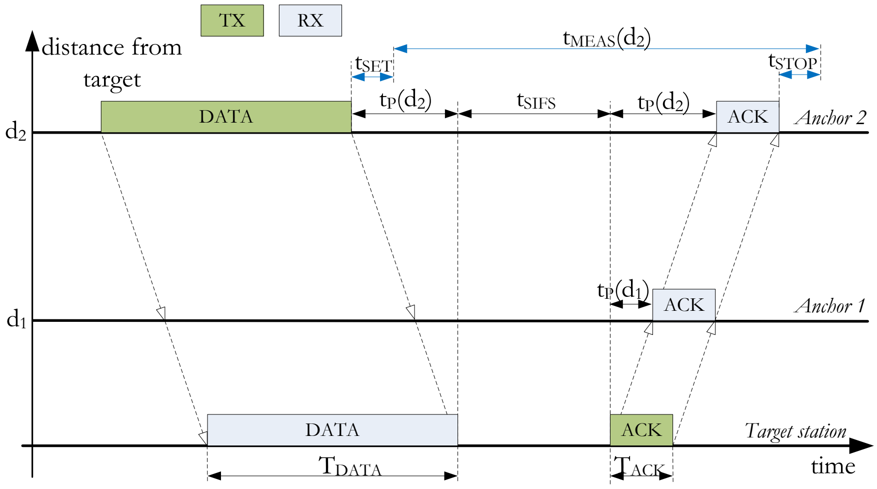


Figure 3.1: Two-way delay measurements: triggering events and timing errors.

Measurement Operations are quite simple: a timer can be activated immediately after a DATA frame transmission and stopped at the start of the ACK reception. Both the MED_START and MED_END events (signaling the start and end of channel activity) are available on the WMP programming interface. Because the MED_END events are captured more precisely than the MED_START events, which might be delayed in the case of low power quality [49], it was decided to modify the previous procedure slightly by stopping the timer at the end of the ACK reception (as shown in figure 3.1). This results in $t_{MEAS}(d) = 2 \cdot t_P(d) + T_{SIFS} + T_{ACK} = q + m \cdot d$, i.e. it is linearly dependent on the radio distance, where $t_P(d)$ is the propagation delay, $m = 2/c$, and c is the speed of the light. Timing errors due to the latency required for activating (t_{SET}) and stopping the timer (t_{STOP}) can be considered as additive noise components.

In principle, the station performing the measurements, i.e. the ranging station, can be either the target or the anchor station, provided that it supports the WMP architecture. This suggests that the ranging station can work on frame handshakes that begin with its own DATA frame transmissions, whereas the other station (sending the ACK replies) can be a legacy station. This type of measurements is called active measurements. As shown in the figure, in this chapter, the delay measurements were performed by the anchor stations to have a system that can work with every (legacy) target station. The only requirement for the target stations is to be associated with the anchors of the ranging system thereby enabling the ACK replies to the DATA frames. The figure also shows that the anchor stations can work in passive mode, as in the case of anchor 1, by activating and stopping the timer during a frame handshake beginning from an another station. This solution allows the measurement overheads to be limited by exploiting traffic frames for ranging or by multiplying the number of independent measurements carried out on a single frame handshake.

Measurement Resolution obviously depends on the clock of the ranging stations. The internal clock of the card used for the WMP implementation was confirmed to works at 88 MHz. The resulting quantization error on $t_{MEAS}(d)$ was 1000/88 ns, which corresponds to a distance of 3.4/2 = 1.7 m at the speed of light. Such an error was considered acceptable for common localization applications. The measurement errors (due to quantization and latencies in the timer management) and physical errors could also be distinguished.

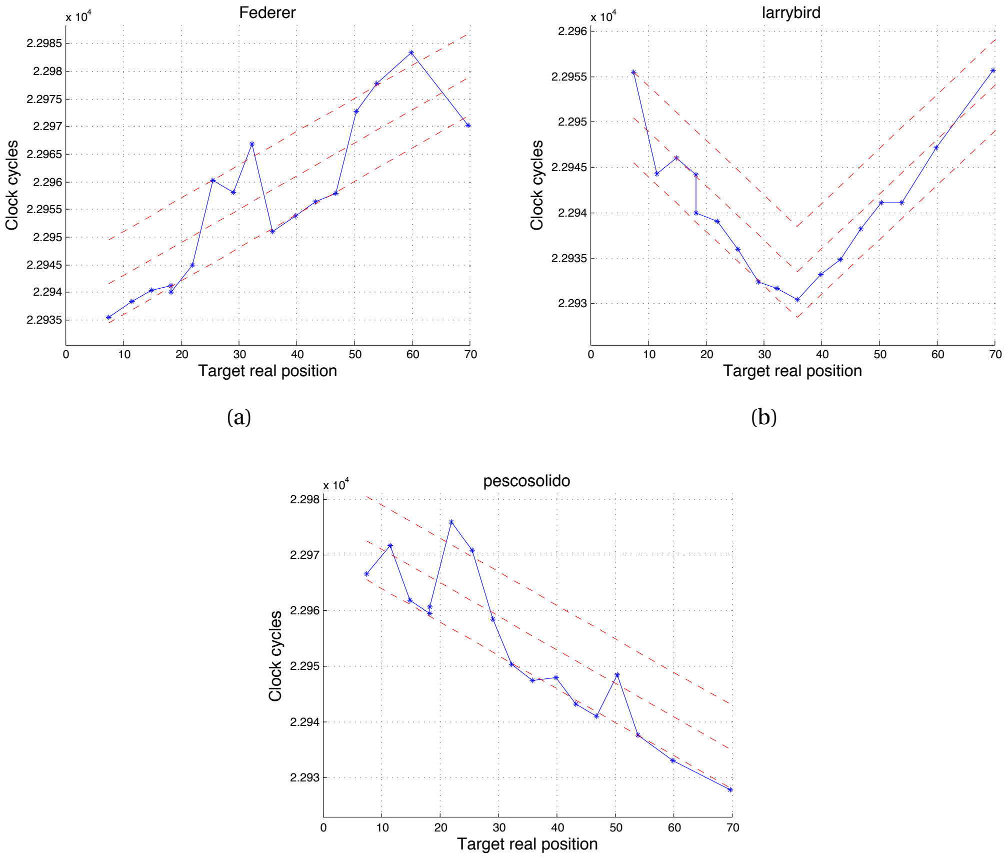
Physical errors are caused by differences in the actual relationship between delay and distance and the expected one. A number of factors can cause such a misalignment. First, because of the multipath, the measured delay can refer to the strongest radio path rather than to the direct one, resulting in a higher estimated distance. Second, because the measurement refers to two-way propagation delays, it is affected by errors on the ACK scheduling time. In fact, the standard defines a tolerance on the SIFS scheduling (up to 1 μs), which is reflected into a measurement error that can be systematic or not for the target stations produced by different vendors[1]. Finally, also for anchor nodes, the physical implementation of the transreceiver introduces some latencies in revealing the MED_START and MED_END events, which might have a random component. These latencies can be higher in passive measurements because in this case, the receiver latency affects both the stopping and starting of the timer.

3.3.3 Experimental results

As a first ranging experiment, three different anchor points (called federer, larrybird and pescosolido) were set up in a long corridor of approximately 70 m at the University of Brescia. Two anchor nodes were placed at the corridor edges, whereas a third is placed precisely in the middle. All the anchors were equipped with an Airforce54G wireless interface running the WMP and loaded with the modified DCF state machine. The target station used the same wireless adaptor but with the original DCF firmware.

Figure 3.2 shows the measurements carried out independently by each anchor node. Each measurement was averaged by considering a large number of measured samples (e.g. 1000 different samples). The delay measurements are expressed in terms of clock cycles, which should be linearly dependent of the radio distance with a slope equal to $2/c \cdot 88\ MHz$=0.5871 clock cycle/m. Indeed, such a linear relationship works well when the target is close to the anchor node (within a distance of approximately 20 m), whereas it can deviate from the linear relationship for larger distances. For example, in figure 3.2-a, it is evident that the delay versus distance relationship switches between two different lines (the bottom one, from 5 m to 20 m and from 35 m to 50 m, and the upper one from 20 m to 35 m and from 50 m to 60 m), which have identical slopes and different constant terms. Such a phenomenon may due to the multipath, because the dominant path of the received signal can change as the target node moves along the corridor. On the other hand, the phenomenon is evident for large distances, where the difference between the direct and reflected path should be, in percentage, less evident. For example, when the target node moves from 32.5 m to 35 m, the federer anchor

[1] Note that systematic errors on the SIFS timings are not a problem because they are eliminated by the localization scheme as a constant additional bias (similar to the clock bias in GPS), whereas SIFS random jitters might originate positioning errors because they vary at each frame handshake started by different anchors.



(a)

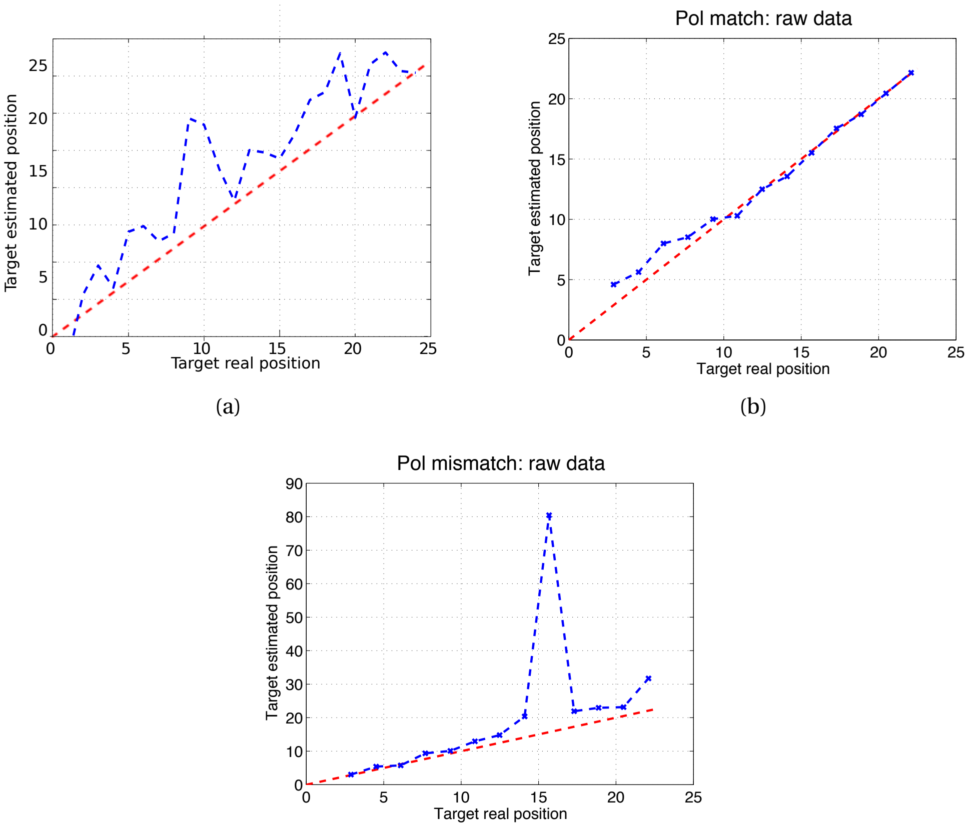
(b)

(c)

Figure 3.2: Delay measurements carried out by three independent anchors while a target station moves along a 70 m corridor.

node measures a delay difference of approximately 18 clock cycles which corresponds to a path difference of approximately 60 m. At a corridor width and height of approximately 5 m, such a difference cannot be justified by a single reflection.

For a better understanding of the multipath effect, some experiments were also carried out outdoors, where it is reasonable to assume that the dominant path is the one in direct line of sight. Figure 3.3 shows the results of different measurement campaigns performed in the outdoor court of the University of Brescia with a single anchor point. The timing measurements were mapped to distance estimates by considering a reference distance of 5 m (for evaluating the q coefficient).The distance range was limited to 25 m, because the phenomena of interest were observed, even in this reduced range. Despite the experiment being performed outdoors, the expected linear relationship between the delays and distance were affected by fluctuations of approximately 6 clock cycles (i.e. 20 m) even at short distances. The same experiment was repeated using a directional Yagi antenna with horizontal polarization for the anchor node and a dipole antenna with the same polarization for the target one. The use of a directive



(a)

(b)

(c)

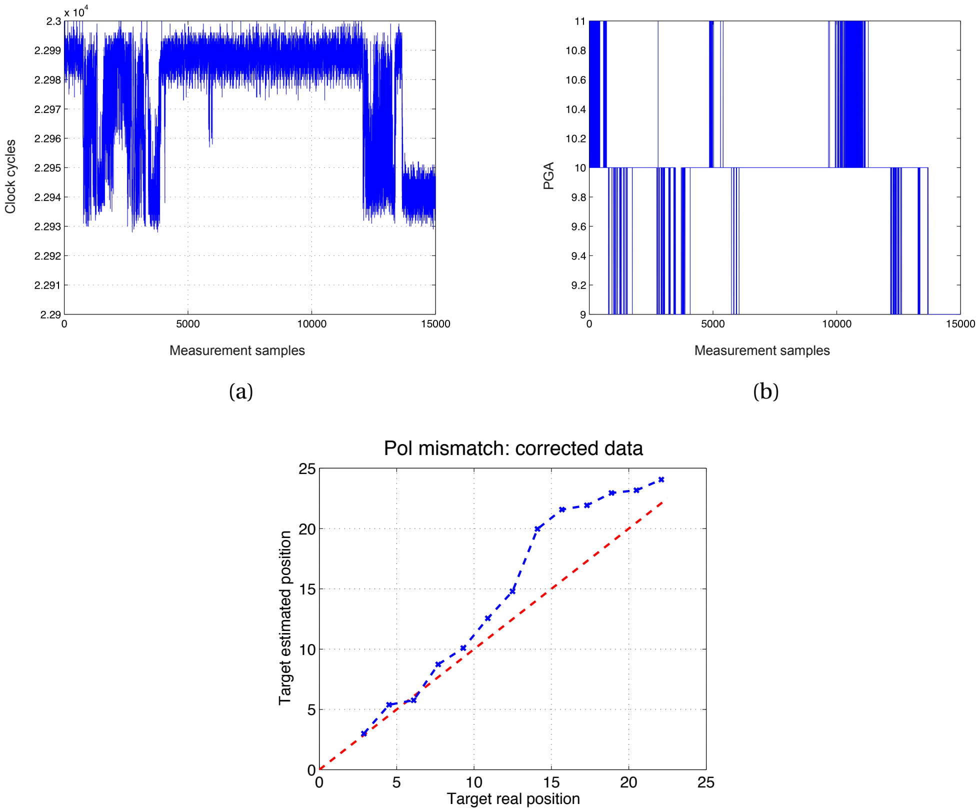
Figure 3.3: Distance estimate in an outdoor scenario under different propagation conditions (omindirectional antenna (a), directional antenna pointed (b) and not pointed (c) to the target.

antenna on the anchor node has an impressive effect in improving the ranging performance, as shown in figure 3.3-b. Such a result might be due to the improved quality of the ACK signal (at the anchor node) that reduces the latency on the MED_END events. On the other hand, in figure 3.3-c, where the Yagi antenna is polarized vertically while maintaining the horizontal orientation of the dipole, the ranging was still better than in the omni-directional case, with the exception of a single point where there was again an error of approximately 60 m.

Based on these results, it was suspected that the main cause of the so called physical errors is the variability of the received signal levels in the subsequent measurement samples rather than their absolute value because such a variability has some effects on adjusting the anchor transceiver.

3.4 Noise Components for ToA at the Receiver

The ranging results presented in the previous section were obtained by considering a long acquisition interval of the measurement samples at each ranging position. The goal to identify



(a)

(b)

(c)

Figure 3.4: Effects of AGC on the two-way delay measurement: temporal trace (a), quantized programmable gain (b), and distance estimate with corrections (c).

the stationary errors by attempting to filter as much as possible the random measurement noise due to quantization and timing inaccuracies. Despite the long measurement runs, systematic errors where found, whose origin did not appear to be related to multi-path propagation. Figure 3.4-a shows a temporal trace of the delay measurements (in clock cycles) for a run of 15000 consecutive handshakes captured at the same distance of 15 m. The figure suggests that apart from the measurement noise, there is a non-stationary noise component which makes the average values oscillate between the two values. A similar phenomenon was also reported in reference [49], and justified in terms of the different latencies of detection mechanisms implemented in an Atheros card for revealing strong and weak packets.

Starting from these considerations, this study conducted further tests to analyze the impact of the latencies introduced by the anchor transceiver, as well as by the jitter of the ACK scheduling times. The characterization of these noise components is important for refining the measurement methodology, providing distance estimates with the desired confidence and tuning opportunistically the observation interval.

3.4.1 AGC

The first element that this study attempted to quantify was the impact of the Automatic Gain Control (AGC) on the detection of measurement events. The AGC adjustments due to sudden variations in the received signal quality can introduce measurement delays. The WMP architecture implemented on the Broadcom open firmware allows the time-varying programmable gain introduced by the receiver amplifier to be read in a 4 bit quantization scale.

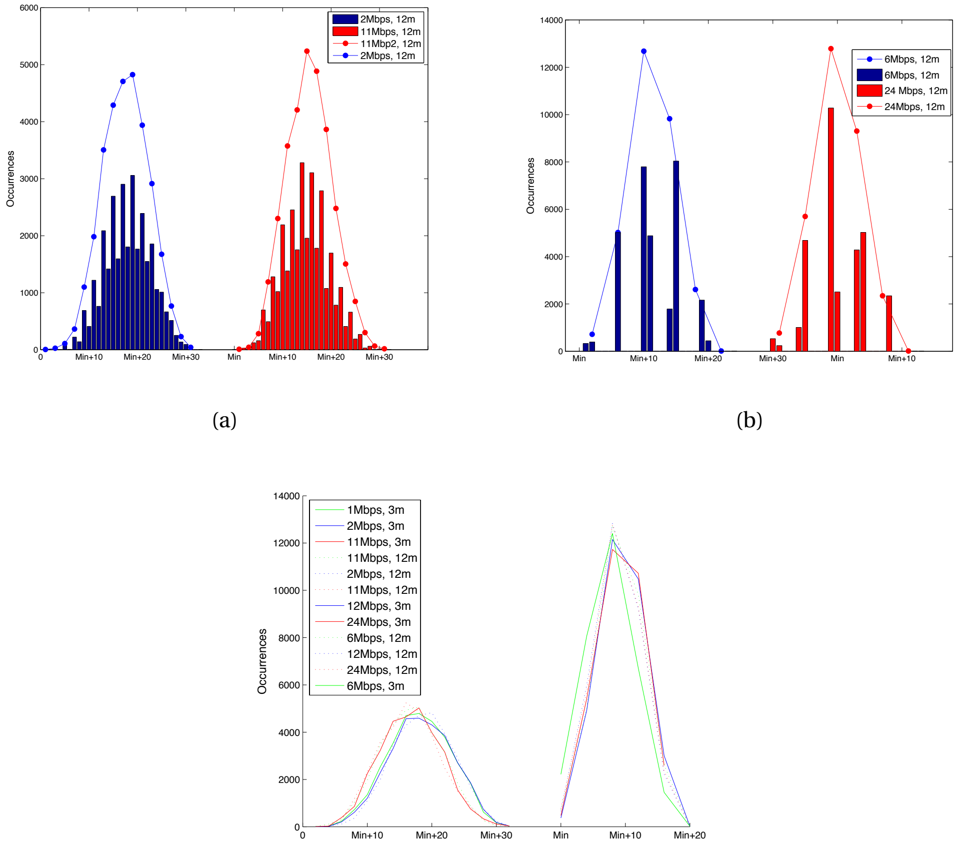
Figure 3.4-b shows the quantized value of the programmable gain amplifier (PGA) for the same experiment plotted in figure 3.4-a. The shortest delay measurements corresponded to a value of 9, whereas the highest delays were obtained when the PGA value was 10 or 11. In several other experiments, it was found consistently that a high value of the PGA always corresponds to additional measurement latency. This latency could be mapped to a significant distance error (up to 60 m in figure 3.3-c) when the expected delay versus distance relationship was tuned for a short reference distance for which the latency was absent. By correlating the delay measurements with the PGA values, it was possible to correct such an additional latency and produce a more reliable distance estimate, as shown in figure 3.4-c.

These considerations also justify our previous findings regarding the impact of directional antennas on the ranging accuracy. Indeed, a directional antenna allows a lower PGA value when correctly pointed, or a consistently high PGA value when non-pointed. Ranging errors emerge only when the distance vs. delay relationship is derived under a given PGA condition and maintained for all values. Although these results were obtained for this reference acquiring card, i.e. the Airforce54G one by Broadcom, it is reasonable to assume that the conclusions can be generalized, i.e. the amplifier latencies depend on the amplification entity and need to be compensated for by producing a reliable ranging mechanism, e.g. by updating periodically the bias of the distance function, as described in 3.5.2.

3.4.2 Transreceiver latencies

Figure 3.4-a shows that for a given latency of the AGC block (e.g. from sample 5000 to sample 10000), there are some other measurement fluctuations. The measurement distributions obtained for a fixed PGA value on a long run of 30000 measurement samples were analyzed to determine which of the random latencies discussed in 3.3.2 is mainly responsible for these fluctuations and understand how to limit or filter such a noise component.

Figure 3.5 summarizes these results for various 802.11b and OFDM modulation schemes while the target was 12 m and 3 m distant from the reference anchor. The ACK frames were sent at the data rate. Rather than plotting the actual clock values, which obviously depend on the employed data rate, to simplify the comparison of the curves, the delay distribution were expressed as a function of the minimum clock value of 99% of the samples. The number of occurrences for each integer value in the range $[Min, Min+\Delta]$ is indicated by the bars in the



(a)

(b)

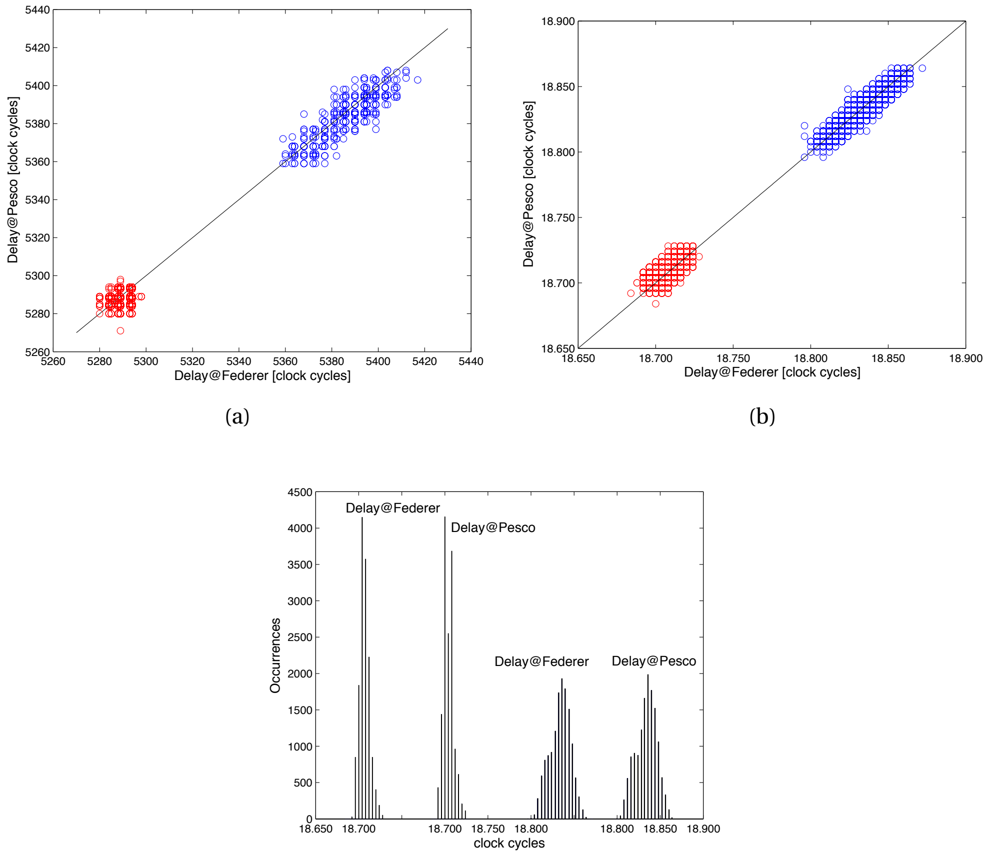
(c)

Figure 3.5: Distributions of the measurement samples collected under 802.11b (a) and OFDM (b) modulation schemes and at different distances (c).

figure, which refer to 2 Mbps and 11 Mbps in (a) and 6 Mbps and 24 Mbps in (b).

The first interesting observation that emerges from the figure involves the quantization and timing errors. Although the timer available on the anchor has a quantization error of 1 clock cycle (1/88MHz), the bars plotted in figure 3.5-a appear to suggest a further uncertainty of an additional cycle, which is likely to be due to the WMP firmware execution (a clock cycle is required to respond to the events and starting or stopping the timer). In other words, a given delay resulting from the propagation delay and transreceiver/ACK latency is mapped into two consecutive bars because of the timer management. A similar phenomenon is presented in figure 3.5-b, where there is another quantization effect of 4 clock cycles (from the start of a group of two bars to the next one) which is justified as a quantization of the transceiver capability to detect the MED_START and MED_END events in the case of OFDM modulations.

The figure also plots the discrete curve obtained by summing the values of the consecutive bars, to allow an easier a comparison between different distributions. Figure 3.5-c shows many



(a)

(b)

(c)

Figure 3.6: Effects of ACK jitters introduced by two different targets (blue points for an iphone, red points for a macbook): correlated delays samples of two independent anchors at 6 Mbps (a) and 11 Mbps (b) and the overall delay distributions (c).

overlapped distributions obtained for different distances and rates. For OFDM modulations, the delay spread of 99% of the samples was lower (approximately 20 clock cycles) than that for the 802.11b (approximately 30 clock cycles). Moreover, the shape of the curve was quite insensitive to the distance and to the specific modulation and coding scheme. These observations suggest that the delay measurements performed on OFDM frames have greater *precision* than the ones performed on 802.11b frames (i.e. they have a lower variability). On the other hand, because of the receiver implementation, the measurements performed on the OFDM packets exhibit a quantization of 4 clock cycles, which generally reduce the measurement *accuracy* (i.e. how the estimated range is close to the actual one).

3.4.3 ACK jitters

The spread of the previous distributions is given by the sum of two independent latencies: the latency due to the anchor transreceiver in revealing the measurement events, and the errors due to the target node in scheduling the ACK frames. Indeed, the standard allows a jitter of 1 μs on SIFS scheduling (i.e. 88 clock cycles), which might completely impair any ranging mechanism based on the propagation delay measurements. To identify the impact of the ACK jitters, an experiment was carried out with two independent anchors performing delay measurements in *passive* mode on the same frame handshakes. The two anchors were placed at a similar distance from the target (actually, there was a 2 m of difference, which is within the clock cycle precision). They registered the sequence number of the data frames for each corresponding measurement sample. The measurements carried out by the two anchors were then correlated based on these sequence numbers. The experiment was then repeated for two different targets (namely, an iphone and a mackbook) and at two different rates (6 Mbps and 11 Mbps).

Figure 3.6 summarizes the results of these experiments. The first two subfigures were obtained by plotting a point cloud, where each point (x_i, y_i), refers to the measurement x_i and y_i performed respectively by the anchor node federer and pesco, respectively, on the same i-th frame handshake. Obviously, in the case of perfect ACK scheduling, a square shaped cloud would expected because for each noise value introduced by the transceiver of a given anchor, the second anchor noise would have been completely independent (between the maximum and minimum most likely values). On the other hand, the figure shows that the two measurements are strongly correlated, i.e. when an anchor produces a given measurement, the one produced by the other anchor spreads on a smaller set of values than the overall possible values. Although the vertical (horizontal) points collected for a given x (y) values can be related to the transceiver noise, the overall spread of the measurement values is related to the ACK jitters. The phenomenon is evident for both the rates (figure 3.6-a and 3.6-b) and targets, with an evident lower accuracy of the iphone (blu points) in ACK scheduling[2].

Although the point spread appears to be much higher than in figure 3.5, a closer look at the number of occurrences shows that most of the measurement samples (90%) fall in a range of 30 clock cycles at 11 Mbps and 20 clock cycles at 6 Mbps. The range is slightly wider for the 99% of the samples, because the passive measurements refer to different events than the active ones. Figure 3.6-c shows the overall distributions of the samples.The two independent anchors (by the same vendor) have almost identical measurement distributions.

3.5 Localization System

The proposed localization system was designed by exploiting the previous findings on the time-of-flight measurement accuracy and precision. The system includes the following com-

[2] For the iphone case, the measurement spread is approximately 70 clock cycles, which is still compatible with the 1μs SIFS accuracy.

ponents: i) the WLAN infrastructure, which is made up of Access Points supporting the WMP paradigm and the (programmable) measurement functionalities, and equipped with directional antennas; ii) the target node, which is equipped with a legacy 802.11 interface; and iii) a localization service running both on the target node and Access Points, for activating, collecting and processing the ranging measurements. Although the measurements are entirely demanded by the WLAN infrastructure, which is also responsible for the simple pre-filtering operations, the localization algorithm runs at the target node. The same data frames sent by the Access Points for measuring the two-way propagation delays can be used to send the previous filtered measurements to the target node.

3.5.1 Measurements

Once the target node is associated to the WLAN infrastructure and begins the localization service, each Access Point in visibility (receiving a signal from the target node at a value higher than a minimum threshold) continuously performs the delay measurements. While the serving Access Point can exploit downlink traffic frames for starting the two-way measurements, the other Access Points necessarily add a traffic overhead for sending their ranging data frames. Although this chapter does not describe the state machine responsible for managing the ranging frames, is is believed that such a mechanism can be modified slightly (due to the WMP flexibility) according to the network load conditions and the number of targets to be localized simultaneously. For example, to prioritize the ranging frames, it is possible to schedule these transmissions with shorter backoff values or to send back-to-back a multiple number of ranging handshakes spaced of a SIFS time. Because of the receiver random latencies, it is required to use a number of measurements between 20 and 100 for averaging before applying the localization algorithm. Because accuracy is more important for this application than precision (being more anchors available for independent ranging), we prefer to use 802.11b modulation schemes for the ranging frames, even though OFDM high rates can reduce the channel resource consumption[3].

For a rough evaluation of the channel resource consumption due to the measurements, the minimum channel time for an handshake performed at 11Mbps with the shortest possible data frame can be considered to be equal to 425 μs, thus resulting in a $0.425 \cdot 20 \cdot 4 = 34\ ms$ minimum channel time (4 being the minimum number of anchor points required by the localization algorithm).

3.5.2 Localization Algorithm

The positioning problem can be faced by the solutions already established for GPS, by considering the peculiarity of this system. For each anchor point a_i, the target node periodically

[3] The delay samples were filtered using a simple moving average filter. Before being passed to the filter, each measurement sample was correlated with the corresponding PGA value, in order to remove the additional AGC delay when necessary.

receives the information t^i_{MEAS} on the estimated two-way delay, whereas the information on the anchor position (x_i, y_i) can be considered to be known to all the nodes because the anchors are static and this information can be broadcast periodically in the beacon frames. The time measurements can be converted to pseudo-distances by considering the expected m slope (i.e. 0.5871 clock cycle/m) of the time vs. distance relationship. Anchor clocks are obviously non synchronized and are generally of a poor quality (typically, with an error of 25 ppm). On the other hand, the lack of synchronization is not a problem for two-way measurements that do not require timestamp comparisons, whereas the different speeds of two anchor clocks correspond to a much lower error than the quantization error for the typical indoor distances.

Basic Scheme. The information relative to each anchor node can be organized in a vector $\mathbf{a_i} = [x_i, y_i, d_i]$, where d_i is the pseudo-distance to the target (i.e. t^i_{MEAS}/m), and the unknown information relative to the target node node in a vector $\mathbf{u} = [x, y, b]$, where b is the fictitious distance bias due to the measurement methodology (i.e. because even with a 0 propagation delay, t_{MEAS} includes the SIFS time, the ACK duration and the latencies introduced by the receiver). With such a formalization, the positioning problem corresponds precisely to the GPS case, provided that the bias b depends on the target only (and not on the anchors). By imposing this on each anchor node, the sum of b and the distance between (x, y) and (x_i, y_i) is equal to d_i, it is possible to find $\mathbf{u}$, without the need for a preliminary evaluation of b. This suggests that there is no need to measure the specific SIFS value of the devices produced by different vendors as considered in reference [49]. An efficient implementation for solving the previous system of equations is to apply the Bancroft's algorithm that transforms the nonlinear problem to a linear algebra problem.

Possible Generalizations. The previous scheme is based on the simplifying assumption that the bias b depends on the target node only. As discussed in 3.3.2, this is true for most delay components included in t_{MEAS}, which contain the ACK duration and the actual SIFS time. On the other hand, the latency introduced by the anchor receiver is only partially corrected by the AGC delay compensation; this latency can vary according to the propagation conditions, resulting anchor-dependent and potentially, for mobile targets, even time-varying. When calibration is not available or not desired, it is possible to apply iteratively some adjustments. For example, the target node can cooperate with the network infrastructure by sending back the results of the previous positioning estimates. Based on these estimates and its own measurements, each anchor can opportunistically translate the delay vs. distance line to compensate for its specific bias.

3.5.3 Experiment Results

Figure 3.7 shows some of results of the localization experiments. The performance of the localization system was analyzed in different (fixed) target locations (a-b) and under target mobility (c). Figure 3.7-a shows the actual coordinates of the target (green points) as well as the estimated ones, with four anchor points located at the vertices of a square in an open court at the University of Brescia. The propagation environment has a few obstacles, whereas the anchor antennas are all oriented towards the internal space. Under these (almost ideal) conditions, 100 different measurement samples were used to produce a single pseudo-distance and the basic localization schemes were run. The figure clearly shows that the positioning works very well in all the different locations. This is quite impressive, considering that no preliminary calibration was performed for the anchors and the target node. Although the figure refers to a target adapter of a given brand, i.e. a Broadcom card, a similar performance was obtained with the adaptors of different brands.

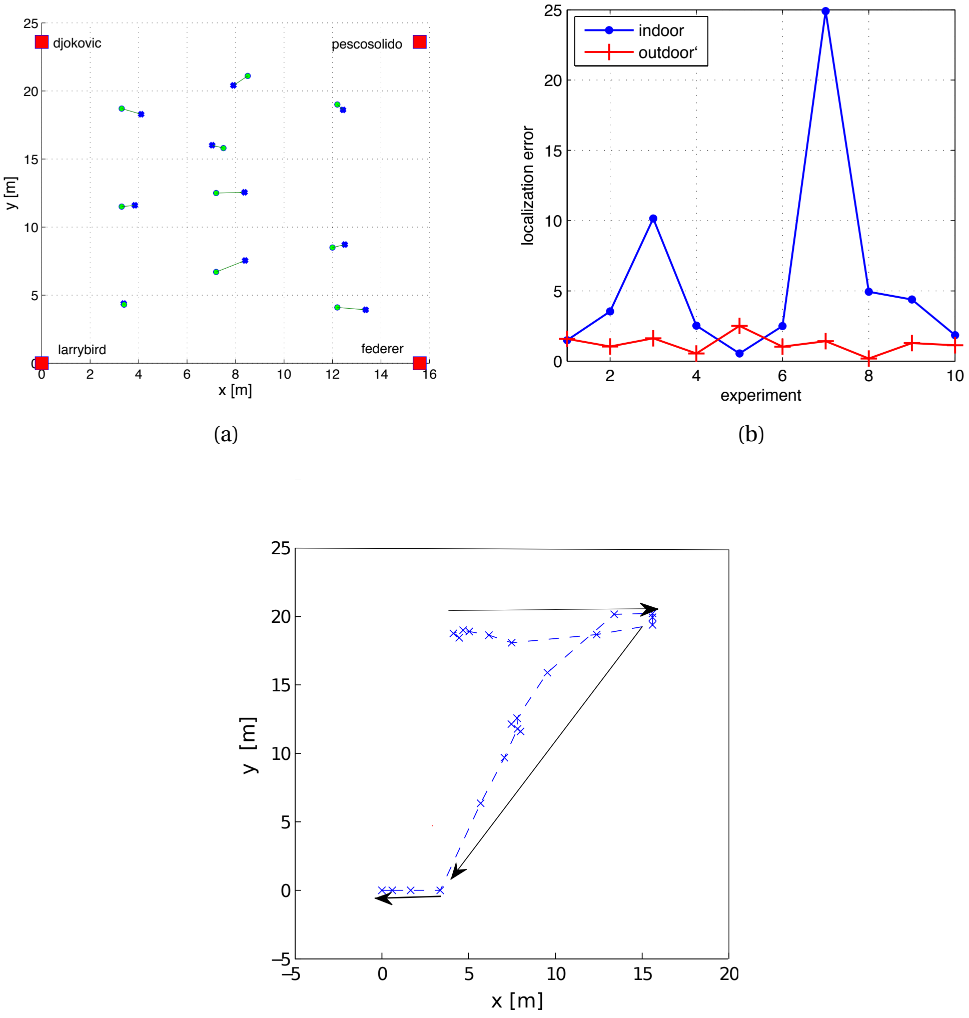
Figure 3.7-b plots the errors (in terms of the distance from the actual position) for each of the positions indicated in the map, as well as for other indoor experiments. For the outdoor experiments, an error $\leq 1.5m$ can be quantified in 9 positions out of 10; it becomes $\leq 3m$ by using a measuring window of 20 samples rather than 100. Indoor experiments were carried out by placing two anchor nodes in a corridor and two other anchor nodes in a perpendicular one, in order to create conditions for path reflections. The basic localization scheme still provided good results (with an error $\leq 5m$) in 8 positions out of 10. On the other hand, there were two cases in which the positioning was poor. This was attributed to the different bias experienced by each anchor that is not addressed by the Bancroft's scheme.

Finally, figure 3.7-c shows a run-time experiment, during which the target position was tracked according to the path shown in the figure. The tracking was based on a measuring window of 100 samples and on the basic localization scheme because the experiment was performed outdoors. The results were considered satisfactory.

3.6 Summary

Current solutions for indoor localization based on 802.11 do not meet the set of conflicting requirements, such as high precision, fast convergence, and minimal environmental calibration. This limits the utilization of WLAN infrastructures to assisting existing navigation systems or providing localization services for indoor areas. Indeed, off-the-shelf devices offer limited access to the low-level hardware signals that can be exploited for ranging, resulting in the need to adopt simple ranging schemes based mainly on the received signal strength indicator, or deploying customized anchor nodes based on dedicated hardware or SDR.

This study examined the potentialities arisen by the availability of advanced API (accessing the hardware signals) for developing localization functionalities on future wireless cards on top of which a localization positioning system can be built. In particular, this chapter discussed how this API can be used to implement a ranging solution based on time-of-flight measurements. After dissecting the benefits and limits of the approach, some interesting conclusions were drawn. The multipath has a negligible impact (in terms of the final localization error) on this type of measurements. The use of directional antennas on the anchors can improve the performance of the ranging system. Higher amplifier gains correspond to higher receiver latencies that can be compensated for by providing reliable delay measurements. Localization experiments exploiting the proposed measurement framework showed absolute errors ≤ 2.5 *m* in an outdoor environment and ≤ 5 *m* in 80% of the indoor positions tested, as well as the ability to track targets moving at pedestrian speeds.



(a)

(b)

(c)

Figure 3.7: Localization experiments: actual position (green points) and estimated ones (blue points) in an open court, errors in different positions for indoor and outdoor experiments (b), dynamic tracking (c).

4 PHY-based flexibility - DToA and bistatic radars

4.1 Introduction

Last decades have been characterised by a huge and still increasing number of location-aware applications, spanning from commerce to e-health [51]. Nowadays, GPS is considered the privileged outdoor localization solution for positioning and navigation, where the receiver computes its position by measuring delays from different satellites. The RADAR (RAdio Detection And Ranging) is mainly used to localize unaware or not-collaborative objects at long distances. Both techniques, originally developed in the military field, are nowadays employed also in civil applications such as navigation, piloting, flight monitoring and control, etc.. Furthermore, both technologies are deployed in outdoor and use time-of-arrival (ToA) measures: one-way for the GPS and round-trip for RADAR. A RADAR localizes objects in polar coordinates ranging them while scanning all possible directions. It scales worse than GPS, as for the number of localized targets, because the computation is centrally done by the RADAR. On the other hand, it does not require any collaboration from the target, that is generally unaware that someone is localizing it[1]. Currently, indoor ranging techniques suffer a trade-off between high accuracy, using dedicated devices, and low accuracy, using legacy ones. On the one hand, RADAR-like solutions use UWB dedicated devices obtaining millimetric accuracy in indoor ranging [59]. On the other hand, legacy 802.11 devices can be ranged with an error lower than 2 m in 90% of the cases, and with a maximum error of 16 m in real-time ranging of a target moving at pedestrian speed [49]. In the present work we provide a novel ranging system based on software defined radio devices, namely WIDAR. The adaptation of selected RADAR solutions to the WiFi field yields an improvement in ranging accuracy. Our system can range off-the-shelf IEEE 802.11b devices in real-time, while operating in legacy networks and without an explicit intervention from the target. WIDAR works also in outdoor, even if, the pervasive indoor deployment of 802.11 access networks and the obtained accuracy with a maximum error of 1.8 m, make it specially valuable for indoor applications. WIDAR is an intermediate solution between two opposites: ranging with dedicated hardware and ranging using only

[1] In legacy RADARs, target can detect impulses; we instead focus on passive RADARS, which are completely non-detectable.

current 802.11 legacy devices. WIDAR performs a passive ranging of existing hardware (WiFi handsets, laptops, etc.); it does not need neither substitution, nor repositioning of the existing access points. Ranging requirements are the same for all technologies [49, 60]: (i) maximum accuracy; (ii) energy efficiency; (iii) minimum packet overhead, (iv) maximum scalability; (v) maximum working range; (vi) low convergence time; (vii) no calibration demanded to the end user; (viii) end-user unawareness. Ranging solutions entail a trade-off among the above conflicting requirements. ToA-based ranging methods share the same basic idea: correlate propagation delays at a known speed with distances. Propagation time measures strongly depend on the triggering events that are chosen to activate/deactivate the stopwatch. As for example, the most intuitive way to measure an inter-packet time is to consider the end of the first packet and the beginning of the second one; we will discuss later on that it is not the best solution. The example above introduces the first problem to solve: the selection of proper events that trigger actions on the stopwatch. These events can be chosen among those available in the MAC/PHY APIs, being the latter more appropriate for fine-grained time measurements. ToA-based ranging accuracy and precision is influenced by: (i) *internal factors* depending on the ranging system and its way to grab time measurements, e.g. choice of triggering events, frequency shift among TX/RX, extra latency introduced by the hardware; (ii) *external environmental factors* such as interferences, multipath and fading.

4.1.1 Justifying the SDR approach

WIDAR employes a USRP2 [9] software defined radio (SDR) platform equipped with the GnuRadio software development toolkit. SDR platforms are generally used by researchers because of their costs and their learning curve. Despite such cons, the SDR choice is justified in WIDAR because of two reasons: (i) money are saved because targets are off-the-shelf devices and a good accuracy is obtained; (ii) one USRP2 is sufficient to range all nodes in its coverage area. These two aspects make the investment affordable. USRP2 permits the application layer to know the instant of detection of the starting frame delimiter (SFD), impossible to be obtained with the monolithic PHY of the current WiFi cards. WiFi commercial receivers are designed neither for ranging nor for localization; time and frequency shifts are compensated with a precision that is enough for demodulation but too rough for ranging. Borrowing some well known tools from the RADAR technology, we are able to refine the estimation of time and frequency offsets by leveraging the USRP2 flexible PHY. The main advantage of using the SDR approach is its measuring instrumentation capability, although it is more than a measuring instrument. A WiFi RF front-end and baseband are engineered to meet the minimum required sensitivity with minimum area/power consumption constraints. The USRP can represent the signal in a predetermined bandwidth via its complex envelope and with an high dynamic range (more than 12 bits per sample for the USRP2).

4.2 Related work

In the present section we focus on time-of-arrival based solutions in 802.11 networks. Active and passive ranging approaches are described in [55] for RSSI fingerprinting; those considerations can be applied also to ToA-based ranging. Active ranging technologies introduce dedicated transmissions/packets for ranging purposes. This approach is potentially detectable by the target node and, specifically in 802.11 systems, it consumes airtime otherwise used for data transmissions over the shared channel. Ranging passively is not detectable by the target, the counterpart is that quasi-silent nodes cannot be ranged. In [47, 61, 49], ranging is performed by measuring propagation delay at MAC level. Propagation is affected by multi-path reflections, they influence ranging accuracy and precision. In [51], multi-path effects were combated with diversity performing antenna or frequency switching. Spacial diversity is considered in case of moving targets, thanks to their motion model. In [51] an analogy between a ToA-based ranging system and a RADAR is drawn. The authors describe how the SIFS interval is not deterministic and they introduce a mechanism to compute the mis-synchronization time among independent unsynchronized nodes. They also introduce a timestamp in their packets, and modify the transceiver of the WiFi card in order to receive also while transmitting (bypassing the low-noise amplifiers). The authors used a testing device from Intel with customized PHY and firmware. In [45], the author classifies ranging techniques with hardware enhancements and purely software ones. He designed an external hardware to improve resolution in measuring the propagation delay, using RTT measurements in order to avoid the need for time synchronization. ToA measurements at MAC level can be obtained using the flexible MAC programmability of the Wireless MAC Processor proposed in [38, 62], but no flexibility on the PHY is provided. Software Defined Radio approaches are described in [46], using 5.8 GHz ISM band. The authors measure both the amplitude and phase of the channel frequency response and the ideal time of arrival for the direct path signal. Multi-path components are recognized via complex sinusoids appearing in the channel frequency response. In the project report [63], the authors use the USRP2 in order to build a localization system using specially configured USRP2 transmitters and receivers, so they can range only special targets, not off-the-shelf ones. In [47] it is proposed a ranging solution based on commercial Atheros cards, equipped with a 44 MHz internal clock. The implementation, based on the open-source driver, polls card registers at regular time intervals. Reading the cumulative durations in which the medium has been sensed idle and busy, the authors compute the time of flight of packets. [64] proposed an hybrid solution that using both angle of arrival and ranging. Having 5 base stations they obtained 3 m accuracy in 50% of cases. In [65], the authors analyze the impact of the IEEE 802.11v standard, lately included in [66], on TOA-based positioning systems. The authors compare a commonly adopted RTT TOA-based positioning in two conditions: with and without incorporating the IEEE 802.11v capabilities. Since authentication and association are no more necessary, scalability lightly improves. The novel processing time computation performed at the AP does not improves ranging accuracy but eliminates the need for manual pre-calibration. Further enhancements are expected to come thanks to the timing measurements mechanism.

Algorithm 1 Ranging algorithm

```
while true do
    TOBERANGED ← load list of targets MAC address
    while NOT RECORDTIMEOUT do
        record trace
    end while
    for frame in trace do
        if (frame is DATAFRAME) AND (MACSRC OR MACDST is in TOBERANGED) then
            compensate frequency offset
            detect start of frame delimiter
            compute frame length
        end if
        if nextframe is ACK then
            frequency offset
            detect start of frame delimiter
        end if
    end for
    compute A to B and B to A ranges
    evaluate target possible positions
end while
```

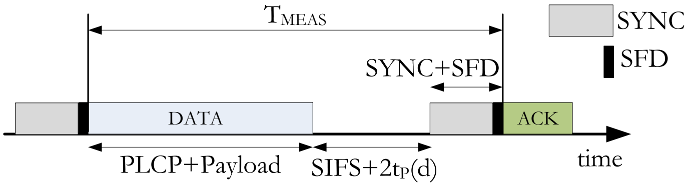


Figure 4.1: Reference points used to activate and deactivate the stopwatch.

4.3 WIDAR ToA based ranging

Ranging can be performed using the propagation time which depends on the measured round trip time (RTT). It is the time elapsed between the transmission of a DATA frame and the consequent reception of the ACK frame. Accordingly to the IEEE 802.11 standard, if a station receives a DATA frame it has to reply with an ACK in a short inter-frame space (SIFS). By considering a couple of DATA/ACK frames, $RTT = 2 \cdot t_p + SIFS$, where SIFS is computed from the end of the last symbol of the DATA frame to the beginning of the first symbol of the preamble of the ACK frame, as seen at the air interface [66]. Nominal SIFS duration is 10 μs long, with allowed variations below ±10% of $aSlotTime$ for the PHY in use. For 802.11b, aSlotTime is 20 μs; it means an allowed SIFS range of ±2 μs which results, by multiplying for the speed of light, in an ugly ranging precision of ±600 m. Our SIFS measurements, taken from cards from different vendors, show that SIFS variance is much lower than the value allowed by the standard, although its shape and variance depend on manufacturer. As example, in fig. 4.2, a SIFS delay distribution is gathered from Broadcom cards. WIDAR operates ranging of targets in an endless loop, as reported in algorithm 1.

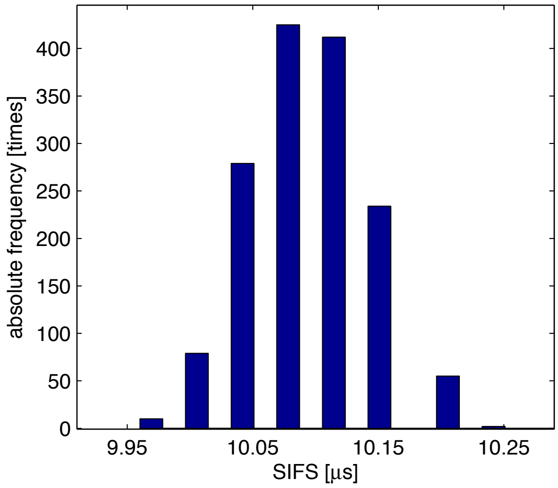


Figure 4.2: SIFS distribution

4.3.1 Methodology

Triggering events

Although the most intuitive manner to determine the propagation time t_p is to have a direct measure of the RTT, we found this not the best way to do it. Time measurements from the end of a frame to the beginning of the next one are affected by uncertainty on *transmit power-on ramp* and *transmit power-down ramp*. Frame timing cannot be taken from its power envelope because the standard gives only maximum duration for ramps: 2 μs in rising/falling between 10% and 90% of maximum TX power [66]. To reduce the uncertainty in determining trigger events, we decided to use the Start of Frame Delimiter, as shown in fig. 4.1, instead of frame edges. The propagation time is then computed from this formula (in case of long preamble): $t_{MEAS}(d) = t_{PLCP} + t_{PAYLOAD} + SIFS + 2 \cdot t_p + t_{SYNC} + t_{SFD}$. PLCP preamble is made by 128 bit (SYNC) + 16 bit (SFD) and PLCP header is 48 bit long which sum 192 bit that means 192 μs at basic rate.

SFD and frame edge detection

Abandoning the edges of frames as triggering events comes with a cost; start/end of a frame are MAC events that are signaled by MAC implementation while SFD detection needs PHY flexibility, currently obtainable only with the SDR approach. To detect the SFD we use a well known method borrowed from the RADAR technology: the matched filter [67]. It is used to correlate a known signal, or template (in our case the SFD sequence), with an unknown signal (the received samples) to detect the presence of the template in the unknown signal. The correlation between the SFD and the received sequence cannot be done 'as they are', because 802.11b uses Direct Sequence Spread Spectrum (DSSS). It means that sequences are spread with an 11-chip Barker sequence, hence the received sequence has to be correlated

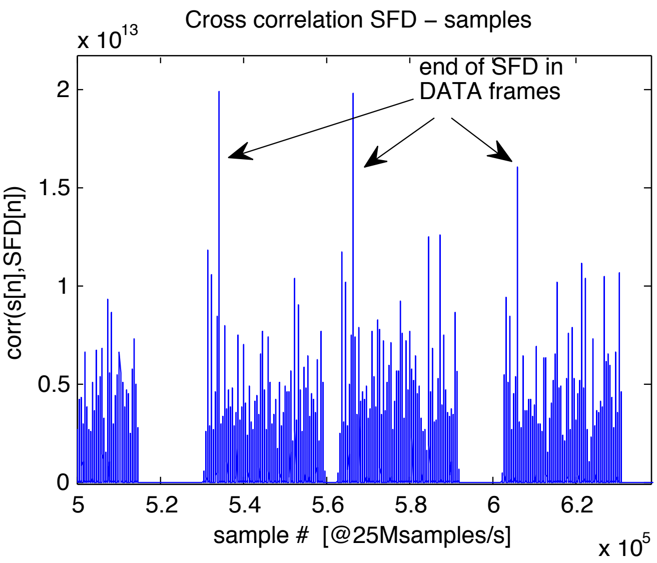


Figure 4.3: Cross-correlation peaks (received samples - SFD sequence).

with the SFD sequence after it is spread with the Barker code and resampled at 25 $Mbit/s$. This correlation will provide several peaks, because the Barker sequence will be recognized as many times as the number of bits in the preamble. The highest peak will delimit the end of the SFD because it is the case where the whole SFD matches. In fig. 4.3 is shown correlation between SFD sequence, spread with Barker code, and sequence captured by WIDAR. The end of the SFD of DATA frames is pointed by the highest correlation peak in each block, as pointed by arrows on the figure. Each block of correlation peaks represents a DATA/ACK couple. Blue blocks are separated by the backoff, since the transmitting station is competing for using the channel. To distinguish the SFD of ACKs in fig. 4.3, the frequency shift between ACK sender and WIDAR have to be compensated.

Bistatic ranging

As shown in fig. 4.4, our system has a strong analogy with bistatic radars. Bistatic radars are made by transmitter and receiver which are separated by a distance that is comparable to the expected ranging distance. This kind of radars use the target as a mirror that reflects electromagnetic waves. On the contrary, in 802.11 systems, the node under ranging (the STA), alternatively sends DATA and receives ACK, (as depicted in fig. 4.4-(a)), then receives DATA and sends ACK (as in fig. 4.4-(b)). Single arrows represent propagation of DATA packets, while double arrows indicate ACKs. Fig. 4.5 depicts, in space and time, the topology described in fig. 4.4-(a) and (b) respectively. Also the single/double arrow convention for DATA/ACK is adopted. Horizontal lines represent the position of AP, STA, and WIDAR. In order to squeeze on a single axis information of the triangular topology, the AP appears represented by two horizontal lines. The lower line defines the AP position considering its distance from WIDAR; the upper line describes the AP considering its distance from the STA. When the AP communicates with

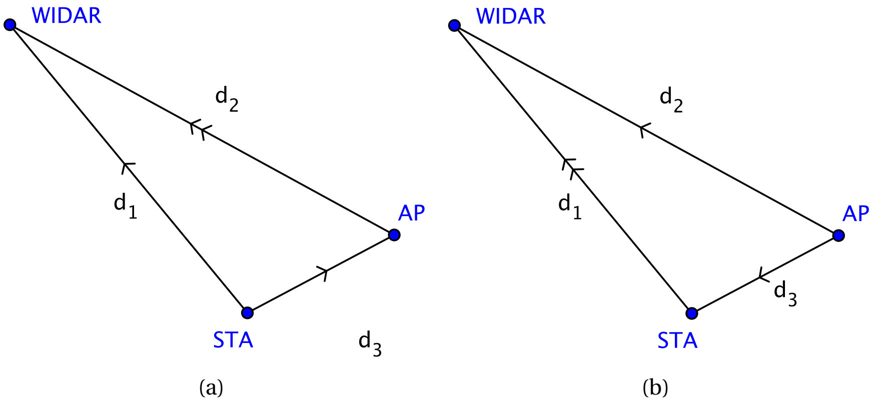


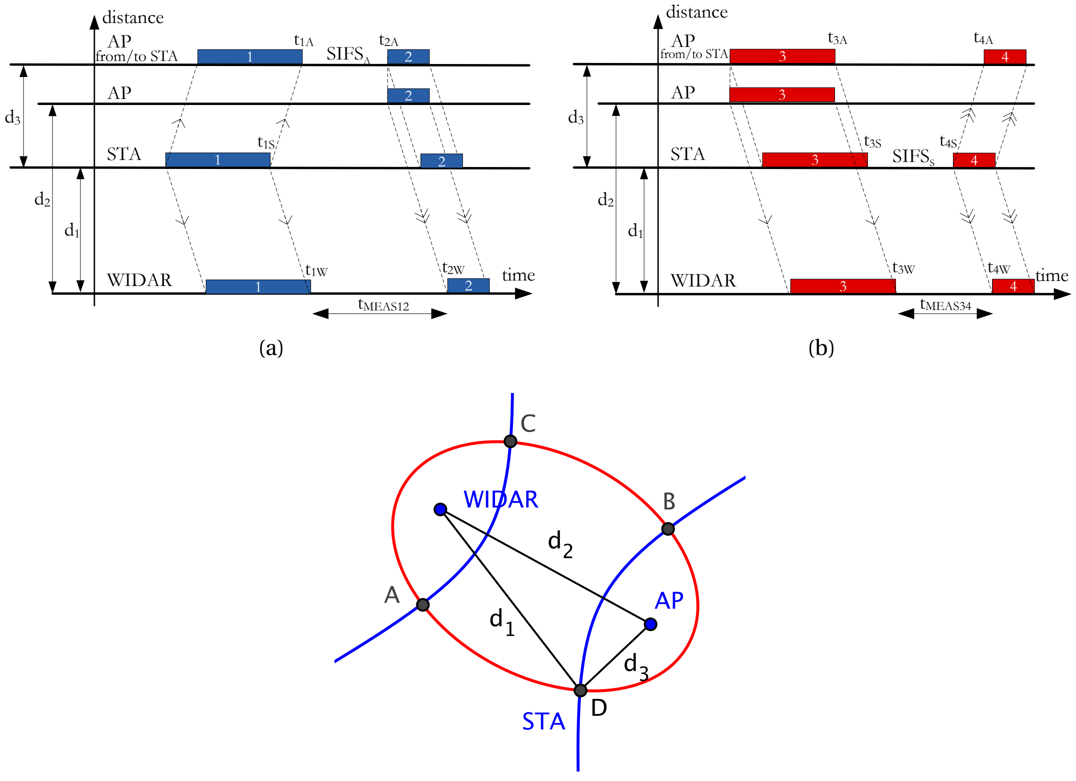
Figure 4.4: Reciprocal bistatic ranging topology: STA transmits data to the AP (a) the AP transmits data to the STA (b).

STA, the higher line is considered, otherwise the lower line is used (propagation till WIDAR). Such representation clarifies the reciprocal distances, with no impact on timing computation. The picture confirms that WIDAR acts passively: only entering arrows towards WIDAR are depicted. All dashed lines have the same slope, that represents the speed of light. 4.5-(a) shows two frames: 1 is a DATA frame sent by the STA, 2 is and ACK frame sent by the AP. They both are listened by the WIDAR, which evaluates t_{MEAS12}. Key time values are indicated with a naming convention where first subscript is tied to the frame id (1 or 2) and the second one represents the node, e.g. t_{1A} delineates the end of frame 1 as seen by the AP, t_{1S} regards frame 1 as seen by the STA, t_{1W} represent the same event as seen by WIDAR. t_{MEAS12} is the time interval between DATA and ACK, as measured in WIDAR. We recall the use of SFD as triggering event, however the begin and the end of frames is computed by considering headers and payload duration. With $t(d_i)$ we indicate the time spent by the electromagnetic wave to propagate along distance d_i with $i = 1,2,3$, therefore $t(d_i) = d_i/c$. Furthermore, distances and consequently times are subject to $d_i < d_j + d_k$ due to the triangle inequality. Furthermore, we use different colors, blue to indicate DATA sent from STA to AP (fig. 4.5-(a)) and red to indicate DATA sent from AP to STA (fig. 4.5-(b)). The same colors are used to draw the ranging loci obtained by the corresponding equations (fig. 4.5-(c)). Looking at fig. 4.5-(a), we can write the following system:

$$\begin{cases} t_{MEAS12} = t_{2W} - t_{1W} \\ t_{2W} = t_{2A} + t(d_2) \\ t_{2A} = t_{1A} + SIFS_A \\ t_{1A} = t_{1S} + t(d_3) \\ t_{1W} = t_{1S} + t(d_1) \end{cases}$$

from which derives:

$$t_{MEAS12} = t(d_3) + SIFS_A + t(d_2) - t(d_1) \tag{4.1}$$



(a)

(b)

(c)

Figure 4.5: Bistatic ranging in time and space: STA sends DATA to the AP (a) the AP sends DATA to the STA (b), ranging loci defined by equations (c).

Since $t(d_2)$ and SIFS are known, t_{MEAS12} is measured, so eq. 4.1 can be written as:

$$t(d_3) - t(d_1) = \alpha \tag{4.2}$$

where α is a known constant. Eq. 4.2 represents an hyperbola having foci in WIDAR and the AP, whose positions are known. By considering fig. 4.5-(b). we can derive the following system:

$$\begin{cases} t_{MEAS34} = t_{4W} - t_{3W} \\ t_{4W} = t_{4S} + t(d_1) \\ t_{4S} = t_{3S} + SIFS_S \\ t_{3S} = t_{3A} + t(d_3) \\ t_{3W} = t_{3A} + t(d_2) \end{cases}$$

from which derives:

$$t_{MEAS34} = t(d_3) + t(d_1) - t(d_2) + SIFS_S \tag{4.3}$$

Here, as before, $t(d_2)$ and SIFS are known, t_{MEAS34} is measured, hence we obtain:

$$t(d_3) + t(d_1) = \beta \tag{4.4}$$

where β is a known constant, whose value is the bistatic range. Eq. 4.4 represents an ellipses having foci in WIDAR and the AP, whose positions are known. Bistatic range β corresponds to the length of the major axis of the ellipse. Loci defined by eq. 4.2 and 4.4 are painted in fig. 4.5-(c), which can be read as follows: given the positions of the AP and the WIDAR, the STA lays on the ellipses whose foci are the AP and the WIDAR and contemporary lays on the hyperbola with the same foci. Reciprocal bistatic ranging has a twofold pro: (i) introduces path 'diversity', useful to combat multi-path effects (ii) defines an ellipses and an hyperbola that intercept in four points. The STA location is one of these points so WIDAR provides an enhanced ranging: it provides a quasi-localization, i.e. the target can be in one of these four possible points. Special cases are when the loci becomes degenerates (the ellipses in a segment and the hyperbola in two rays). A single AP has usually N associated STAs; in this case WIDAR acts as a multi-static ranging system, providing an ellipse and a hyperbola for each STA. However, in our implemented algorithm, we perform an easier interception between circles.

Frequency offset

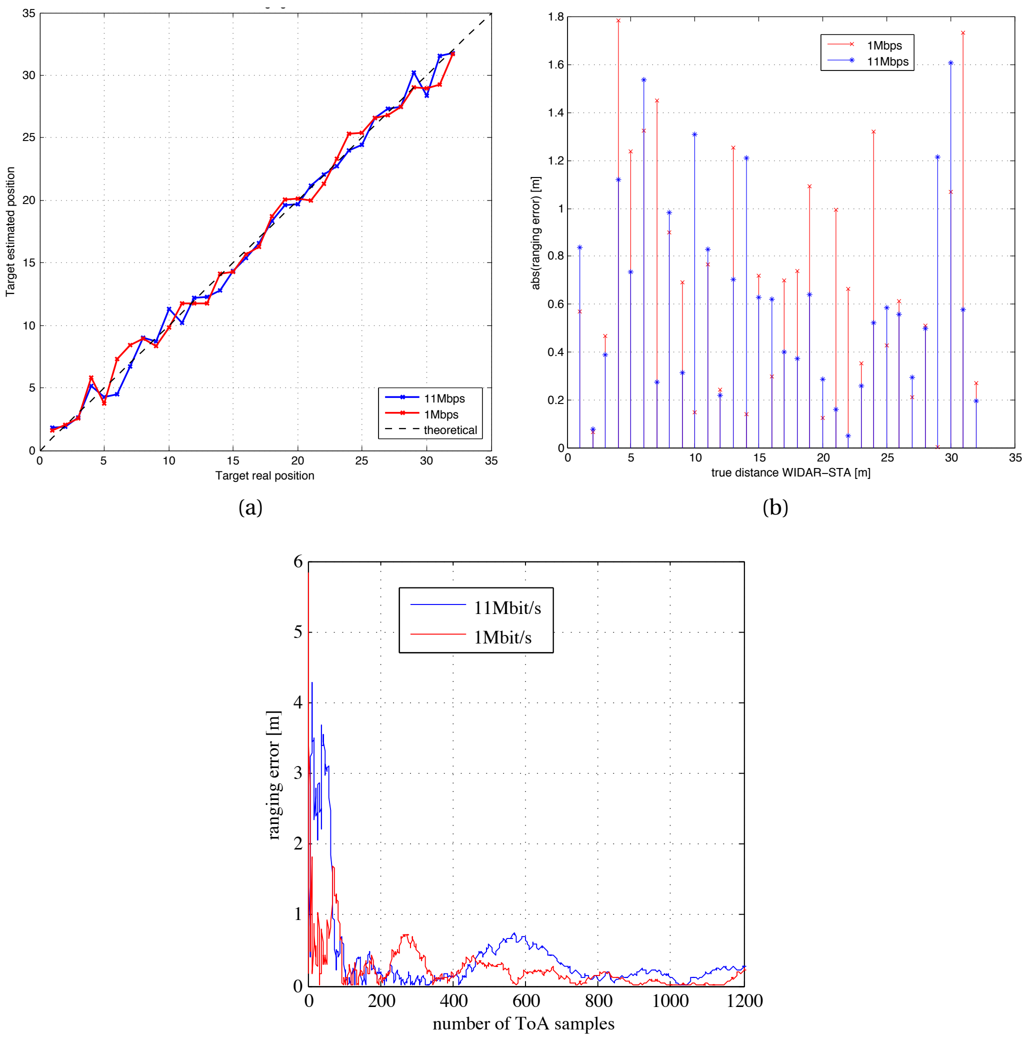
The WIDAR has a frequency offset towards both the STA and the AP, due to the low quality quartz oscillators in wireless cards. Their frequency tolerance is about 10 ppm, meaning that the frequency offset can be dozens of kHz. In order to have a successful communication, the offset must be roughly compensated for; to have an accurate ranging, the frequency offset has to be finely corrected. In [63], the authors use USRP both as ranging device and target ones, so they transmit a signal at a known frequency from a device and analyze the FFT in the other one, to evaluate the offset. We cannot apply this method because we range legacy 802.11 nodes, so we compute frequency offset using the ambiguity function, a standard mathematical tool in RADAR defined as $\chi(\tau, f_d) = \int_{-\infty}^{+\infty} s(t)s^*(t-\tau)e^{-j2\pi f_d t}dt$ [68]. The ambiguity function permits a joint estimation of time and frequency offsets. It is generally used to evaluate the Doppler frequency shift due to relative motion among nodes; in indoor scenarios, at pedestrian speed, the Doppler shift is negligible so the ambiguity function helps in evaluating receiver tuning offset.

4.4 Testbed setup

WIDAR is composed by a USRP2 platform which includes a Gigabit Ethernet interface, a Xilinx Spartan FPGA and RF transceiver, two input channels and two output channels. It receives I/Q samples from the ADC, at a sampling rate that we fixed at 25 MS/s. The maximum sampling rate obtainable from the USRP2 is 50 MS/s, however we opted to sample at 25 MS/s because the dynamic range at 50 MS/s is very small, about six bits per sample, so the choice of amplifier gain becomes very critical. The host is equipped with GNU Radio 3.6, and UHD 3.4 running on Linux. Using the USRP2, computation on samples are done in a regular PC, being possible to elaborate at any OSI level, PHY included. The STA is a notebook running Linux 2.6; the wireless card is an Intel WiFi Link 5100. The AP is a PCEngine Alix2 vers. 0.99h including a Broadcom B4318 card running Linux 2.6. Ranging tests have been performed both in the corridor on third floor of the authors' department and on the adjacent terrace. In both cases the radio environment has revealed crowded and noisy and some metallic shelves where positioned along the walls.

4.5 Experimental results

In this section we report experimental results for both ranging (fig. 4.6-(a-b-c)) and localization (fig. 4.7). To validate the true WIDAR potentials in ranging, we employed a scenario where it was positioned close to the transmitter (both fixed) measuring round-trip time of flight towards the STA. The distance between the station and the AP spans from 1 to 32 m along the same direction (1-D localization). Results are shown in fig. 4.6-(a), where the true position of the target is compared with the estimated one. Points fit extremely well the first theoretical line. At 11 $Mbps$ the error keeps lower than 1 m for 26 times out of 30, and 23 times out of 30 at 1Mbps, as shown in fig. 4.6-(b). Ranging distances are obtained with a sampling rate at the USRP2 of 25 MS/s, which results in 40 ns sampling period. The reason why we obtain better-than-nominal accuracy lays on interpolation; in facts, at 25 MS/s the WiFi signal can be fully represented. To evaluate the time needed by WIDAR in order to estimate a single position of a WiFi node, we show the ranging error vs the number of computed samples. From fig. 4.6-(c), it appears that for more than 100 samples, the error keeps lower than 1 m. Quasi-localization brings to the position estimation in two possible locations, as explained in Sect. 4.3.1. The two possible points present a reflection symmetry along the segment WIDAR-AP. In fig. 4.7 it is shown the topology used for the quasi-localization testbed; the blue quads indicate the true station positions, while the blue crosses depict the estimated ones. Green arrows represent the quasi-localization estimation error. The position of the AP is represented by the red circle, while WIDAR is identified by a red square marker. For sake of figure readability, only one estimated point is shown for each position (the closest one). It is evident that because of topological constraints (walls, floor delimitation, etc.), only a single estimation, of the possible two, can be taken into account. In the present chapter we do not claim to provide any contribution on the localization algorithm, in fact we use a legacy trilateration algorithm. Although localization results can be improved by considering multiple WIDARs acting cooperatively and by applying optimized localization algorithms, the main strength of WIDAR is the ability to localize nodes using a *single device* and without focusing on localization algorithms. Figure 4.7 shows that WIDAR is able to detect the station position without the use of multiple anchors. The use of a single localizing device comes with a side-effect on accuracy, being higher than the accuracy obtained with localization systems based on multiple anchors.



(a)

(b)

(c)

Figure 4.6: Estimated distances over true distances at different rates (a); ranging error by distance (b); ranging error vs number of samples (c).

4.6 Summary

WIDAR leverages both MAC and PHY peculiarities of the 802.11b standard and employs RADAR-specific tools for precise frame timing. This chapter introduces the use of a bidirectional and bistatic ranging technique; it allows a quasi-localization using a single WIDAR device. As a future work, the system can be expanded taking into account not only the DATA/ACK couple but also the RTS/CTS one and the presence of multiple WIDARs will be evaluated, as well as the effects of NLOS. The limitation to 802.11b can be lifted, our approach can be easily generalized to 802.11g signals. Furthermore, the effects of a precise localization will be evaluated to increase the awareness of cognitive wireless networks.

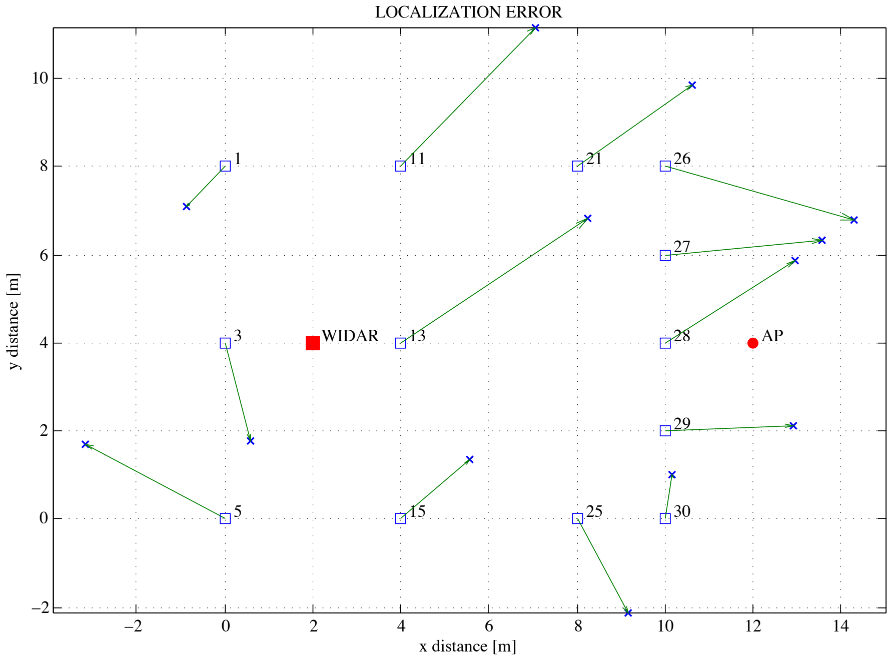


Figure 4.7: Position estimation error using WIDAR with 802.11b at 2 $Mbps$ rate.

5 Human-assisted localization - AoA

5.1 Introduction

In this chapter we present a human-assisted localization methodology based on angle of arrival [69]. This focus is motivated by the recent proliferation of location-aware services, boosted by the pervasive diffusion of powerful sensors and computation capabilities on common and cheap smartphone devices. Positioning services spread over a wide spectrum of applications including navigation, safety, tracking, and socialization, both indoors and outdoors.

The Global Positioning System (GPS) provides an effective and accurate localization in outdoor but is not reliable or even usable in urban canyons, indoor environments, and underground spaces. Deployment costs play an important role in the evaluation of indoor positioning systems (IPS), therefore those that do not require dedicated instrumented environments are preferable.

A large set of IPS exploits the capillary diffusion of WiFi networks. However, the wavelengths used in WiFi are comparable to typical dimensions of indoor obstacles, this exacerbates multipath and fast fading phenomena and leads to poor accuracy. The intrinsic challenges of WiFi-based localization techniques (channel dynamics, multipath, fast fading, etc.) have been faced by the research community that exploits different physical phenomena and technical solutions. Several indicators have been used on WiFi signals to infer locations, including time of arrival (ToA), differential time of arrival (DToA), angle of arrival (AoA) and received signal strength indication (RSSI).

Depending on the sensing and computing devices IPSs are distinguished in network-based and handset-based. They have different features, requirements, and privacy implications. This chapter provides a new methodology of self-localization for mobile devices in uncooperative legacy networks. This approach covers most of the localization services, unlike localization of unaware and uncooperative targets that is used in niche surveillance applications [70].

Among the already mentioned techniques, those based on RSSI are available on common smartphones. In fact, WiFi card drivers make the received power level, expressed in dBm, available to the application layer.

RSS-eye main contribution resides in its novel method to build and analyze RSSI angular profiles to perform indoor localization at the handset-side. The RSSI angular analysis is more independent from context in comparison to RSSI-based ranging. Angular profiles are independent on the beaconing transmission power because the information is kept by RSSI relative changes and not by their absolute values.

RSS-eye guarantees privacy of user's location because it completely works on the mobile target. The user is not obliged to share her position with third parties, unlike in network-based localization. RSS-eye user is not required to be associated to any network, therefore the system works even in environment that are not familiar. Even if not strictly required, the wireless network may facilitate they user positioning, for example, by disseminating AP coordinates inside beacons frames.

5.2 Related work

WiFi positioning uses the WiFi access points (APs) deployed in indoors during recent years. Access points periodically broadcast WiFi beacon frames to announce their existence and in urban areas with high density wireless networks, several APs can are heard at any point. RSSI-based localization has been generally based on ranging or fingerprinting. *Ranging* is the process to determine the distance between the AP and the station. It has been traditionally computed by measuring the received power and comparing it with a propagation model, generally a power-law.

On the other hand, *fingerprinting* algorithms search for the best matching entry in a radio map that was pre-recorded at different positions. Fingerprinting relies on two phases: first a radio map is created (off-line or training phase) and then RSSI values are sampled and compared with the radio map during the on-line or positioning phase [71, 72].

Using previously recorded radio maps can cause errors due to system dynamics occurred between the training and the positioning phases. Furthermore, errors and computational burden depend on the density of the radio map. The use of ranging and fingerprinting for WiFi localization has been recently diverted towards other strategies: (i) network-based direction finding, (ii) handset-based angular power-delay analysis, (iii) use of the channel state information (CSI) rather than RSSI [73, 74]. The use of rotating radio beacons in analogy to rays emitted by lighthouses was proposed in [75] and in [76] the angle of arrival is estimated in line of sight and along few meters, through IEEE 802.15.4 RSSI and omnidirectional rotatable antenna. In both cases, rotating antennas are expensive devices and make this solution unfeasible in legacy environments, both on the network-side and the handset-side.

Spinloc is a handset-based localization system that uses direction of arrival [77]. The system requires a little human intervention to collect the power-delay angular profile. The system exploits the human body as an obstacle to WiFi signals and therefore the whole system behaves as a directional antenna [78]. The power-delay profile mitigates errors due to multipath but it can be currently obtained only on a limited set of platforms [79]. Building multipath profiles and using them in analogy to synthetic aperture radars (SARs) and in conjunction with vision-based techniques allows to obtain accurate results [80]. However, the system requires specific hardware to use CSI information, which is equipped with Intel 5300 wireless cards [79].

Another key block of the system is the localization algorithm. It receives directions of arrival from the access points and provides position estimation. Finding a location by measuring angles from known points is recognized in the navigation field as *triangulation or resection*. Several hundreds of algorithms have been proposed from the dawn of time.

The RSSI variability with the angles was already studied in RADAR [72] where the user heading was roughly approximated to one of {North, East, South, West}. However, adding angular information to RSSI fingerprinting increases not only the accuracy but also the computational burden because of the growth of the radio map. Another approach considers the RSSI values at a location as the average of the RSSI collected with different heading angles [81, 82]. This goes towards simplicity but useful information is irremediably lost.

5.3 System description

The system relies on the following fair assumptions: mobile devices are equipped with digital compasses, access points are deployed at known locations and they can be distinguished by their MAC addresses, which are also known. These assumptions are justified by the birth of public repositories of access points locations such as [83] and by observing that owners of large public indoors may provide maps with indications of AP positions, e.g. using Google indoor maps. RSS-eye uses received power levels to estimate the direction of arrival from legacy access points and infer the node position. The key advantage of this methodology resides in its simplicity, availability over multiple platforms and its independence on several factors: AP transmission power, driver/firmware implementation, IEEE802.11 protocol variant, as well as context dynamics due to movements of people, layout, etc.. RSS-eye does not require radio maps and therefore it is more robust and less effort demanding than systems that use fingerprinting. This simplicity and flexibility comes with a tradeoff on the requirement of little human intervention.

When the *on-demand localization service* is requested, the user spins around grasping her phone and spanning the dashed arc, as depicted in figure 5.1. The user heading at time t_n is $\beta(t_n)$, N indicates the North reference direction and β_{min}, β_{max} are the extremal angles of the rotation.

The system architecture is composed by two main blocks reported in figure 5.2: the sensing module and the processing module. The first block collects RSSI data from the WiFi card

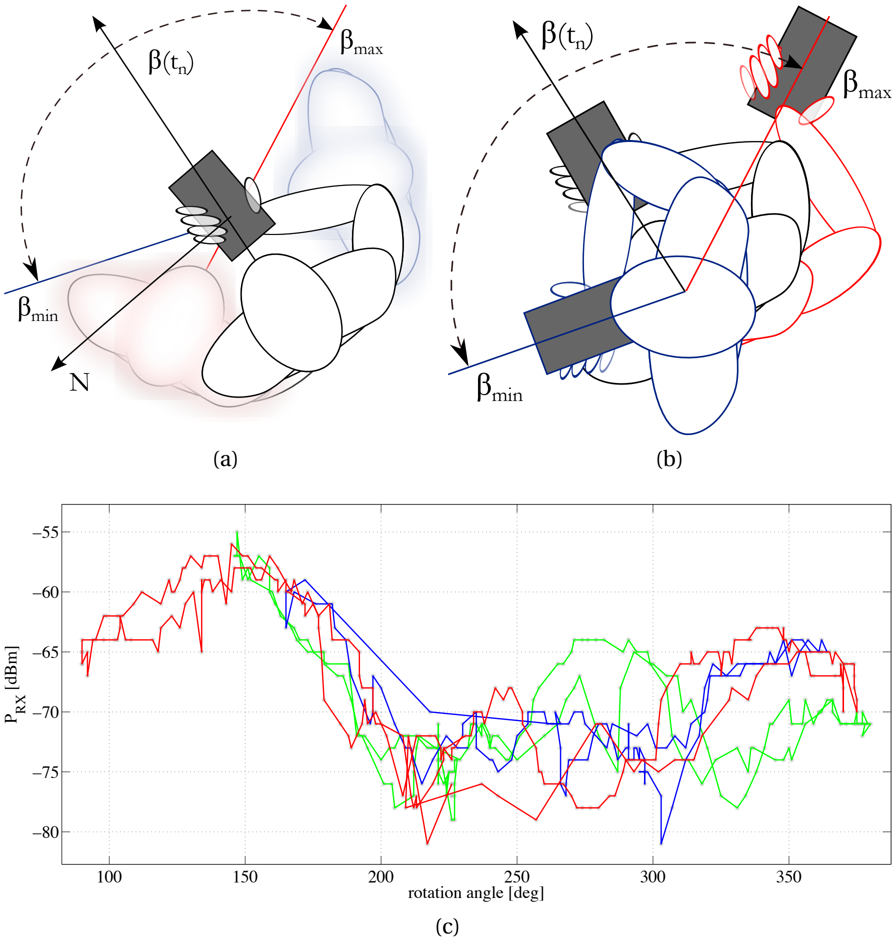


Figure 5.1: Definition of angles (a) and pivoting mismatches (b). Hysteresis errors affecting an RSSI angular profile (c).

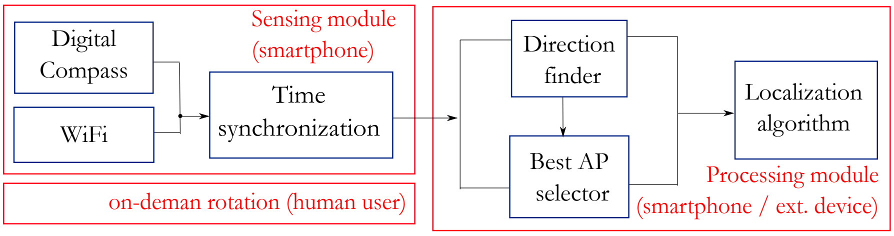

Digital Compass
WiFi
Time synchronization
Sensing module (smartphone)
on-deman rotation (human user)
Direction finder
Best AP selector
Localization algorithm
Processing module (smartphone / ext. device)

Figure 5.2: Block scheme of the localization system architecture.

and reads true bearing angles from the magnetometer digital compass, given the magnetic North as reference. Measurements are time-stamped with the same time reference, in order to perform later synchronization. The processing module analyses the data from several APs and finds the direction of arrival, then selects the best two APs that, accordingly to a quality criterion provide the minimum angular error.

The computed directions are given as input to the localization algorithm, which extends triangulation. Both sensing and processing modules reside on the user's smartphone. There is no specific requirement on the processing module, it can also be located remotely to mitigate the local computational burden. The sensing module collects RSSI data from all access points in the hearable range in a single spinning. The RSSI angular profile contains data from all the hearable access points. It is a panoramic radio image of the received power, which is shot at one point including all available directions. This explains the RSS-eye name.

5.3.1 Performances of the proposed system

Indoor localization is evaluated accordingly to several metrics including accuracy, latency, energy efficiency, scalability, repeatability, and environment robustness. A general agreement on such metrics is still to come, as demonstrated by dedicated European projects [84], however the tradeoff between accuracy, pervasiveness and cost is generally considered the primary comparability metric. As RSS-eye has no monetary costs for dedicated infrastructures and devices, we evaluate costs in terms of energy consumption and human effort and performances using response time and accuracy. Under the assumption of static localization, we focus on the following question: how long to spin to be localized with reasonable accuracy? The sensing module works on beacon frames, which are periodically sent by the APs to announce networks and have no relation with number of stations and traffic loads. Beacons are transmitted at the basic rate, so RSSI readings are independent on the modulation and coding scheme and the receiver works at its top sensitivity.

The target beacon transmission time (TBTT) is the beaconing period, whose default value is 100 ms. Scanning 13 channels in a single direction requires more than 1.3 s and an entire RSSI profile, spanning 360°, would require 7.8 minutes, un unfeasible amount of time. The spinning time was reduced by: permitting less than 360 angular samples spread over the full circle, collecting beacon frames from multiple APs simultaneously, as well as limiting the scanning to a single channel. The last limitation is not a problem, given the high density of hearable APs. Additionally, although not implemented in our solution, an initial scan over all WiFi channels may indicate the best one to use. The resulting spinning period is about 20s.

5.3.2 Sources of errors for the sensing module

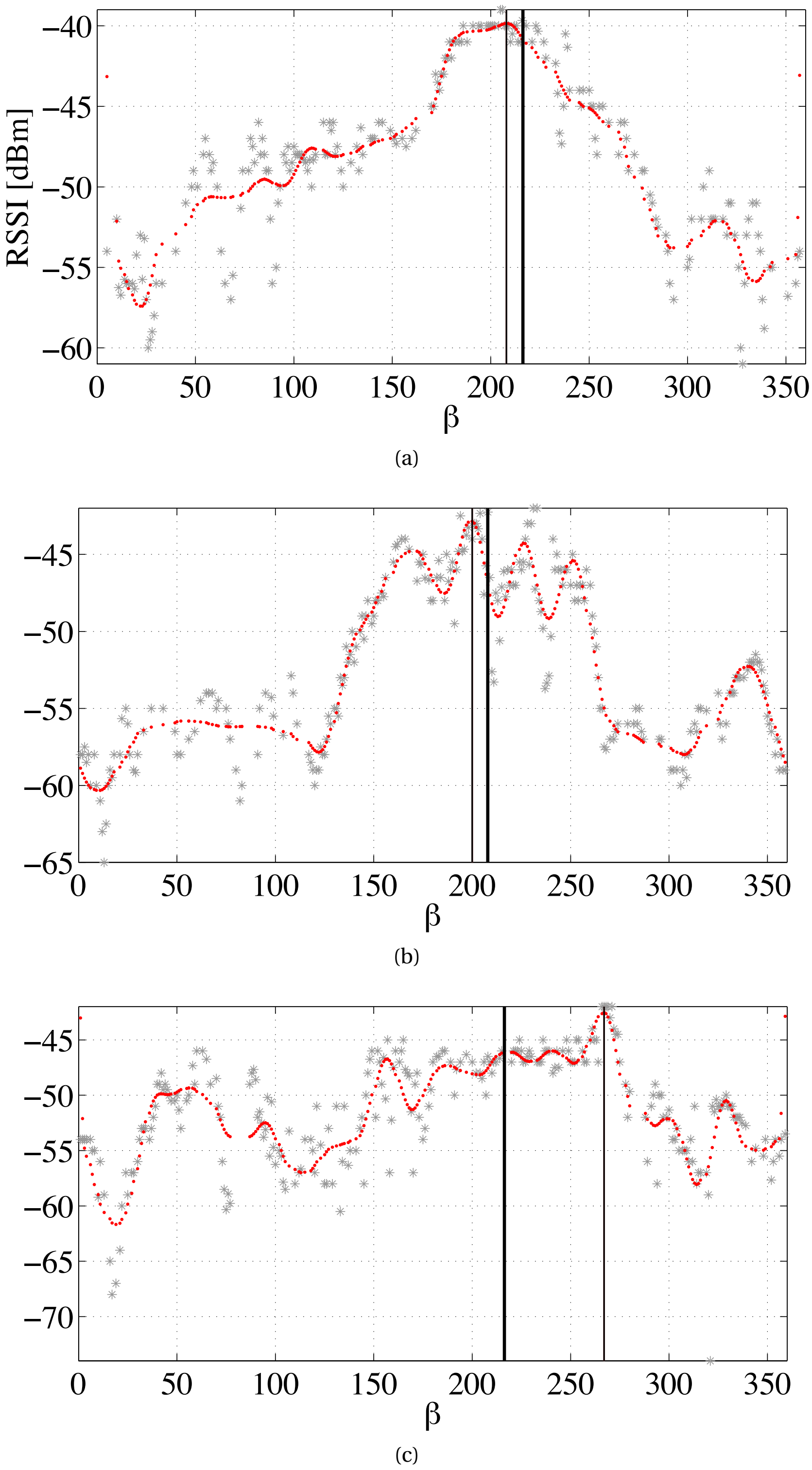
RSSI-based localization systems are prone to errors due to several factors including multipath, context dynamics, device heterogeneity (radio front-ends, antennas, RSSI computation and granularity, geometrical mismatches). Furthermore, our system is also sensitive to errors from sensors and pivoting mismatches, reported in figure 5.1. In the following we describe the phenomena and how they are mitigated or cancelled by RSS-eye.

Digital compasses measure bearing angles and introduce a latency due to the inertia of the sensor and its embedded low-pass filters. Figure 5.1-(c) reports this phenomenon through experimental RSSI values. The user spins three times back and forth and her heading vector spans 300° in the range $[80°, 380°]$. The center of rotation is not exactly fixed, therefore while spinning she moves around about a wavelength, about a dozen of centimeters. RSSI traces are depicted with blue, green and red colors according to the different rotations. Analyzing data coming globally from the three rotations, a cloud of RSSI values with high dispersion would be recognized in the right-most part of the figure. The three rotations correspond to well-shaped traces whose local maxima are shifted each other because of two concurrent phenomena: multipath and digital compass inertia. The compass inertia introduces latency, it causes the hysteresis curves depicted with different colors. Given the clockwise angular convention to measure bearing angles, latency shifts RSSI traces forward when the user spins from left to right (towards increasing angles) and backward while spinning in the opposite direction (towards decreasing angles). To mitigate the effects of inertia, a single rotation is preferable and wherever multiple RSSI values are sampled at the same angle, they are averaged.

Digital compasses are interfered by permanent magnets and ferromagnetic materials at close proximity. Those errors can be partially corrected using an iterative magnetic triangulation technique [85] and by removing artifacts in the temporal trace provided by the compass.

Multipath is the main source of errors in indoor localization based on radio propagation. RSSI angular profiles contain several peaks due to multipath rays. In severe multipath conditions the direct path can have less power than some rays due to multipath. A key observation appeared in [74] is helpful to RSS-eye, although it was originally applied to a different methodology. The authors measured angles of arrival at network-side using phase shifts and antenna arrays and observed that the peak corresponding to the direct path persists in their profile when the receiver moves a wavelength away while peaks due to multipath rays completely change or disappear. This phenomenon is observed also in our RSSI angular profiles, as shown in figure 5.1-(c) where the local maxima of the green trace is shifted to the left, in comparison to red and blue curves. Since the wavelength of WiFi signals is about 12 cm, it is not difficult to create two (or more) RSSI traces with spatial diversity. The comparison between those RSSI angular profiles filters out directions of arrival of multipath rays.

Another different approach to analogous observations was described in [86]. The user mobility leads to changes in the direction of arrival of the direct path, which can be geometrically determined. The physical displacement acts differently on the direction of arrival for the direct path and those for multipath rays. The matching between the geometrical computation and the measured variation indicates the direct path among the multipath ones. Unfortunately, such solution requires sensing both at client and at network sides; this strongly limits the



(a)

(b)

(c)

Figure 5.3: RSSI angular profiles; raw RSSI data (gray stars) and low-pass filtered RSSI profiles (red curves). LoS (a), near LoS (b), NLoS (c).

deployability in real scenarios.

The user rotation should be done pivoting around the phone, as depicted in fig. 5.1-(a). *Pivoting mismatches* impact on RSSI profiles by introducing spatial diversity at each angle. Additionally, the arm movement tends to keep the phone closer to the body when bent and farther when distended (see fig. 5.1-(b)). Finally, instead of complete a revolution, limited rotations spanning the sector between β_{min} and β_{max} are allowed. If the direct direction to the i-th AP falls into the described sector, then its direction of arrival is detected, otherwise, even in presence local maxima, they are filtered out by the multipath suppression procedure described above and no direction will be provided. Left-to-right and right-to-left scanning over small sectors fits the needs of recently appeared navigation applications such as ARIANNA, which navigates blind or visually impaired people. By the mean of their phones, they can follow markings on the floor [87].

Antennas with different polar radiation diagrams produce diverse RSSI angular profiles. This heterogeneity changes the shape of the RSSI curves but has no effects on their local maxima, which are used to compute angles of arrival.

5.4 Methodology

As discussed in section 5.3.1, collecting a single RSSI profile using standard Android APIs may require even minutes. In order to speed up the sensing process, we flashed our Nexus S with the CyanogenMod aftermarket firmware [88], then gained root privileges to set the wireless interface to monitor mode. The sensing module collects RSSI values for all received beacon frames, therefore spanning the full circle requires less than 20 s.

Collected RSSI samples are unequally spaced both in time and in angle. They are not equally sampled in time because of the clear channel assessment. In fact, even periodically scheduled beacon frames have to be postponed if the channel is busy, therefore RSSI is sampled at t_n where $t_n \neq n \cdot TBTT$. RSSI are sparse also over angles, in fact not all angles in $[0°, 360°]$ have one corresponding RSSI reading. This is due to the user's spinning speed and the magnetometer inertia. Given the mobile user at position $P = (x, y)$, the sensing module timestamps the sampling instant t_n and add the user heading β_n joining them in the following tuple:

$$[r_n \; \beta_n \; t_n \; MAC_j]; \tag{5.1}$$

where $n \in 0, 1, ...N_j - 1$ and $j \in 0, 2, ...M - 1$. We indicate with r_n the RSSI sample, N_j is the number of samples collected for the j-th beaconing AP whose address is MAC_j. Finally, M is the number of available APs in the area. This raw dataset may contain records sampled at the same angle, these are removed by averaging the corresponding RSSI values. This operation changes the set of β_n but for the sake of notation simplicity, this new set without duplicates will again be indicated as β_n but this time $n \in 0, 1, ...N_j^{(\beta)} - 1$, where $N_j^{(\beta)}$ the number of distinct angles appearing in the j-th profile. The resulting dataset has no duplicates, however some angles can have missing RSSI readings because they are sparse over the full circle.

With $r_j(\beta_n)$ we indicate the noisy RSSI readings at different angles, for the j-th access point. This has to be low-pass filtered in order to obtain a smoothed version. RSSI profiles are filtered through windowing, being the frequency coefficients of the window W_k, $k = 1, 2, ...K$. Standard FFT algorithms are not applicable to RSSI profiles because they often are sparse over 360° and non uniformly spaced.

We apply a non-uniform FFT algorithm [89], and obtain the smoothed profile $\tilde{r}(\beta_n)$, that we name *RSSI angular profile*.

$$R_k = \frac{1}{N^{(\beta)}} \sum_{n=0}^{N^{(\beta)}-1} r(\beta_n) e^{-jk\beta_n} \tag{5.2}$$

$$\tilde{r}(\beta_n) = \sum_{k=1}^{K} R_k W_k e^{jk\beta_n} \tag{5.3}$$

For brevity of notation, we omit the pedis j in equation 5.2 and 5.3, being evident that it has to be applied for each access point.

The profile $\tilde{r}(\beta_n)$ presents several local maxima corresponding to the angles of arrival for the direct path and for multipath rays. Fig. 5.3 displays three exemplary experimental RSSI profiles. Raw RSSI sample $r(\beta_n)$ are depicted as a scatter plot of grey markers while the low-pass filtered profile $\tilde{r}(\beta_n)$ is depicted in red. The vertical thick segment corresponds to the ground truth of the direction of arrival for the direct path. The thin vertical segment indicates, in a first approximation, the angle corresponding to the absolute maximum of the received power.

Figure 5.3 reports three possible cases: line-of-sight, near line-of-sight and non line-of-sight. In subfigure (a), the power received along the direct path describes a well-shaped peak and the absolute maximum corresponds to the right direction. Multipath is more evident in (b), where rays incide with different angles at the receiver and with similar power, but the chosen peak is the right one, therefore the error is due again only to compass. On the other side, in (c) it is shown a severe multipath condition in non-line-of-sight. Multiple rays contribute in phase over the direction $\beta = 267°$ and the received power in this direction is higher than the power coming from the direct path. This error is larger than in the first two cases and the ground truth lays on a different peak.

From this example it is evident that RSSI profiles lead to directions of arrival with different errors, therefore a quality indicator is needed to compare and order profiles. We consider the following observation: the more peaks appear in the profile and the higher is the probability to incur in errors due to multipath. The *quality factor* for the j-th profile is:

$$Q_j = \frac{N_j^{(\beta)}}{p_j \cdot 360}; \quad 0 \leqslant Q_j \leqslant 1 \tag{5.4}$$

where p_j is the number of peaks in the RSSI angular profile $\tilde{r}(\beta_n)$ and $N_j^{(\beta)}$ is the number of distinct angles appearing in the j-th profile. Given N_j the total number of RSSI samples in time, it is valid the relation $N^{(\beta)} \leqslant min(360, N)$. Profiles with $Q = 1$ have only one peak and are sampled over the full circle, therefore they lead to the best estimation of the direction of arrival.

The *localization algorithm* receives a set of bearing angles β_j and a set of quality coefficients Q_j. The higher is Q_j and the better is the angle estimation, therefore the lower is the aperture of the corresponding circular sector, as shown in figure 5.4 where $Q2 < Q1 < Q3$. Pointing on the j-th AP, whose position is known, the direction towards the target point is $180 - \beta_j$ (see fig. 5.4-(b)). The intersection of sectors provides a polygonal area whose centroid is the estimated location.

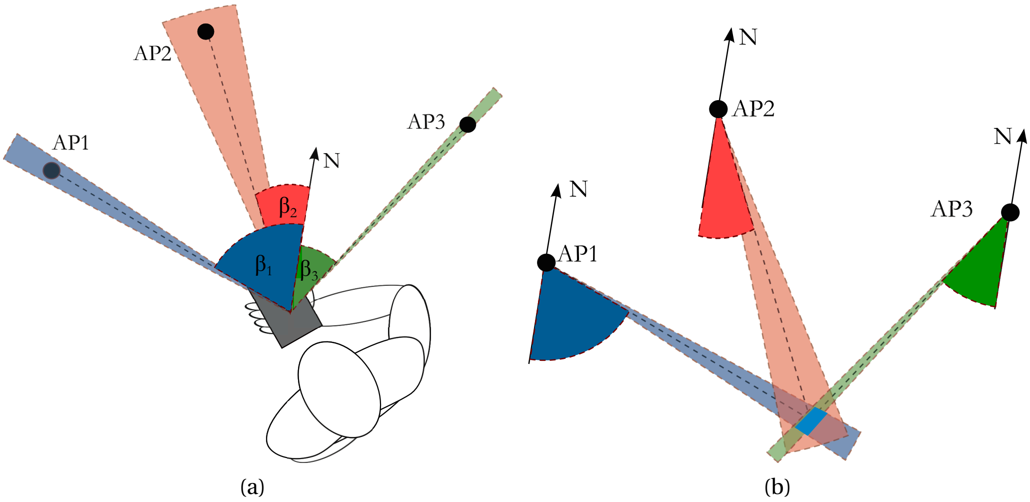


Figure 5.4: Building the RSSI profile (a) and estimating position (b).

5.5 Experimental results

We ran two series of experiments, one in a multipath-controlled environment and the other on the second floor at our department. In the first case we analyze, one by one, errors on angular profiles, in the second, the whole localization system is validated under difficult indoor conditions.

As discussed in section 5.3.1, multipath and human body generate artifacts in RSSI angular profiles, therefore comparative experiments with and without such error sources allow to isolate their effects. This set of experiments are performed in a semi-anechoic chamber, which provides a controlled multipath environment. The walls absorb WiFi signals while the floor is a reflector, therefore propagation fits the two-ray ground-reflection model.

The semi-anechoic chamber is instrumented with a rotating table, programmed via NI Lab-View to span $\alpha \in [0°, 180°]$ with 1° steps, lasting 3 seconds each. The table guarantees fine-grained angular positioning and no pivoting mismatches. Angles are measured clockwise using the segment between the center of the platform and the AP as reference. The AP is in the opposite side of the chamber 4.3 m away. RSSI profiles are shown in fig. 5.5 both with and without the human body. Both curves fit the trend of the theoretical two-rays ground-reflection model and we explain fluctuations with non-eliminable multipath caused by metallic parts of the experimental setup including the shell of the piloting laptop. Therefore the residual multipath provides RSSI fluctuations also in the semi-anechoic chamber. The second set of experiments is performed in the offices at the second floor of our department covering more than 1000 square meters. The severe multipath conditions, due to metallic bookcases, allow to validate the robustness of our solution. The wireless access network is composed by seven Routerboard APs of the RB951G-2HnD model. These broadcast beacon frames every 100 ms. Measurement repeatability is checked using different platforms at the handset side: one Samsung NEXUS S smartphone and one Alix device from PCengine, equipped with SR71-A wireless card containing an Atheros AR9160 chipset. This has -82 dBm and -79 dBm sensitivity at MCS0 and MCS1 respectively and is terminated on a 8 dBi Yagi printed antenna. The RSSI profile changes depending on the device, however local minima and maxima persist.

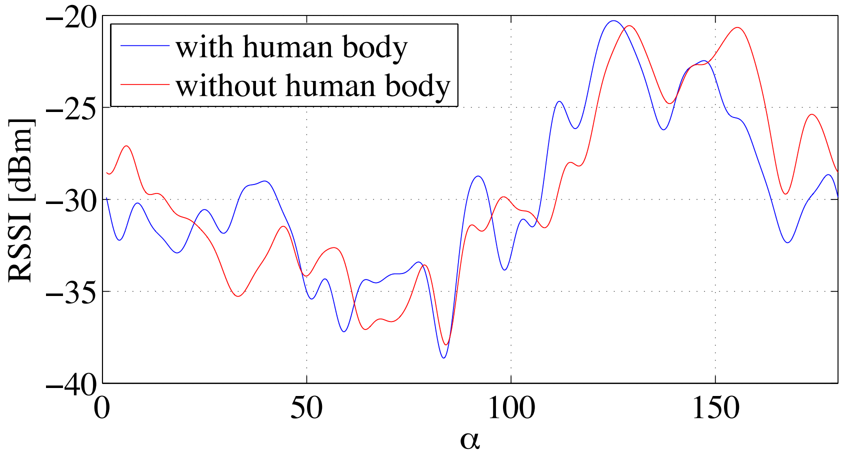


Figure 5.5: RSSI angular profiles taken with and without the human body in the anechoic chamber.

Figure 5.6 reports the map of the second floor at our department, with an exemplary estimation. Our implementation is a discretized variant of the intersection between sectors. Multiple rays are spread from the positions of the best APs. The intersections among lines are marked with red dots and describe a polygonal area whose centroid corresponds to the estimated location. The blue dot stands for the ground truth position while the estimated one is marked in green. RSS-eye estimates the angle of arrival of the direct path with a median error of 24°, while the localization algorithm provides the user location with a median error of 3.7 m. Despite RSS-eye does not provide breaking-through accuracy in localization, its good accuracy in the estimation of the angle of arrival makes it a promising solution.

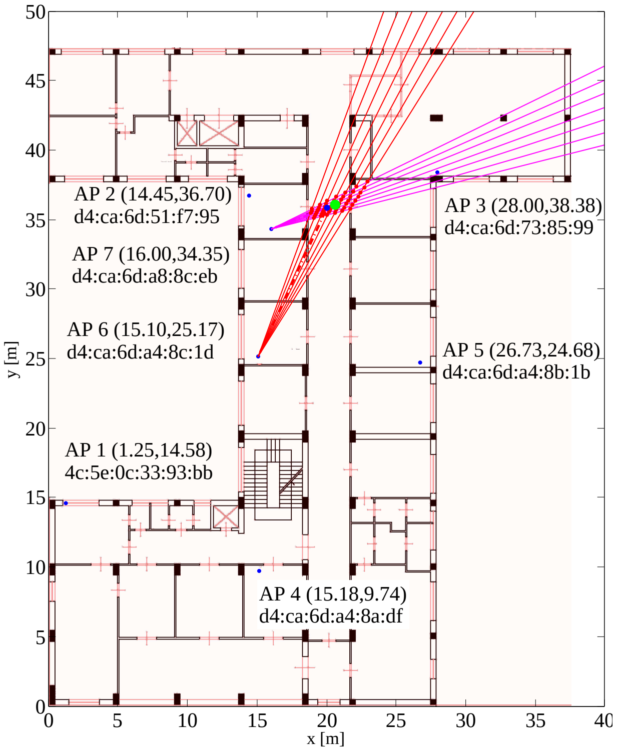
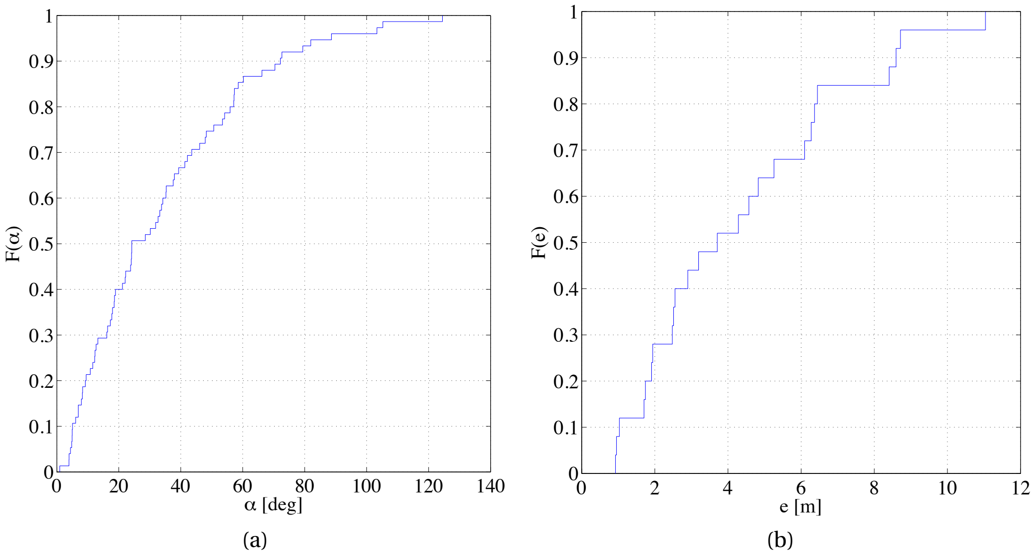


Figure 5.6: Plan of the floor including APs, ground truth and estimated position.



(a)

(b)

Figure 5.7: CdF of AoA errors (a) and of positioning error (b).

5.6 Summary

The RSS-eye localization system has reasonable accuracy, the highest coverage due to beacons frames, good scalability as well as low complexity and costs. The wireless network does not require optimized planning, dedicated infrastructure, nor radio maps. The system self-adapts to the context and to causes of errors because angular RSSI profiles deal with variations rather than absolute values. RSS-eye tradeoff is the active user collaboration in the sensing process, pivoting around to create a panoramic radio vision, the RSSI profile. RSS-eye is a on-demand indoor positioning system. It is complementary to inertial navigation methods because it periodically reduces the error they accumulate. Finally, RSS-eye methodology on spinning and angular profile can be easily applied to non-WiFi signals, especially those who natively provide RSSI values.

6 Panoramic and angular fingerprinting

In this chapter we validate the proposed architecture by defining two new fingerprinting methods: *panoramic fingerprinting* and *angular fingerprinting*. Our work on these directions is still ongoing, therefore we provide only basic ideas and preliminary results.

Both methods consider fingerprinting using multidimensional RSSI vectors and a comprehensive information on user's orientation, thanks to the intuition, in common with chapter 5, that user and smartphone, together, have directional properties.

6.1 Related work

As introduced in chapter 1, fingerprinting techniques share a key basic principle: signal strengths are measured or theoretically estimated over a grid of known positions. These values are named training data and are collected during the off-line phase. Training data regarding Wi-Fi signals are named radio maps.

Afterwards, testing data are taken in the target point during the online phase. Comparing training and testing data using deterministic or probabilistic approaches, the estimated position is provided.

Fingerprinting was introduced in [5], by exploiting signal strength to estimate user's position. The authors recognize the importance of the user's orientation but they curb their study to a limited set of directions {North, South, East or West} and consider the angular dependency as a challenge rather then an opportunity.

Traditional fingerprinting suffers from context dynamics, which occur between the training phase and the positioning phase. Static and moving obstacles cause similar fingerprints in points at distant locations, therefore large positioning errors affect the estimation.

In [90], fingerprinting tests were run with the user holding a fixed orientation in all testing points. The lack of accuracy drove the authors to use an hybrid solution where both Wi-Fi and acoustic signals were exchanged among peer smartphones.

When fingerprinting is applied to images and sounds it is generally called patter matching. This may be used to estimate logical locations (in the shop, at school, at home, ...) rather than physical ones (in room 509, at coordinates (x,y), ...) [3].

Cameras and microphones are the sensing devices used to estimate user's position. Light and noise conditions change dramatically from night to day therefore two sets of training data are taken into account accordingly to the time of day.

In order to avoid the off-line measurement phase, it is possible to use a propagation model and the geometry of the building to compute a fine-grid signal strength map. This solution is adopted in ARIADNE [91]. A set of potential positions are hierarchically clustered, then centroid of the cluster containing the highest number of points is taken as the estimated position.

ARIADNE considers the clustering history to track mobile users. The sensing module is implemented at the network side and it is required that the mobile node is associated to the network.

6.2 Definitions

Panoramic fingerprinting receives RSSI values from the Wi-Fi card, relative to multiple AP, whereas the digital compass provides user's current heading. The methodology is enabled by advanced sensing and fine-grained timestamping, in order to correlate angles and power samples. Furthermore, an high RSSI sampling rate is needed, enough to read RSSI values for all received beacon frames.

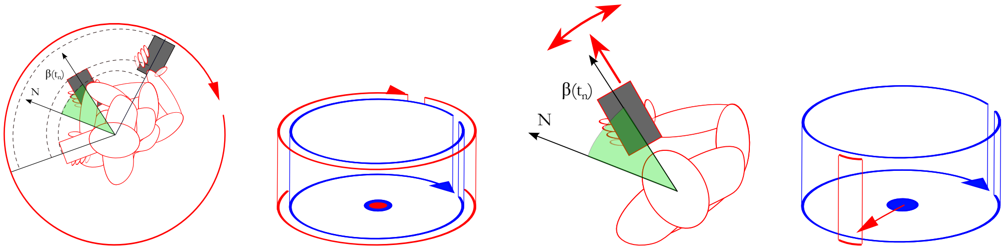
PPI are wrapped images with $Mx360$ pixels, where M is the number of hearable APs and 360 is the number of samples over the whole circle, taken with a resolution of one degree. The angle at 0° corresponds to the reference direction, generally assumed with the magnetic North. When two or more RSSI are available in correspondence to the same angle, their average is considered.

6.3 Panoramic power images

$$PPI^{(P_k)} = \{r_{ij}^{(P_k)}, i = 1,2,\ldots,M; j = 1,2,\ldots,360^\circ\} \tag{6.1}$$

PPIs, taken at point $P_k = (x_k, y_k)$, are made by RSSI values sampled at different angles over the whole circle. A panoramic radio map, is a set of PPIs taken over a grid of points. For brevity of notation, we will use the index k as an identifier of the position P_k, rather than the point itself.

PPIs may have some pixels values unknown because of missing RSSI readings for a set of AP



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

Figure 6.1: User's circular rotation used to create PPIs (a). PPIs taken during the off-line phase (blue) and the online one (red) (b). User's heading and oscillations during traking (b); PPI taken during the off-line phase (blue) and the sector taken during the on-line phase (d).

along specific angles. When an AP is not hearable in a position, its row of the PPI will contain unknown values. PPIs are circular and wrapped images having $r_{i,0} = r_{i,360}$. This simplifies image comparison because all images share the same reference even if user's rotation was not regular and started and stopped at different angles.

When the user rotates to obtain a PPI, the sampled RSSI can change due to multipath, start and end angles while rotating, pivoting mismatch (see chapter 5). Considering all these causes of errors, we provide a qualitative indication of PPI repeatability.

Several tests demonstrate the permanence of the shape in the analysis of the angular power profile. Results of this experiment are reported in fig. 6.2. The user rotates ten times standing at the same position at different time of day. Measured RSSI samples are drawn as a scatter plot marked by gray asterisks. Angular power profiles, after being averaged and filtered, are reported with colored dotted curves. Vertical lines indicate angles of arrival, which are readable in the x axis. The red dashed line remarks the true angle between the AP and the North direction, centered in the target position. Colored vertical lines remark the maximum RSSI for corresponding profiles. In figure, peaks are clearly affected by angular shifts and changes in amplitude, due to different multipath conditions experienced by the smartphone during rotation[1]. Even without perfect overlap among consecutive circular trajectories, multiple repeated measures trace profiles with the same overall trend.

6.3.1 Distance between panoramic power images

Let's consider two PPIs, taken respectively at point k and h. The distance between these two PPIs is obtained by dividing the pixel-by-pixel difference for the number of pixels having a

[1] it is worth remembering that at Wi-Fi wavelengths, even few centimeters of distance may dramatically change multipath conditions

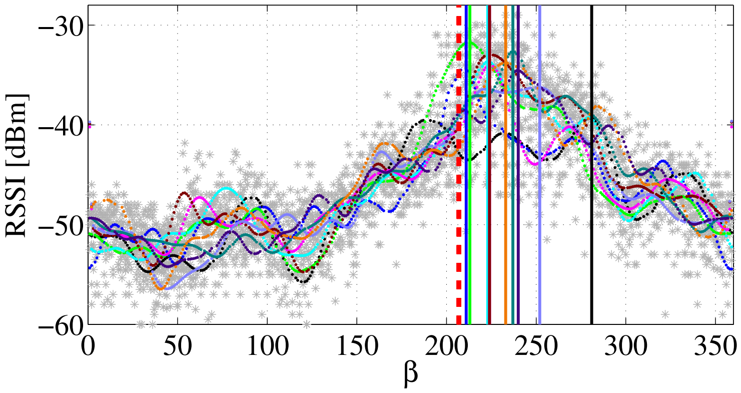


Figure 6.2: .

valid known value.

$$\Delta(k,h) = \frac{PPI^{(k-h)}}{\sum_{i=1}^{M}\sum_{j=1}^{360} x_{ij}} \tag{6.2}$$

where:

$$PPI^{(k-h)} = \begin{cases} abs(r_{ij}^{(k)} - r_{ij}^{(h)}) & if \ \exists \ r_{ij}^{(k)}, r_{ij}^{(h)} \in \mathbb{R} \\ 0 & otherwise \end{cases} \tag{6.3}$$

$$x_{ij} = \begin{cases} 1 & if \ \exists \ r_{ij}^{(k)}, r_{ij}^{(h)} \in \mathbb{R} \\ 0 & otherwise \end{cases} \tag{6.4}$$

6.4 Panoramic fingerprinting

Panoramic fingerprinting takes an image during the online phase and looks for the nearest image in the panoramic radio map, as symbolically depicted in fig. 6.1. Several algorithms have been proposed to estimate the target position. The simplest one approximates target position with the radio map point which is at minimum distance. This method provides static positioning, because the users have to spend few seconds rotating around at the same position.

We name "angular fingerprinting" the fingerprinting limited to a specific angle. The user, with his natural movements around the area, has an instantaneous heading. RSSI values are taken along this direction during the online phase and are compared with the column restriction of available panoramic images in the radio map, restricted to the same direction. The result is a comparison of two vectors: one measured during the online phase at the j-th angle, the other one extracted as the j-th column from the panoramic radio map.

In angular fingerprinting the dimension of the signal space is 360 times smaller than in panoramic fingerprinting because RSSI information is available in one direction only. Our intuition is that on one hand, this method is less accurate because it acts only on a subspace of the radio map, on the other hand, its simplicity permit to be used while the user is moving, therefore a status on position can be maintained, as in Kalman Filter or an Hidden Markov Model.

Compared to traditional methods, panoramic fingerprinting have two main advantages: (i) better context matching and (ii) more robust to context dynamics.

Panoramic fingerprinting preserves fine-grained angular information about the context. Rays converging to the sensing node from multiple scatterers make fingerprints more distinguishable. This transforms multipath from a source of errors in a valuable resource.

Furthermore, panoramic images improve the system robustness against context dynamics, accordingly to a simple intuition. In fact, moving objects and people primarily influence fingerprinting under a limited angle, i.e. under a limited field of view. The remaining circular sector is subject to lighter differences.

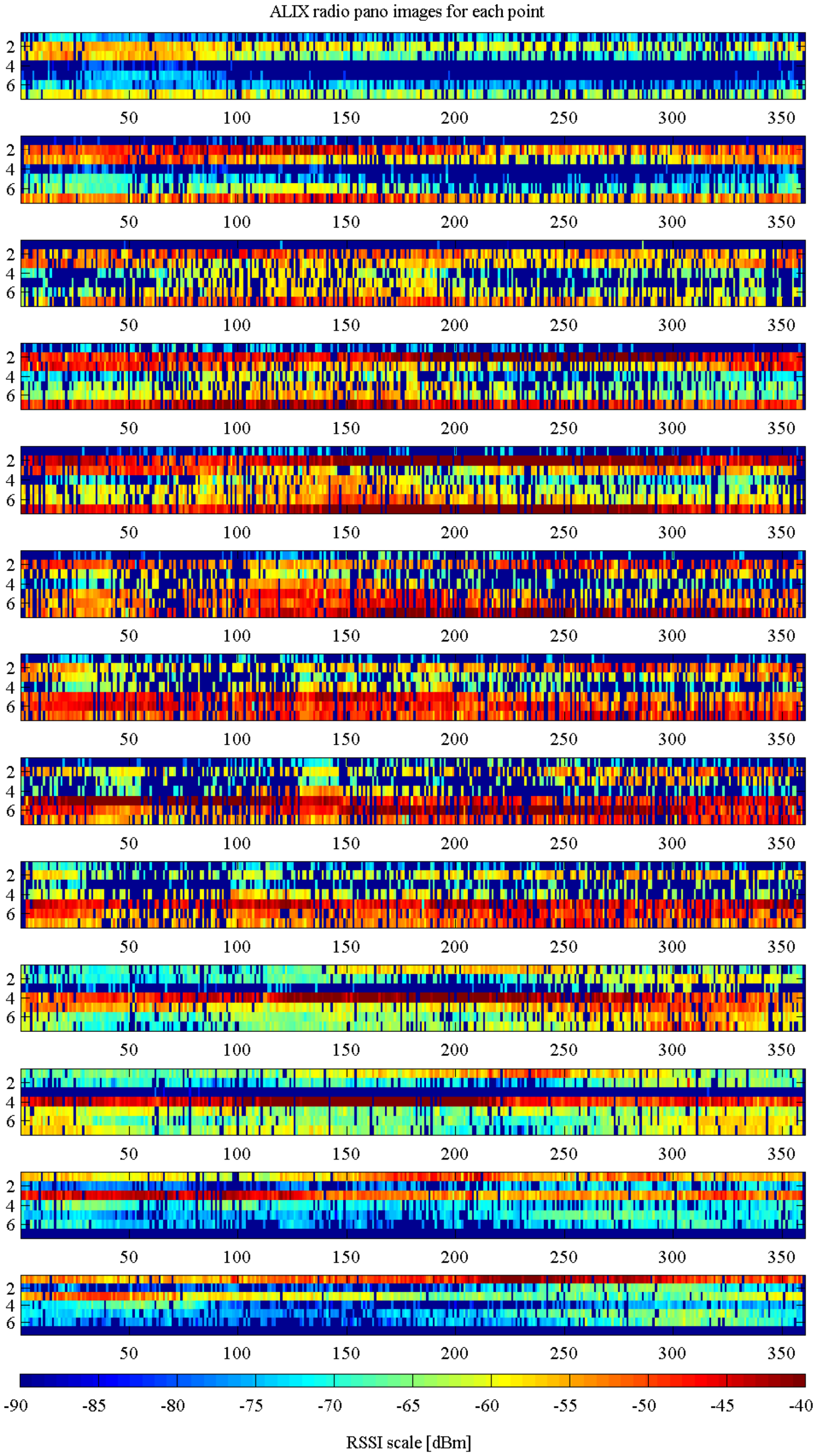
PPIs can be exploited both to estimate the angle of arrival (as discussed in chapter 5) and to apply panoramic fingerprinting, which may be also simultaneously used.

Figure 6.3 reports panoramic power images for 13 positions. Each subfigure shows angles on the x axis and AP id on the y axis. The color represents the received power. Dark blue pixels represent unknown or unavable readings.

Figure 6.4 shows similar images, rearranged by AP, showing, for example, that moving from position 1 to position 13, power received from AP1 becomes stronger.

6.5 Experimental results on panoramic fingerprinting

Experimental results demonstrate that panoramic fingerprinting provides excellent performances even under extreme multipath conditions. Panoramic fingerprinting, computed at 25 points in the second floor of our department building, provided the true best matching point of the radio map in all cases, with the only exception of point 4, as depicted in fig. 6.5. This figure reports the online position id on x axis and the position id of the radio map on the y axis. Colors represent the distance between panoramic radio images, which is lowest in the diagonal, therefore the best approximation is the right point, or at worst the closest one.



(a)

Figure 6.3: Panoramic power images taken at 13 positions.

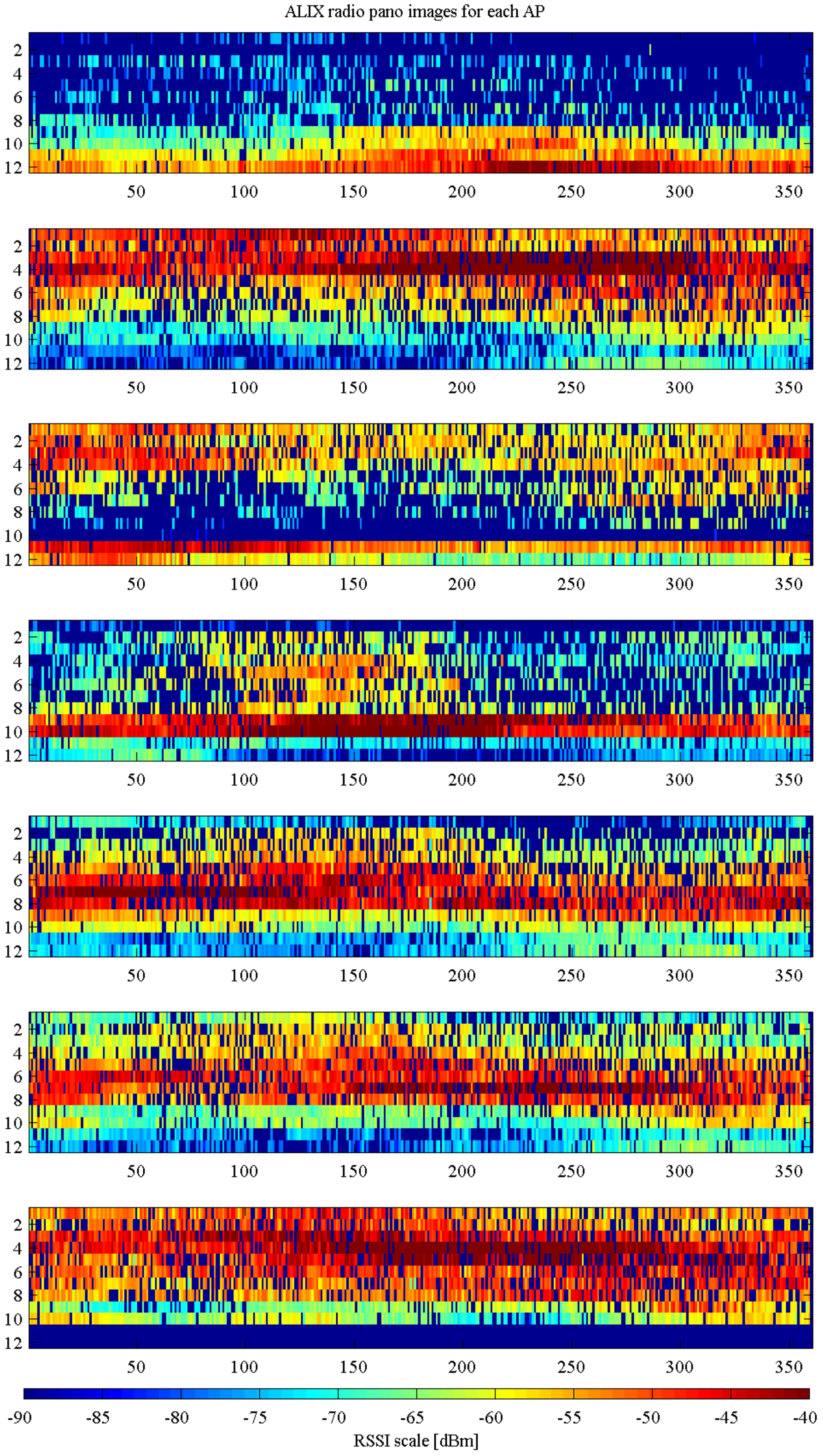
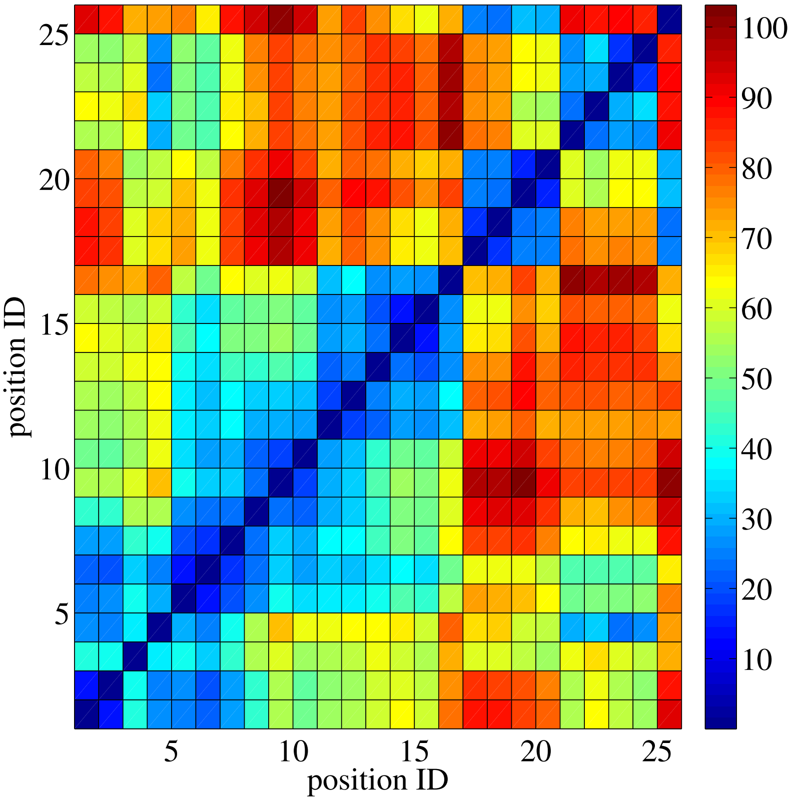


Figure 6.4: Power images rearranged by AP. Each image represents the power received by the AP, as position changes along a path. Angles are reported on the x axis in degrees and position ID on the y axis.



(a)

Figure 6.5: Tracking mobile users with angular fingerprinting

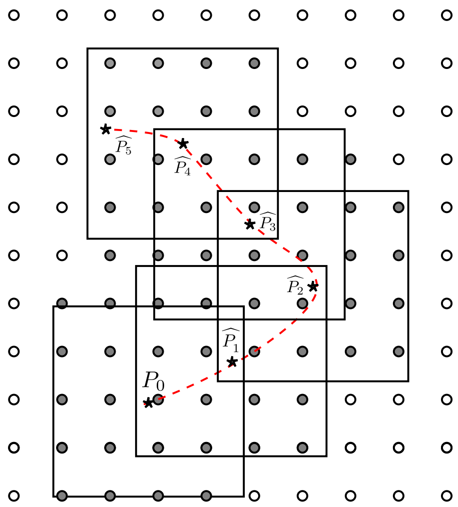


Figure 6.6: Angular fingerprinting for mobile users tracking.

6.6 Future work

6.6.1 Tracking

As depicted in fig. 6.6, angular fingerprinting can be used to track mobile users. As a future work we plan to demonstrate the use of a tracking algorithm to be run at regular time intervals. Given the estimated position at previous step, the angular fingerprinting uses the same data required in previous panoramic fingerprinting, i.e. user's heading and RSSI values. These values are then compared with the radio map, using angular fingerprinting limiting the search to grid points within a squared area centered at user's position. Furthermore, only rssi values taken along the direction measured by the compass will be taken into account. The initial user's position is assumed known or estimated using panoramic fingerprinting.

6.6.2 Radio map and propagation models validity check

As already stated, radio maps are affected by context dynamics and their usage can bring to erroneous results. To face this problem, we defined a method that improves context-awareness by monitoring context changes. Because of the high-density of indoor Wi-Fi networks, they partially overlap, permitting APs to hear each other. Using the flexible architecture that has been provided in chapter 2, we can instruct the access points to use a small fraction of the TBTT to monitor the signal strength of frames received by other APs. This is a variation of the tomographic imaging presented in [36, 37], which requires big numbers of well-positioned beaconing devices if directly used for device-free localization. Our understanding is that the use with a limited number of access points is not sufficient for estimating positioning of the obstacles but can be efficiently used to determine variations in the context, in order to understand, for example, if used radio maps and propagation models are yet valid and applicable.

6.7 Summary

This chapter presents two undergoing activities on new methods for fingerprinting. Panoramic and angular fingerprinting are presented, the first to give still positioning, the second to provide tracking. Extra validation activities are scheduled as future work, encouraged by preliminary experimental results.

7 Pattern matching and path follower

In this chapter we present a low cost navigation system, called ARIANNA, primarily designed for visually impaired people. ARIANNA (pAth Recognition for Indoor Assisted NavigatioN with Augmented perception) permits to find some points of interests in an indoor environment by following a path painted or sticked on the floor. The path is detected by the camera of the smartphone which also generates a vibration signal providing a feedback to the user for correcting his/her direction. Some special landmarks can be deployed along the path for coding additional information detectable by the camera.

In order to study the practical feasibility of the ARIANNA system for human users that want to follow a pre-defined path (by only using the smartphone feedback signals), we study how to incorporate human behavior models into the feedback control loop. We also implement an Extended Kalman Filter for localization, in which the user coordinates, speed and orientation represent the filter state (whose updating law depends on the user reaction to the vibration signals), while the smartphones sensors provide the set of measurements used for state estimation.

7.1 Introduction

Outdoor navigation based on GPS signals is a common technology which is nowadays included in many off-the-shelf smartphones. Many GPS-based applications are available in the market for aviation, naval and terrestrial uses. Recently the research community has focused the attention on indoor navigation, where GPS is not available. Navigation requires target localization, which can be done using different methods, such as triangulation of RF signals (mainly WiFi), direct sensing (with RFIDs, ultrasound, bluetooth, etc.), pattern matching, dead reckoning based on odometry readings (accelerometers, magnetometers, compasses, and gyroscopes). For example, dead-reckoning techniques are employed in Navatar [92] where users interact with the application and help correcting possible navigation errors. RF-PATH-ID [93], instead, is based on disseminating passive RFID tags and using a dedicated reader to acquire information on the user location. More examples and detailed information on indoor localization techniques may be found in [94].

Assistive tools for indoor navigation have specific requirements in terms of reaction time (they require to run in real-time to be useful) so they need an adequate refresh frequency. Tools must be light-weight, portable, low-power and low-cost and should require minimum training time. Solutions based on off-the-shelf devices can be easily spread, even better if the used devices are already available to people. In this sense, smartphones are light-weight, portable, and affordable devices that are already in everyone's pocket. For this reason, we exploit computer vision capabilities of common smartphones provided with cameras and present 'path following people' accordingly to some key ideas presented in [87]. In the present chapter, ARIANNA (pAth Recognition for Indoor Assisted NavigatioN with Augmented perception) is described, equipped with a robust tracking system based on an Extended Kalman Filter (EKF) that estimates the states from noisy observations. Kalman Filters (KF) are widely used in computer vision and robotic systems for object tracking, path following, simultaneous localization and mapping (SLAM), leader-follower systems and 3-D modeling, just to cite a few. Many applications of the KF in robot vision are summarized in [95]. With the help of the EKF-based tracking, the proposed system recovers even in case the path is temporary lost by providing a shift to a new user-centric perspective, where the navigating user runs corrective actions and is a controller in the interactive control system. The human intervention in interactive control systems is named 'human-in-the-loop' (HIL), [96]. A key part of the HIL control are human responses to stimuli: they depend on physiological, psychological and environmental factors. For example, several alternative paths might be taken due to unexpected obstacles, orientation disorders, sleepy conditions, different step lengths, etc. Primarily designed for visually impaired people, the ARIANNA system is a particular example of HIL feedback control system.

Regarding the specific case of visually impared, many recent technologies have been developed to help them move autonomously in unfamiliar environments and different interfaces have been designed to communicate with the visually impaired. For example, virtual acoustic displays and verbal commands issued by a synthetic speech display are used in [97]. AudioGPS [98] and Melodious Walkabout [99] use audio cues to provide information on the surrounding environment. However, acoustic feedback is perceived as a distraction and overloads visually impaired hearing which is already used to catch information on the near environment. It is thus preferable to avoid audio indications in favor of tactile alternatives. Indeed, haptic principles and a list of possible applications are presented in [100], while benchmark metrics for haptic interfaces have been recently proposed, based on a combination of physical and psychophysical data [101]. Frictional forces arising from the stroke of a finger moving on a surface are studied in [102, 103], while tangential skin displacement at the fingertip (stimulus speed and displacement length) communicate direction or displaying static friction in haptic applications [104]. In [105], vibrational feed-back is given by a special glove in the Finger-Braille language. This system requires some dedicated hardware and is specific to the language used. Other examples of haptic feedback can be found in [106, 107]. Compared to other solutions, the approach we will present hereafter is much easier to use because it employs the

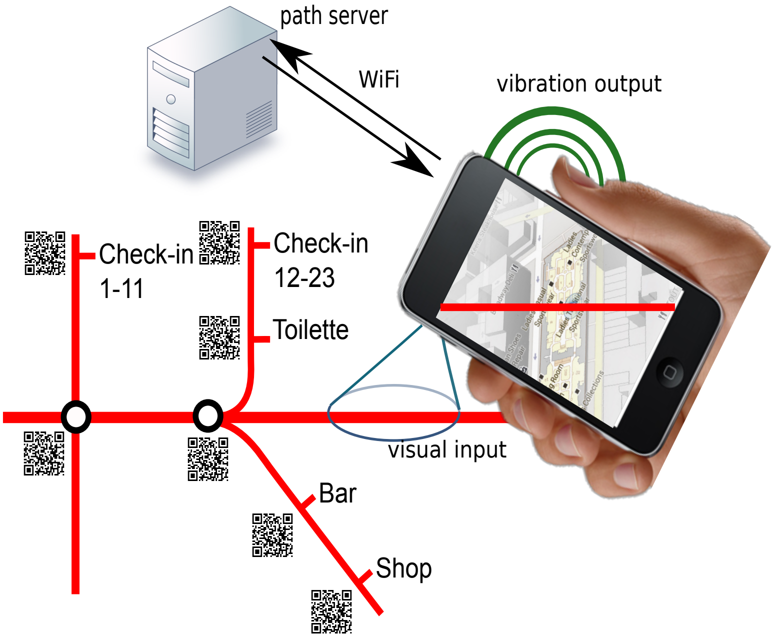


Figure 7.1: ARIANNA navigation system description

smartphone as a visual-to-vibration translator for directional information.

7.2 Navigation system description

The ARIANNA navigation system is especially designed for indoor scenarios, where GPS-based solutions are unavailable, and exploits the visual sensor and vibration signals of commodity smartphones. Figure 7.1 shows a typical use of ARIANNA in an airport scenario. The paths of interest are marked with colored lines on the floor. This is the only dedicated instrumentation applied to the environment and is a quite simple and low cost solution; QRcodes are settled close to points of interest, on line intersections and are also used for landmarking. They provide information on the right line to follow in order to get to the desired destination. The user interface employs tactile stimuli to receive feedback on the heading corrections to be employed, as better described in the following. The systems itself is composed by five main components: (A) ambient instrumentation; (B) sensors; (C) data transport network; (D) path server; (E) tactile interface.

7.2.1 Ambient instrumentation

The ambient instrumentation is composed of colored tapes which can be easily sticked on the floor or carpets to define different paths. This is the only dedicated instrumentation applied to the environment. Paths can intersect each other forming a planar graph where intersections are nodes of the graph. To add information on the paths, any segment (the graph edges) may be deployed with two parallel strips with different colors, so the ordered couples (color1, color2) and (color2, color1) encode both direction and orientation. Additionally, using bar codes or QRcodes it is possible to encode relevant information regarding the edges (as for example the distance from/to the extremes of the segment) and for landmarking.

7.2.2 Sensors

The main sensor used in the ARIANNA system is the camera, which most smartphones on the marketplace are equipped of. The camera is used to reveal the presence of lanes on the floor and acts as a visual control for the haptic transducer. We also use the embedded compass and accelerometer sensors to help maintain or recover the visual control of the line in the EKF-based tracking. All these sensors are available on most commodity smartphones: this is a key aspect for keeping the system low-cost and affordable for a vast public.

7.2.3 Data transport network

We assume that a data network connection is available for downloading the ambient map (e.g. via a WiFi or 3G network). The data transport network does not require specific adaptations but is a facility that permits communication between the phone and the ARIANNA server. The server is used to provide localization information, correlation between paths and points of interest, routing towards the destination. The presence of the server and the wireless network is necessary only in case the application is unaware about the building topology and its deployed paths. On the contrary, if the application loaded on the phone has such information locally available, the presence of network and server is optional (even if flexibility is possible only with those elements, as explained below).

7.2.4 The path server

The path server stores and retrieves information from a path repository via the url printed into the QRcode. The content pointed out by the (fixed) url can be changed on the fly with a simple update on the server. Such flexibility permits path adaptation required by topological changes due to maintenance or load balancing. When the smartphone detects a QRcode on the path, it immediately runs an http request to the server using the url inside the QRcode. The server knows the position of the user (because of its proximity to the QRcode position) and sends back to the smartphone the next edge to follow. In fact, among all paths deployed in the building, thanks to the indications provided by the path server, the smartphones provides haptic feedback only towards the “enabled” paths according to the server indication.

7.2.5 Tactile Interface

The tactile interface is a key point of the system. The behavior of the haptic feedback can be summarized as follows: the camera continuously grabs the scene in front of the person.

The tracking system incorporates the information on the line (together with the compass and accelerometer data) and provides feedback with the phone vibration. The intensity and type of the vibration is based on the output of the EKF and is designed to keep the camera always in contact with the line or to bring back the visual contact when it is lost. Vibration is a native functionality of the phone obtained through a rotating eccentric mass. It has been shown that the current consumption of typical vibration motors has a limited impact on the battery life of commercial smartphones [108] and that the energy savings coming from switching off the screen are higher than the costs introduced by vibrational cues [109]. Unlike other approaches in haptic interfaces, our solution does not need a selective vibration of the touched point (that is also difficult to obtain and requires special piezo-electric materials, etc.).

7.3 Human in the Loop

In order to study the practical feasibility of the ARIANNA system for human users that want to follow a pre-defined path (by only using the smartphone feedback signals), we need to determine how to incorporate human behavior models into the formal methodology of feedback control as in [96].

For sake of simplicity, we assume that smartphone and user positions coincide and movements are possible in a 2D environment (i.e. we do not consider changes in user elevation). The paths are represented by a piecewise constant function of the space that is different from zero along the path points. We also assume that the user starts navigating from a point belonging to the path and that the smartphone is able to observe a squared region of the floor whose dimensions are about 50 cm each. The goal of the application is allowing the user to reach the end point of the path. Since smartphone signals are generated and updated at discrete time steps, we consider a discrete time system in which the walking behavior of the user is updated at the same temporal scale of the feedback signals. Finally, we consider that the pedestrian speed is approximatively constant for a given user (although a real estimation of the speed is possible and will be discussed in the next section).

Let β be a generic constant direction of a portion of the path. User reaction to the smartphone signals is described in terms of a change of his heading direction α. Such a change is based on two different information provided by the phone: i) the user direction relative to the path $\beta-\alpha$, that is perceived according to the alignment of the vibration points on the phone display; ii) the distance between the user and the path, that is perceived according to the distance of the vibration points from the center of the phone display. It is reasonable to assume that the user will correct the walking direction by trying to be aligned to the path and to null the distance from the path in the next steps. Let α_k be the user direction at the discrete time k, v the user speed, and T_X the desired maximum time interval for nulling the path distance.

We consider the following *human in the loop* model. When the smartphone is able to see the painted line on the floor, the user heading direction α_{k+1} is updated by considering a first correction proportional to the perceived deviation from β (i.e. $\beta-\alpha_k$), and a second correction

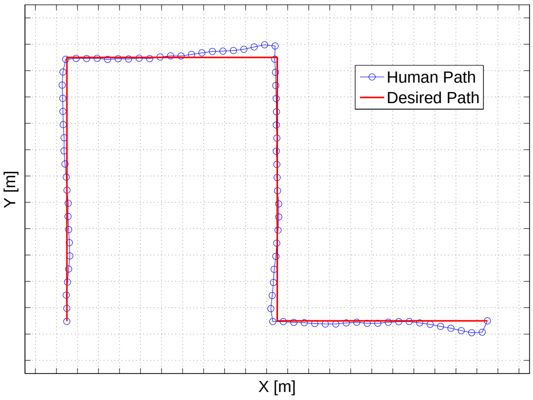


Figure 7.2: Human path with wrong initial orientation, $m = 0.9$ and $n = 0.4$.

proportional to the direction required for nulling the path distance by T_X (i.e. $sin^{-1}\frac{d_k}{v \cdot T_X}$). When the path is not captured by the phone camera, the user heading direction is corrected by an angle with constant module Δ, whose sign is positive (negative) if the path was lost on the left (right) side of the user. Being T the discrete time step of feedback and movement updates, we have:

$$\alpha_{k+1} = \begin{cases} \alpha_k + m(\beta - \alpha_k)^u + n \sin^{-1} \frac{d_k^u}{v \cdot T_X} & |d_k| \leq 50cm \\ \alpha_k + sign(d_k) \cdot \Delta \quad |d_k| > 50cm \end{cases}$$

$$d_{k+1} = d_k + v \cdot T \sin(\beta - \alpha_k)$$

where $(\beta - \alpha_k)^u$ and d_k^u represent the *human perception* of the signals $\beta - \alpha_k$ and d_k displayed at time k (that can be assumed equal to the real values plus an additive noise), and the coefficients m and n model the *human behavior*. Perception noises are generally assumed with zero mean, although a bias can be considered for taking into account the asymmetrical space perception of some users.

7.3.1 Examples of human behaviors

For visualizing the effects of different human perception and behavior models, we run some simulations in which at each time instant T the user coordinates are updated according to our human-in-the-loop model. All the simulations refer to the same path (the red lines plotted in figures 7.2-7.4) and have been obtained by setting $v = 0.5m/s$, $T = 1s$, $\Delta = \pi/10$. Perception noises and correction factors m and n have been used as configuration parameters for modeling different users.

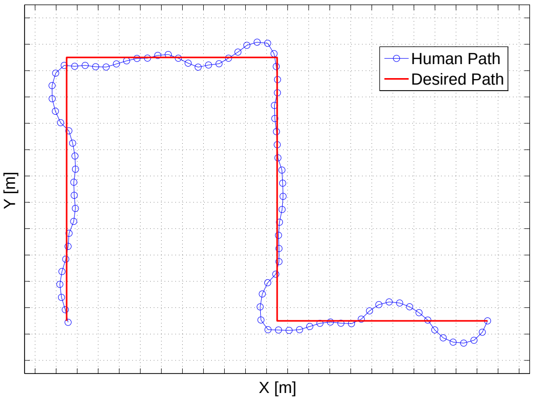


Figure 7.3: Human path with wrong initial orientation, $m = 0.6$ and $n = 0.4$.

Figure 7.2 shows the typical *normal* behavior, when the user starts its navigation from a point belonging to the path (the rightmost point of the figure) with a wrong orientation. Perception noise on the path direction has been assumed to be uniform in the range $[-\pi/15, \pi/15]$, while an additive Gaussian noise with zero mean and standard deviation equal to 5cm has been added to the perceived distance from the path. In this experiment, we set $m = 0.9$ and $n = 0.4$. Indeed, when the path is not lost (i.e. α_{k+1} is updated according to the $d_k \leq 50cm$ equation), we can easily study the system stability by considering that $\sin^{-1} d_k/(v \cdot T_x)$ can be linearized to $d_k/(v \cdot T_x)$. A good control design can be obtained if n is equal to about $m^2/(4T_x v)$. The figure clearly shows that after the transient phases to the wrong orientation (occurring at the beginning of the experiment and after each direction change), the user movements are almost on the path.

Since user real movements depend on the user strategy to follow the path and cannot be configured according to stability considerations, figure 7.3 plots the results of an experiment with non-optimal settings ($m = 0.6$ and $n = 0.4$). We can observe that the user is still able to reach the end of the path in an higher number of steps. In some cases, the user loses the path because the distance from the path is higher than 50cm. However, thanks to the second equation of the heading control ($d_k > 50cm$), the user is able to go back to the painted line.

Figure 7.4 shows another example in which corrections due to the distance from the path are almost neglected. In this case, the user reaches the end of the path by keeping an almost constant distance from the painted line. The distance is lower than 50cm, thus allowing to have continuous perception feedback. Finally, figure 7.5 shows a last example of human movements in presence of significant errors in the direction perception.

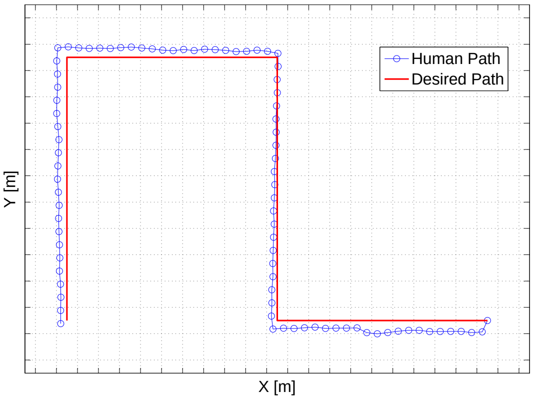


Figure 7.4: Human path when distance corrections are almost neglected ($m = 0.9$, $n = 0.01$).

7.4 Tracking System

Although our application has been designed for allowing users to follow a pre-defined path, we also envisioned the possibility to track the position of the users. To this purpose, we exploit not only the vision-based signals captured by the smartphone camera, but also a set of additional measurements provided by most smartphone models. The idea is calibrating or resetting the estimator of the user coordinates when the smartphone detects a reference point and integrating the information provided by the camera, compass, accelerometer, step counter, and so on, for updating the estimates when reference points are not available. The estimator is based on an Extended Kalman Filter in which we also include the user reaction to the phone signals in terms of an additive external signal on the state. Figure 7.6 shows the overall picture of the control system: on the basis of a state model describing the user movements and his reactions to the feedback signals, the measurements collected by the smartphone are filtered for producing an estimate of the user position and for helping in finding different destinations. When the user looses the path, vibration signals will drive him along a circular trajectory that permits to find again the instrumented path.

7.4.1 State Model

The paths are deployed on the floor as colored tapes, along which landmarks (e.g. QR codes) can be periodically applied for providing the absolute coordinates of the corresponding application point.

Let x and y be the 2D coordinates of the user, and $|v|$ and α the module and the direction of user velocity. We chose to model the velocity in terms of module and direction (rather than in terms of orthogonal components v_x and v_y) because in the previous section we assumed that the human correction actions work on the direction of the movement. Being the user reaction

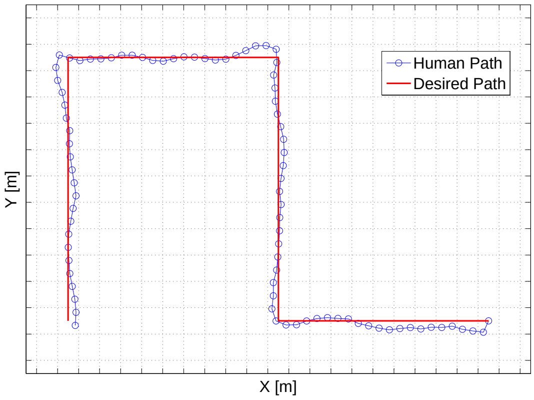


Figure 7.5: Human path with an perception direction noise uniformly distributed in $[-\pi/6, \pi/6]$.

signal u_k at time k equal to $m(\beta - \alpha_k)^u + n \sin^{-1} d_k^u / (v \cdot T_X)$ when the path line is visible to the phone and Δ when the path line is not visible, we can consider the following discrete-time state model:

$$\begin{cases} x_{k+1} & = x_k + |v|_k \cos(\alpha_k) T \\ y_{k+1} & = y_k + |v|_k \sin(\alpha_k) T \\ |v|_{k+1} & = |v|_k + w_{k+1}^{|v|} \\ \alpha_{k+1} & = \alpha_k + u_k + w_{k+1}^{\alpha} \end{cases} \tag{7.1}$$

where T is the update time interval, and $w_{k+1}^{|v|}$ and w_{k+1}^{α} are the state noise components. The additive noise on the velocity component represents the random variations in the pedestrian velocity that can be assumed as Gaussian distributed, while the noise on the user direction is given by the random fluctuations due to the real walking behavior of the user.

7.4.2 Measurement Model

The measurement model is based on the sensors available in the smartphone and on the information that can be inferred by the environment. Measurements are generally provided at regular time intervals T, but some specific components can be available only in some conditions (e.g. when the smartphone camera is able to read a landmark). More into details, we exploit the following measurements:

- *velocity* (z^v) : the user velocity is measured by some smartphone applications by exploiting the intertial sensors for counting the number of steps during an observation

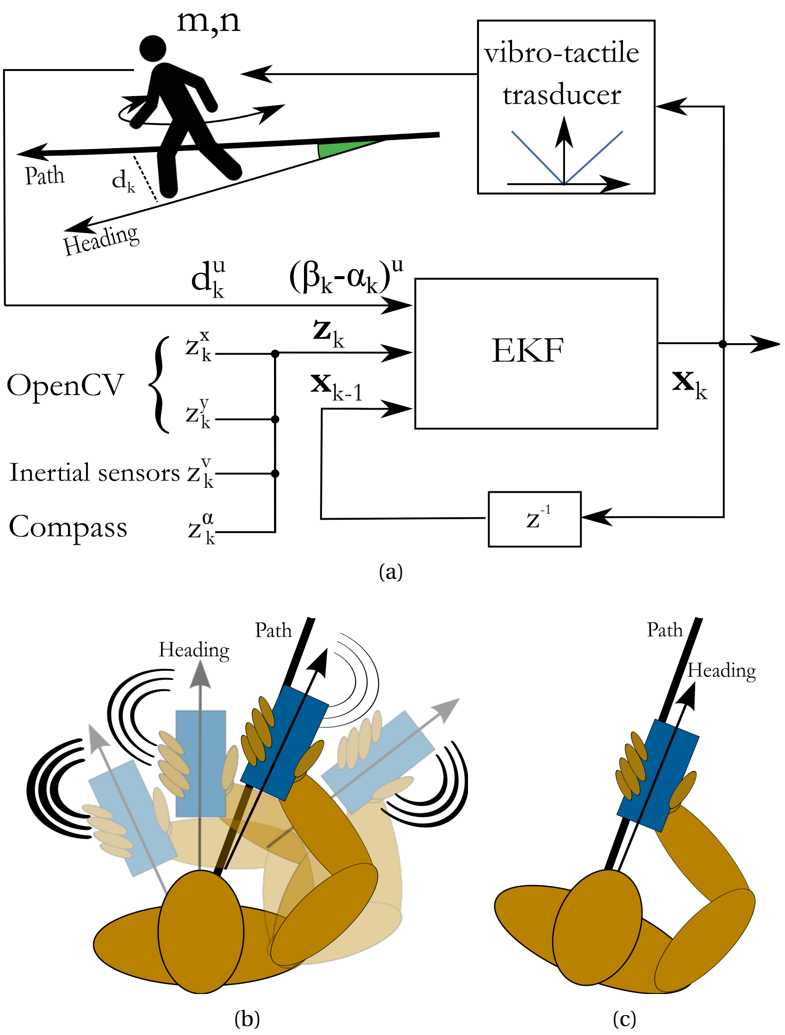


Figure 7.6: The block diagram of the tracking system with an Extended Kalman Filter (EKF) (a); exploring and detecting the best heading direction through arm movements (b); rotation of the body to follow the desired direction (c).

interval that can be assumed as an integer multiple of T. Such a mechanism requires to be calibrated to the user-specific step length. The measurement noise depends on the step detection sensors and on the approximation of fixed step length.

- *user heading (z^α)* : the direction of user movements is measured by means of the digital compass, which evaluates the user direction by referring to the south-north direction. The noise affecting this measurements is basically the compass noise. Additionally, computer vision techniques, based on optical flow concepts, can be used for providing another measurement of the heading direction (as well as another measurement of the user velocity module).
- *user coordinates (z^x and z^y)* : the user position can be read in the landmarks captured by the smartphone camera when they are visible in the current measuring interval

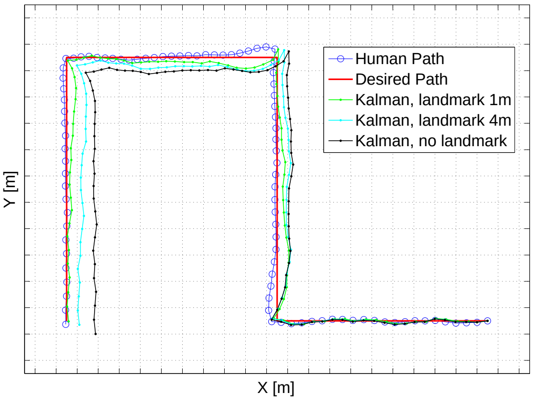


Figure 7.7: Kalman-based estimates of human position under different landmark scenarios.

> T. This measurement is not always available and is affected by a noise representing the difference between the user coordinates at the end of the T interval and the real landmark position.

Being v the generic noise on the measurement components, we can relate the measurements to the state vector by using the following equations:

$$\begin{cases} z^x_{k+1} & = x_{k+1} + v^x_{k+1} \\ z^y_{k+1} & = y_{k+1} + v^y_{k+1} \\ z^v_{k+1} & = |v|_{k+1} + v^{|v|}_{k+1} \\ z^\alpha_{k+1} & = \alpha_{k+1} + v^\alpha_{k+1} \end{cases} \tag{7.2}$$

where the apix of each noise component z has been explicitly related to the physical meaning of the relevant measurement.

7.4.3 Tracking Example

Figure 7.7 shows the results of three different experiments of user position estimation obtained with the same trace of real user movements and noise settings, under three different scenarios of landmark deployment. Specifically, the green curve refers to a scenario in which consecutive landmarks are spaced of 1m along the path, the cyan curve refers to a scenario in which landmark inter-space has been increased to 4m, and finally the black curve refers to a scenario without landmarks. Since the state model assumes that user direction is constant, the transient phases due to the user alignment on the path direction after each direction change are obviousy affected by some fluctuations. Moreover, while the direction estimate works well in all the cases, the accuracy of the position estimates degrade over time and cannot be improved without deploying some landmarks in the environment. Indeed, the user position estimated by the black curve at the end of a path whose overall length is 40m is about 1.5m far from the real position.

7.5 Summary

The ARIANNA system is a solution for helping autonomous navigation of visually impaired people with minimal deployment costs (the colored tapes on the floor) and very simple user interface. Despite of this simplicity, modeling the human walking behavior when navigation is assisted by ARIANNA is a challenging problem. Differently from robot, where feedback decisions can be driven by some state estimates based on optimal filtering of the environmental measurements, user decisions rely on his own perception of the current state and correction strategy.

In the chapter, we propose to model human navigation by assuming that user velocity is almost constant and the heading direction is proportionally corrected according to two metrics provided by ARIANNA: the misalignment between the heading direction and the path and the distance from the path. Even when the path is lost, vibration signals guide the user by indicating a circular path oriented in a direction opposite to the one in which the path has been lost, thus allowing to reach the end of the path under various settings of the perception noises and correction coefficients. Since the user always starts navigating from a known point on the path, we also consider tracking the user position during navigation. To this end, landmarks need to be deployed along to path line to avoid the accumulation of position errors.

8 Conclusions and future work

This thesis focuses on indoor localization systems based on Wi-Fi signals. We believe that Wi-Fi based indoor localization is following, for better or worse, the path traced by IEEE 802.11 evolution towards a universal solution for all possible operational scenarios. Unfortunately, heterogeneity and variability of usage contexts, hardware capabilities, and application requirements make this desirable universal solution difficult to emerge.

This thesis provides a twofold contribution: on one hand, it suggests to cease the quest for a one-size-fits-all localization method and provides a novel architectural solution to support multiple context-depending sensing strategies and localization algorithms. On the other hand, we designed, implemented and experimentally validated several new methodologies, along the whole spectrum of positioning working principles.

As for ToA, we provided a new method for better measuring the round trip time of Wi-Fi frames, taking into account errors introduced by multipath and automatic gain control adjustments. We then provided contribution in the Differential Time of Arrival (DToA) analyzing the timing of frames transmitted by the AP and its associated stations. A novel method was proposed to infer angle of arrival by RSSI measures, arranged in angular power profiles. We then exploited sensing information coming from the digital compass, to define panoramic and angular fingerprinting, respectively used for still localization and tracking. Finally, our novel vision-based method implements a path-follower mechanism to navigate visually impaired people.

Besides the scientific contributions to indoor positioning techniques provided by these methodologies, they have common requirements on: flexible MAC, flexible PHY, flexible use of sensors and actuators. This lead to the definition of a flexible indoor positioning system able to cognitively adapt to context dynamics. Our proposed architecture lies in the intersection among multiple enabling technologies: SDN as inspiring paradigm for a decoupled control plane, WMP for programmable MAC, SDR to enable tunable PHY and OSGi for modular positioning functions and algorithms.

Modular positioning subsystems are orchestrated according to the cognitive approach by exploiting radio, vision and inertial programmable *sensing devices*. These provide data to *positioning, tracking and network controllers* that after context estimation take configuration decisions, in the form of configuration rules. These policies are then applied to radio, vibrational, or even human *actuators*.

These elementary blocks provide well-defined interfaces and abstractions for the vision, radio and inertial platforms by means of modular and adaptable framework for networking and localization bundles.

The IPS architecture guarantees, by design, the coexistence between data transmission and location based applications as well as among multiple operators. This is enabled by virtualization capabilities of our architecture obtained through platform abstractions for sensing, computing, and enforcement functions.

This thesis provides a flexible indoor architecture, validated through multiple advances in the Wi-Fi based positioning field. It however provides a change of direction, therefore further effort is needed to refine abstractions, especially for non-network devices. Novel privacy and security issues could arise by the envisioned multi-tenancy positioning ecosystems holding user's data. Our approach promises a significant potential impact upon the main players of the indoor localization market: manufacturers, location-based service providers and final users.

Manufacturers would expose open APIs, protecting their investment and intellectual property beyond closed implementations without providing details about hardware and software internals. Service providers could approach a huge potential market of indoor positioning and provide adaptable solutions, optimized for the specific context, with limited investments on localization infrastructures. Final users should enjoy seamless indoor localization, which dynamically adapts to his needs.

A The WARP platform

A.1 Software and Hardware Specifications

WARP is a scalable and extensible programmable wireless platform, built from the ground up, to prototype advanced wireless networks. WARP combines high-performance programmable hardware with an open-source repository of reference designs and support materials [11].

The open-source repository is the software part of the WARP project and the two most developed reference designs are the 802.11 Reference Design and the WARPLab7 Reference Design. The first one is a real-time Field Programmable Gate Array (FPGA) implementation of the IEEE 802.11 Orthogonal Frequency-Division Multiplexing (OFDM) PHY and DCF MAC able to interact with commodity 802.11 devices, acting as an AP or as a Station (STA). The WARPLab7 Reference Design is an extensible framework that gives the user the flexibility to develop and deploy large arrays of nodes and allows for coordination of arbitrary combinations of single and multi-antenna transmit and receive devices making it a good tool for rapid physical layer prototyping.

The hardware of the WARP platform has revised several times since 2006 and, currently, the most updated hardware revision is the WARP v3 board. The latter is also the only platform able to run the 802.11 Reference Design and it is also the platform we use to implement WMP. The development of WMP is based on the 802.11 Reference Design.

Previous generations of WARP hardware ship with old FPGAs that are not compatible with the 802.11 Reference Design, which requires a Xilinx Virtex-6 FPGA [110]. The latest generation of WARP hardware integrates two programmable RF interfaces and a variety of peripherals. Figure A.1 gives a high level overview of the WARP v3 hardware design.

The configuration of the Xilinx Virtex-6 FPGA is volatile and must be downloaded every time the power is cycled. The FPGA design can be loaded in the form of a bitstream using three configuration methods: JTAG, SD card or SPI flash. The bitstream contains both the hardware part (IP blocks instantiated on the FPGA) and the software part of a reference design.

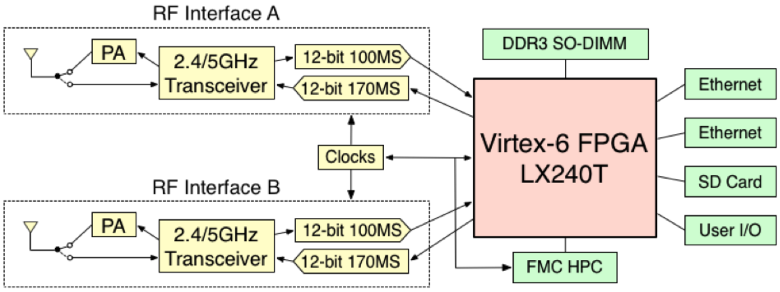


Figure A.1: WARP Hardware Peripherals Interconnections

In the 802.11 Reference Design the hardware definition includes the IP blocks that implement the 802.11 PHY and time critical MAC functionalities. The software definition includes drivers to control the peripherals integrated on the WARP v3 hardware and operations which are not time critical.

Xilinx Virtex-6 is connected to various peripherals including:

- Clocking resources;
- RF interfaces;
- Memory peripherals;
- FMC Expansion slot;
- Ethernet ports;
- User I/O peripherals;

Clocking function in WARP v3 is very flexible and maybe one of the most complex aspects of WARP v3 hardware. Figure A.2 reports an overview of WARP v3 clocking resources.

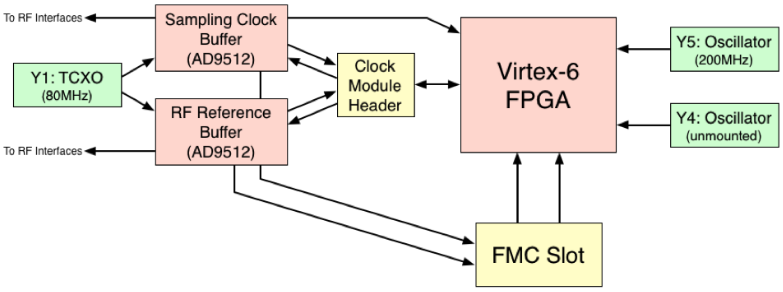


Figure A.2: WARP v3 Clocking

Two oscillators are connected directly to the FPGA, indicated with Y4 and Y5. Y5 is a 200MHz LVDS oscillator and Y4 is not mounted but can be added by the end-user if needed.

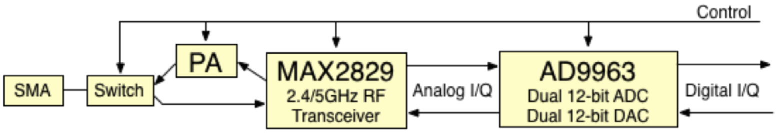


Figure A.3: WARP v3 RF Interfaces

Multiple clock connections to and from the FMC slot can be used to extend WARP hardware capabilities through FMS expansion modules. The RF interface clocking design is centered on two AD9512 2-to-5 clock buffers: one to manage the sampling clock and the second for the RF reference clock. Both AD9512 buffers have SPI interfaces connected to dedicated FPGA pins for configuration at run time.

In order to configure the clock buffers the FPGA must work as an SPI master and this task is performed by means of the $w3\_clock\_controller$ IP block included in the 802.11 Reference Design that can be used to manage both clock buffers from user code in processor-based design in the FPGA.

The WARP v3 board includes also a header for connecting clock signals to external equipments. This header could be used to share clocks between nodes or drive custom clock frequencies from an external source. The WARP v3 board integrates two identical RF interfaces that are in turn composed by many IC blocks and whose basic structure is illustrated in figure A.3.

The conversion between the analog and digital I/Q domains is handled by the Analog Devices AD9963 MxFE. It integrates two 100MSps 12-bit ADCs, two 170MSps 12-bit DACs, interpolation and decimation filters and programmable analog gain and offset adjustments.

The AD9963 is very flexible and includes a register bank to control various functions on the chip accessible through a dedicated SPI interfaces by means of the $w3\_ad\_controller$ IP block provided by the 802.11 Reference Design. The digital I/Q ports on the AD9963 operate at double data rate, with I/Q interleaved. The 802.11 Reference Design provides the $w3\_ad\_bridge$ IP block to connect these DDR ports to separate I/Q buses in user design. The WARP v3 RF interfaces use the Maxim MAX2829 transceiver to translate between baseband and RF both for 2.4 and 5GHz Tx/Rx paths.

As for the other components of the RF interfaces the MAX2829 transceiver has a number of digital control lines and a dedicated SPI interface for internal register access. These interfaces can be managed in user design by means of the $radio\_controller$ IP block. Each transceiver generates its own RF carrier signal, derived from the reference clock input by the AD9512 clock. The MAX2829 requires either a 20MHz or 40MHz reference clock so the AD9512 is configured to divide its 80MHz input to generate the desired reference frequency.

The WARP v3 design currently uses the Anadigics AWL6951 dual-band power amplifier and every board is tested to ensure >20dBm output power at both 2.4 and 5GHz.

Memory peripherals includes a DDR3 SO-DIMM slot, a 128 Kbit IIC EEPROM, an SD card slot and a 128 Mb SPI flash device. The DDR3 SO-DIMM RAM memory slot is routed to dedicated pins on the FPGA and when it is combined with a DDR3 memory controller core in the FPGA it is available to user design. For example, in the 802.11 reference Design, the SO-DIMM controller interface is designed to support modules up to 8GB in size and run at up to 400MHz.

The 128 Kbit ICC EEPROM is a non-volatile device able to retain its data indefinitely even when the WARP is switched off. The full EEPROM is readable and writable from user application by means of the $w3\_icc\_eprom$ IP block provided by the 802.11 Reference Design. By default some board specific values are written to the top few bytes of the EEPROM; these values include the Ethernet interfaces MAC addresses, the Tx DC offset values calibrated during manufacturing, the board serial number and the FPGA unique identifier.

The WARP v3 board includes also an SD card slot that supports standard SD cards up to 2 GB in size and a 128 Mb SPI flash device. Both these peripherals can be used as an alternative way to configure the FPGA at power up without an external JTAG cable.

WARP v3 hardware include also two Ethernet interfaces each one connected to dedicated FPGA pins for totally independent operation. Both Ethernet ports are connected to a Marvell 88E1121R dual Ethernet PHY that implements two tri-speed Ethernet PHY cores. The FPGA interfaces with the Marvell IC through two hard TEMAC cores, which implement a tri-speed Ethernet MAC compatible with the Marvell PHY on the WARP v3 board.

Finally, the WARP v3 board includes a variety of user I/O for observing and interacting with designs at run time: there are 12 LEDs, 3 push buttons, 2 4-position DIP switches and 2 seven-segment displays. All these elements are accessible from user designs through the $w3\_userio$ IP block provided by the 802.11 Reference Design.

A.2 802.11 Reference Design

As reported in the previous section, the 802.11 Reference Design is a real-time FPGA implementation of the IEEE 802.11 OFDM PHY and DCF MAC able to interact with commodity 802.11 devices, acting as an Access Point (AP) or as a station (STA). The reference design is completely open source and researchers can modify it to explore extensions and improvements to the standard MAC and PHY and test their extensions in networks of real 802.11 devices.

The WMP port to WARP is based on the 802.11 Reference Design, therefore its high level architecture and its main building blocks are described in this section. Before digging into the details of the 802.11 Reference Design, we introduce some terminology and general concepts concerning FPGA based system development. This might be useful for better understanding the rest of the Section.

A.2.1 Terminology and General Concepts for FPGA-based Systems

The core of the WARP v3 board is the Xilinx Virtex-6 FPGA. The Field-Programmable Gate Array (FPGA) is a semiconductor device that can be programmed after manufacturing. Instead of being restricted to any predetermined hardware function, an FPGA allows the developer to program features and reconfigure the hardware for specific applications even after the product has been installed in the field.

An FPGA may perform all the logical functions that can be implemented by an Application-Specific Integrated Circuit (ASIC), with the extended ability to update such functionalities. In the case of WARP v3 it is possible to run different reference designs in different times and build MAC and PHY layers prototypes. The FPGA configuration is generally specified using a Hardware Description Language (HDL) that is compiled through a synthesis process into the FPGA loadable bitstream. Though, the typical workflow does not require specifying the FPGA configuration from scratch using a HDL because there are many logical function that are shared almost among every FPGA configuration (e.g., memory modules, clock generator, bus controllers) and that are provided by FPGA vendors through libraries. Moreover, also custom logical functions can be inserted into libraries in order to easily reuse them when needed.

Logical functions provided by libraries can be called in different ways and the most common are IP blocks, FPGA cores or simply cores. An FPGA configuration is composed by a set of these IP blocks: inserting a block in an FPGA configuration is generally known as “instantiation”.

The typical workflow to build an FPGA configuration requires to define the set of composing IP blocks, the configuration parameters for each of these IP blocks (I/O ports, operational frequencies and so on), how they are connected together and how they interface to the external (peripherals). The tools provided by the FPGA vendor synthesize the specified configuration into the FPGA loadable bitstream.

As for the WMP port for the WARP platform, there are three main aspects that are worth to be discussed. First, special IP blocks implement a CPU and are able to run general-purpose software. The CPU interacts with others IP blocks instantiated on the FPGA or with external peripherals connected to its pins. The only CPUs of the WARP boards are software-implemented in the FPGA, therefore CPU blocks become very useful to increase the board flexibility. The 802.11 Reference Design uses a two-processor architecture to implement the 802.11 MAC layer. The tools provided by the FPGA vendor generate a bitstream that includes both the hardware and the software design that runs on these CPUs.

The second aspect is related to drivers, which are software modules associated with IP blocks. Drivers are used to simplify controlling IP block instances. Usually, vendors provide both the IP block and its driver. Drivers are very useful when the corresponding IP block implements a complex logical function. In such a case, the IP block includes many hardware registers, requires complex procedures to initialize, reset or shutdown the logical function or to activate the functionalities it provides. Drivers provide a simple way for accessing and controlling the hardware registers, procedure execution and functionalities activation and are generally provided in the form of software libraries. Drivers make the development work easier and the developers can ignore internal details of IP blocks.

Finally, FPGAs may access and control the external peripherals through special IP blocks called controllers. Controllers can be very complex therefore they are usually provided with corresponding drivers. In the example illustrated in Figure A.4, in order to control the external peripheral A (access hardware registers, or control FPGA PINs) the software that runs on the CPU uses functions provided by the driver to control controller A, which is implemented with an IP block. Then controller A translates the commands coming from the CPU in the appropriate procedures to control the peripheral A.

The WARP project provides the controllers and the corresponding drivers for every peripheral installed on the WARP board.

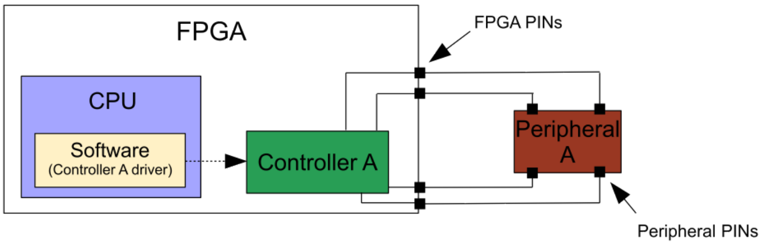


Figure A.4: External peripherals access

A.2.2 High Level Architecture

The 802.11 Reference Design contains two implementations: one for APs and one for glsplsta. The Reference Design is implemented entirely in the FPGA of the WARP v3 node and its high level architecture is shown in Figure A.5. PHY processing is divided across two custom cores ($wlan\_phy\_tx$ and $wlan\_phy\_rx$) that interact with WARP hardware by means of the controller cores. The MAC is implemented primarily in software running in two MicroBlaze CPUs. A support IP core provides accurate inter-packet timing ($wlan\_mac\_dcf\_hw$).

The main IP blocks that compose the 802.11 Reference Design are the following:

- *CPU High* is a software implemented MicroBlaze micro-architecture. It is dedicated to the execution of the top-level MAC code and other high level functions. The code in the CPU High is responsible for constructing all non-control packets for transmission and for implementing a number of manage frame handshakes with nodes (probe request/response, association request/response, etc.). CPU High is also responsible for bridging the wireless domain with the wired one, implementing encapsulation and de-encapsulation of Ethernet frames according to the wired-wireless integration described in Annex P of the IEEE 802.11-2012 standard. CPU High is clocked at 160MHz;

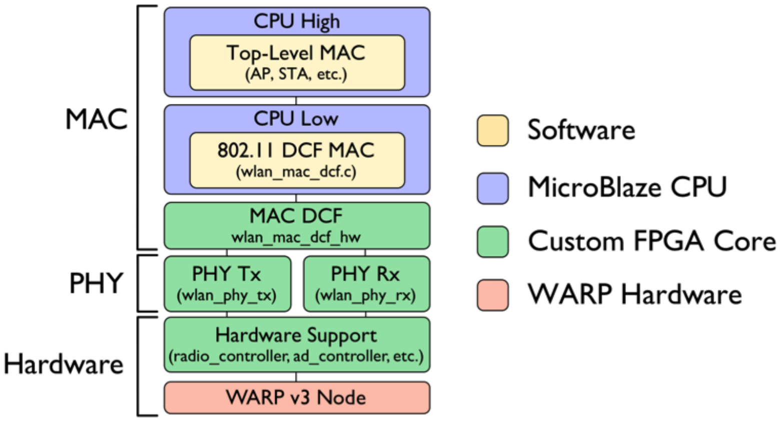


Figure A.5: WARP 802.11 Reference Design Architecture

- *CPU Low* is a MicroBlaze CPU that executes low-level code for the MAC Distributed Coordination Function (DCF). This is clocked at 160MHz and is responsible for all MAC-PHY interactions and for handling time critical operations including ACK transmission, backoff scheduling, maintaining the contention window and initiating re-transmissions. The $wlan\_mac\_dcf$ software project implements Section 9.3.1 of the 802.11-2012 standard [66];

- *MAC DCF core* ($wlan\_mac\_dcf\_hw$) is a custom FPGA core that interfaces the MAC software design and the Tx/Rx PHY cores. It implements the timers required by DCF (timeout, backoff, Distributed (coordination function) Inter-Frame Space (DIFS), Short Inter-Frame Space (SIFS), etc) and the various carrier sensing mechanisms. MAC DCF core monitors Tx and Rx PHY cores and sequences Tx and Rx events based on the configuration provided by the MAC software;

- *PHY Tx/Rx* are IP cores that implement the OFDM physical layer transceiver specified in Section 18 of the 802.11-2012 standard [66]. PHY cores are clocked at 160MHz (8x the I/Q sample rate);

- *Hardware Support* are controllers. These IP cores permit the code in CPU Low to control various peripheral interfaces on WARP v3.

Many IP blocks that are required by the 802.11 Reference Design are not shown in Figure A.5. These are standard cores provided by Xilinx, the most important ones are: Buffer Random Access Memory (BRAM) cores for storing compiled software and frame queues, and Mutex/Mailbox cores for handling inter-process communication.

Interconnections between the main components of the 802.11 Reference Design are depicted in figure A.6. As packets move through the 802.11 Reference Design, their content must be accessed by CPU High, CPU Low and the PHY cores.

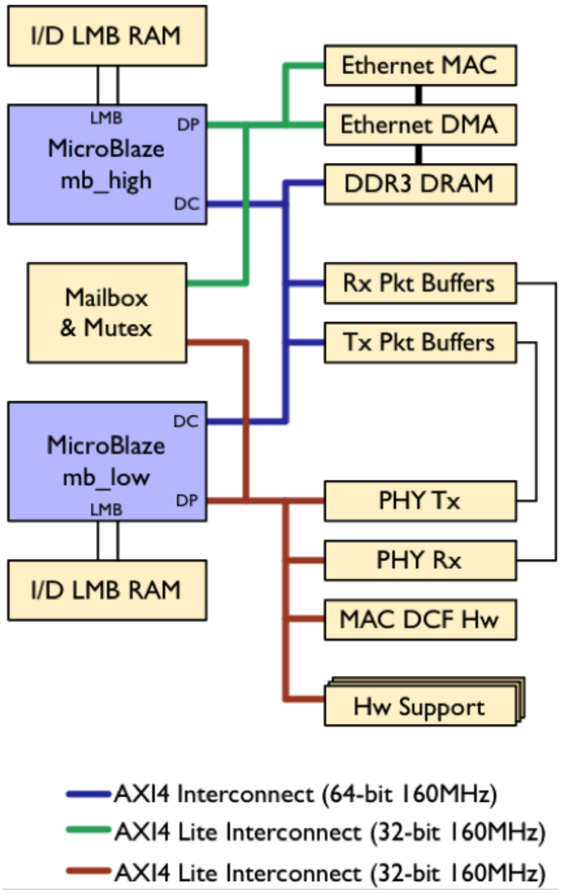


Figure A.6: WARP 802.11 Reference Design FPGA Interconnects

Some of the depicted IPs are connected to hardware peripherals described in Section 2, which are not illustrated for the sake of simplicity.

The low-level MAC running in CPU Low handles one packet at a time. The high-level MAC in CPU High manages many packets at once via a series of queues stored in DDR3 DRAM. In the AP reference implementation one queue is created per associated node plus one queue for all broadcast traffic. In the STA reference implementation, instead, there is only one high-level MAC queue.

Whenever the low-level MAC finishes transmission of a packet the next available packet is dequeued from the appropriate queue and passed to CPU Low for transmission. The station implementation in the Mango 802.11 Reference Design can associate and communicate with 802.11 Access Points (WARP or others) and the traffic source or sink for the station is Ethernet interface that can be accessed by the process running in CPU High.

Each MicroBlaze has access to two AXI interconnects. For both CPUs the MicroBlaze DP port (non-cached peripheral memory access port) is connected to an AXI4 Lite interconnect. The peripheral cores connected to each AXI4 Lite interconnect are accessible by only one CPU.

The cores are divided between CPUs based on which part of the MAC needs to access them. For example the $radio\_controller$, $w3\_ad\_controller$ and PHY configuration registers are all attached to the interconnect for CPU Low. Similarly the Ethernet cores are attached to the peripheral bus for CPU High. The mailbox and mutex ports for each CPU are also attached to their corresponding peripheral buses. The MicroBlaze DC (cached memory access port) for both CPUs are attached to a shared AXI4 interconnect.

The data cache is disabled in the Reference Design; memory access via the DC ports is routed to a slave memory device via the AXI4 interconnect. Both CPU DC ports are masters on this interconnect. The primary slave devices on this interconnect are the block RAMs used to implement the 802.11 frame buffers (Rx pkt buffer and Tx pkt buffer). Both CPUs can read and write any location in both memories.

The AXI4 interconnect is a 64-bit crossbar clocked at 160MHz, which provides sufficient throughput to avoid contention between the CPUs. The 802.11 Reference Design uses two dual-port 64KB RAMs as Tx and Rx packet buffers. One port of each RAM is attached to a BRAM interface controller (axi_BRAM_ctrl), that maps the RAM onto the address space of the two CPUs. The other port of each RAM is attached directly to the corresponding PHY core.

These direct PHY connections do not traverse an AXI interconnect. Instead both PHY cores (Tx and Rx) implement native 64-bit BRAM interfaces in logic. The PHY cores divide each 64KB BRAM into 16 4KB buffers. The PHY identify these buffers through their id and the low-level MAC code provides the packet buffer index to the PHY for each Tx and Rx event. A 32-entry mutex is used to avoid contention for Tx and Rx packet buffers between CPUs. One mutex entry corresponds to one packet buffer.

CPU High locks a Tx packet buffer while it prepares a packet for transmission. It unlocks the buffer when it notifies CPU Low that the new packet is ready for transmission. CPU Low locks the Tx buffer while it awaits the PHY transmission, unlocking the buffer when it notifies CPU High of completion. CPU Low locks the Rx packet buffer into which the Rx PHY is writing received packets. When the PHY notifies the MAC a packet has been received without errors, CPU Low locks another Rx buffer and configures the Rx PHY to begin receiving new frames there.

CPU Low unlocks the buffer containing the valid received frame and notifies CPU High. CPU High locks the packet buffer and processes the received packet. When processing is finished, it unlocks the Rx packet buffer, allowing CPU Low to use it for a future reception. All the notifications between CPU High and CPU Low are performed by means of the mailbox core. Given that Virtex-4 does not support the double processor architecture and features like mailboxes, the 802.11 Reference Design is not backward compatible with WARP v2 platform.

A.2.3 MAC Layer

The MAC implementation is based on DCF and is composed by two parts: the *Upper-level MAC* is responsible for inter-packet actions that are not time critical and it is executed by CPU High; the *Lower-level MAC* is responsible for intra-packet actions that are time critical and it is executed by CPU Low.

The lower level MAC interfaces directly to the PHY Tx and Rx cores and handles all wireless transmissions and receptions. Minimizing processing latency in the lower level MAC is critical in order to meet the 802.11 channel access timing requirements.Minimal state is maintained at this level. Only the contention window and station retry counters are stored across packet transmission events. All other states (AP vs. STA, association state, etc.) are maintained by the upper layer MAC.

The MAClet manager has been easily integrated in the Upper-level MAC code; it is just a separate module that intercepts every frame to check if it is a WMP control frame or not. The situation is more complex in the case of WMP engine and API implementation in CPU Low.

The Lower-MAC implementation has been completely replaced because the way WMP works is not compatible with the MAC API implementation provided by the 802.11 Reference Design.

MAC API, as designed in the 802.11 Reference Design, are not usable by WMP engine. To justify this assertion we take as example two fundamental WMP actions: frame transmission and reception. Figure A.7 shows how the frame transmission is implemented in the 802.11 Reference Design.

Every packet provided by the upper level MAC for transmission initiates a new software Tx process (Frame Tx SW) in the lower level MAC. When the lower level MAC software is ready to submit a frame for wireless transmission it passes control to the $wlan\_mac\_dcf\_hw$ core in the FPGA (Frame Tx HW).

This core implements a state machine that is compatible with the channel access timing requirements of the 802.11 DCF. The core transmits the frame as soon as the DCF protocol determines a transmission opportunity (Frame Tx HW Done?) and returns the result of the transmission to the MAC software (Frame Tx HW Result?). Figure A.8 shows the implementation of frame reception (software level) in the 802.11 Reference Design.

When the PHY receives a valid frame it notifies the $wlan\_mac\_dcf\_hw$ core via status signals in the FPGA. The core then notifies the MAC software via status bits in a register.

The lower level MAC software then executes the action that handles the reception (Frame Rx). The $wlan\_mac\_dcf\_hw$ core includes a small state machine, which can initiate a PHY transmission in response to a valid PHY reception. This state machine enables transmission of ACK packets fast enough to meet the SIFS timing requirement of the 802.11 DCF.

If the MAC header of an incoming frame indicates an ACK should be transmitted in response,

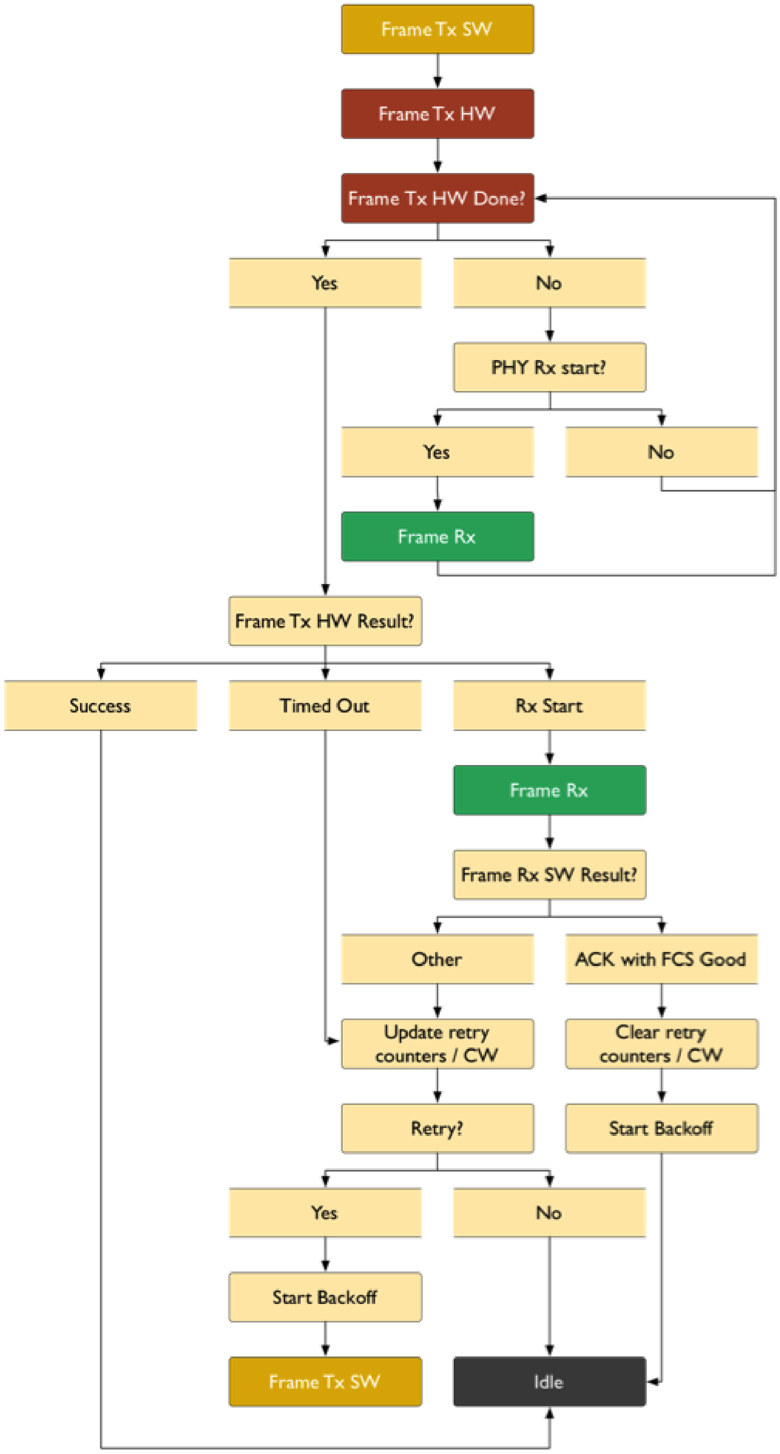


Figure A.7: 802.11 Reference Design Tx Software implementation

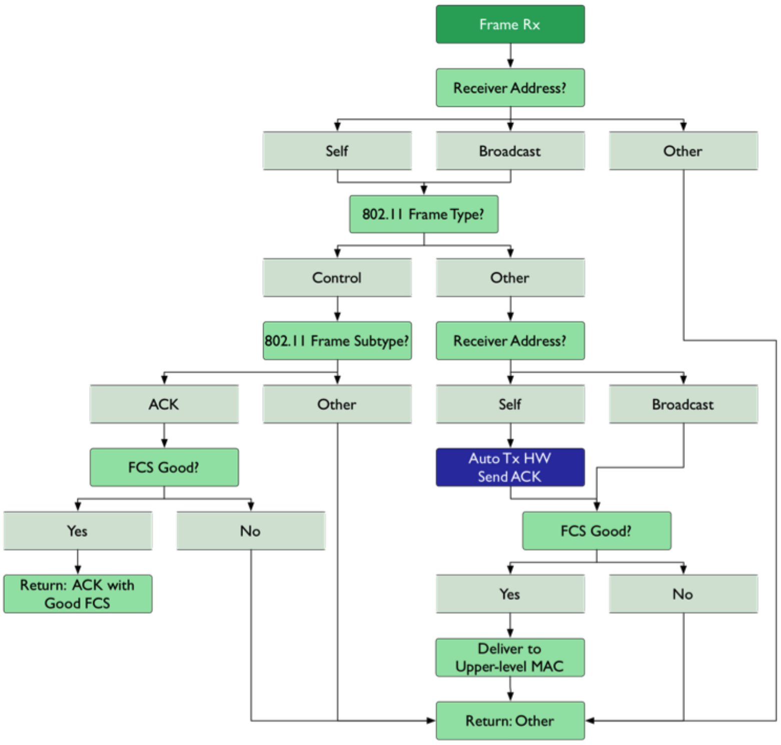


Figure A.8: 802.11 Reference Design Rx Software implementation

the lower level MAC code prepares an ACK frame in a spare Tx packet buffer. The code then configures the Auto Tx state machine to enable a post-Rx transmission from the ACK buffer if the current reception completes without errors (Auto Tx HW Send ACK).

If the reception ends with errors the ACK is not transmitted and the Tx packet buffer is recycled. The preparation of the ACK frame occurs while the PHY is still receiving the packet: the lower level MAC software must ensure ACK is ready before the reception completes. The Auto Tx state machine initiates a transmission only if enabled before the last byte of the incoming frame is decoded.

After the ACK frame has been scheduled, the lower level MAC code loops until all the bytes of the incoming frame have been received and only at that time the reception action returns. The WMP design requires that actions and procedures verify conditions and events atomically; the only loop of the system has to be the WMP engine.

This requirement simplifies the design and the development of WMP engine, but the APIs implemented in the 802.11 Reference Design does not satisfy it and therefore, are not usable in WMP implementation. In fact, from the examples provided above, it appears that the transmission action loops until the $wlan\_mac\_dcf\_hw$ core returns.

This action calls itself recursively to reschedule a frame transmission for retries. Also the reception action has similar problems: it loops until all the bytes of the incoming frame

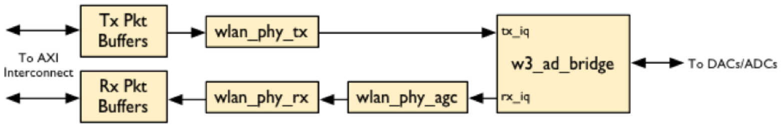


Figure A.9: PHY high level architecture

have been received, executes another action (ACK scheduling) and can be nested inside the transmission action. This kind of issues is similar to other MAC APIs provided by the 802.11 Reference Design.

A.2.4 PHY Layer

In this section we briefly describe them main structure and the main features of the PHY level implemented in the 802.11 Reference Design. Given that in the current implementation of the WMP we kept the PHY layer provided by Reference Design, the description that follows reflects the actual PHY capabilities of the WMP. The physical layer implementation, whose high level architecture is shown in figure A.9, is based on the OFDM PHY specified in section 18 in [66].

The physical layer implementation of the 802.11 Reference Design is divided across three FPGA cores:

1. $wlan\_phy\_tx$ that implements the OFDM transmitter;
2. $wlan\_phy\_rx$ that implements the OFDM and DSSS receiver;
3. $wlan\_phy\_agc$ thet implements the Automatic gain control (AGC).

At one end the PHY cores interface directly to the Analog to Digital Converters (ADCs) and Digital to Analog Converters (DACs) on the WARP v3 hardware via the $w3\_ad\_bridge$ core. These interfaces exchange complex sample streams at 20MHz. At the other end the cores connect to the Tx and Rx packet buffers.

The packet buffers are implemented as dual-port BRAMs, one port dedicated to PHY access and the other one tied to the AXI interconnect for granting access to the Microblaze CPUs.

The main PHY specifications are the following:

- Clock frequency: 160MHz;
- Bandwidth: 20MHz;
- OFDM format: 64 subcarriers (48 data, 4 pilots), 16-sample cyclic prefix;
- Frame format: as specified in section 18.3.2 of 802.11-2012: Preamble (10 repetitions of 16-sample short training symbol, 2.5 repetitions of 64-sample long training symbol),

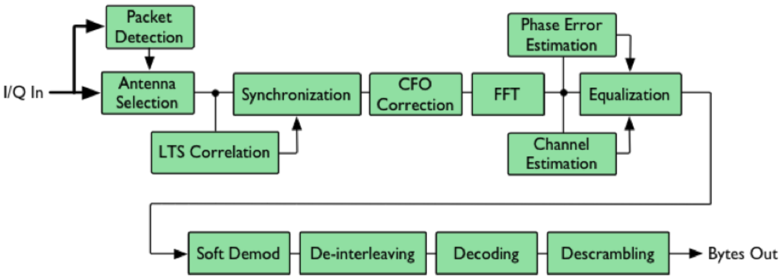


Figure A.10: PHY Rx diagram

SIGNAL field as first OFDM symbol (3 bytes as BSPK, rate 1/2 code), Remaining OFDM symbols filled with SERVICE field (2 bytes) and payload (up to 2048 bytes);

- Rates: The following OFDM data rates are implemented. Each data rate is realized by a combination of modulation and coding rates.
- Multi-antenna Support: The current PHY Tx/Rx pipelines are SISO. The PHY antenna interfaces implement selection diversity across the two RF interfaces on WARP v3 hardware. The antenna selection is made per packet. For transmissions the antenna selection is always controlled by C code in CPU Low. For receptions the PHY can automatically select the higher-SNR antenna based on the AGC gain selections. Alternatively the C code in CPU Low can force the selection of the receiving antenna.

Modulation	Code Rate	Data Rate (Mbps)
BPSK	1/2	6
BPSK	3/4	9
QPSK	1/2	12
QPSK	3/4	18
16-QAM	1/2	24
16-QAM	3/4	36
64-QAM	2/3	48
64QAM	3/4	54

Figure A.10 depicts the receiver architecture whose main features are:

- Packet Detection: implements two packet detection schemes: simple energy detection based on RSSI and auto-correlation of the I/Q samples searching for the preamble STS, based on the well-known Schmidl-Cox algorithm. When selection diversity is enabled parallel packet detectors are enabled so that either antenna can trigger a detection;
- Antenna Selection: automatic selection of which I/Q stream feeds the rest of the PHY pipeline, using AGC gain selections as an indicator of received SNR;

- LTS Correlation: cross correlator searching for the 64-sample LTS in the preamble. The two LTS correlation peaks establish timing for the rest of the reception, marking the boundary of each OFDM symbol fed into the FFT;

- Synchronization: a dual-port circular sample buffer records all incoming samples. Once the LTS correlator establishes sample-level timing the buffer begins reading samples into the FFT using the correlation timing to set the boundary of each OFDM symbol;

- CFO Correction: the carrier frequency offset (CFO) is estimated before the FFT by comparing the phases of identical samples in the two LTS. The CFO is estimated by averaging the 64 phase comparisons, then removed by multiplying the I/Q samples by the output of a DDS; v FFT: translates the time domain received samples into the frequency domain. Each FFT consumes 64 time domain samples and produces 64 frequency domain samples. The boundary of each FFT is established by the synchronization blocks above. The cyclic prefix of each OFDM symbol is removed by advancing the boundary of each FFT 16 samples per transform.

- Channel Estimation: a complex channel coefficient is calculated for each non-zero subcarrier by averaging the estimates from the two LTS;

- Phase Error Estimation: the pilot tones embedded in each OFDM symbol are used to calculate a phase error estimate per OFDM symbol. Every subcarrier in the OFDM symbol is then de-rotated by the estimated phase error;

- Equalization: the channel estimates and phase-corrected data symbols are fed into the equalizer to remove amplitude and phase errors incurred by propagation through the wireless channel. The current implementation uses a simple zero-forcing equalizer, dividing each subcarrier by the corresponding channel coefficient and using the same channel coefficients for the full packet;

- Soft Demod: each data symbol is then demodulated to a soft value per coded bit;

- De-Interleaving: the coded bits, represented as soft 4-bit confidence values, are de-interleaved along OFDM symbol boundaries using the interleaving pattern specified in the standard;

- Decoding: the de-interleaved soft values are decoded using a standard Viterbi decoder;

- Descrambling: the de-coded bits are finally descrambled using the LFSR specified in the standard;

All logic in the 802.11 receiver FPGA core is clocked at 160MHz and supports a maximum bandwidth of 20MHz (clock rate = 8x max sample rate). The PHY receiver also implements the 1Mbps DSSS rate specified in the original 802.11 standard (section 16.2 of the 802.11-2012 standard).

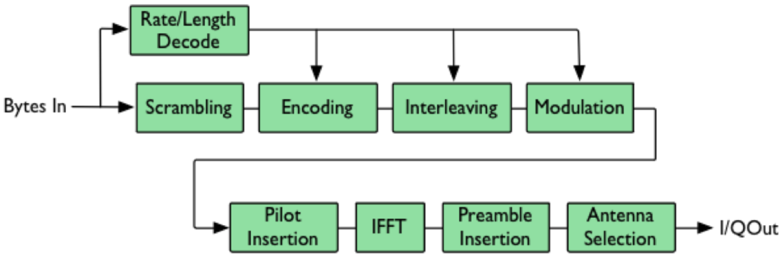


Figure A.11: PHY Tx diagram

The receiver allows reception of management frames transmitted by 802.11 devices at 1Mbps which are still very common in many networks. For example, Beacon and Probe Request frames are frequently transmitted at 1Mbps by commercial devices.

The basic STA/AP association handshake requires reception of these frames. The 802.11 Reference Design does not implement a DSSS transmitter, as modern 802.11 devices are able to receive management frames at higher rates (including 6Mbps, the lowest OFDM rate, which is commonly used for management frames at 5GHz).

Figure 11 shows the transmitter architecture whose main features are:

- Rate/Length Decode: the length and modulation/coding rates are stored in the first 3 bytes of the packet, part of the 802.11 SIGNAL field. The Tx core uses these values to configure the relevant blocks per packet;
- Scrambling: payload bits are scrambled to avoid long runs of constant values;
- Encoding: payload bits are encoded by a standard 1/2 rate convoLookUp Table (LUT)ional encoder and optionally punctured to rates 2/3 or 3/4, depending on the selected coding rate;
- Interleaving: coded bits are interleaved in blocks along OFDM symbol boundaries;
- Modulation: the coded bits are mapped on to complex values using the selected modulation scheme. The modulated symbols are then mapped on to the data-bearing subcarriers;
- Pilot Insertion: four pilot tones, represented by BPSK symbols with scrambled signs, are mapped onto the dedicated subcarriers in each OFDM symbol;
- IFFT: the IFFT translates 64 frequency domain samples into 64 time domain samples. A 16-sample cyclic prefix is added by repeating the last 16 IFFT output samples for each OFDM symbol;
- Preamble Insertion: the standard 320-sample preamble is prepended to the IFFT output;

- Antenna Selection: the complete waveform is finally transmitted via the selected RF interface;

All logic in the 802.11 transmitter FPGA core is clocked at 160MHz and supports a maximum bandwidth of 20MHz (clock rate = 8x max sample rate).

A.3 WMP on WARP

This section describes both the hardware (FPGA configuration) and software architecture used to implement WMP on WARP boards. The implementation of WMP on WARP is based on the architecture of the 802.11 Reference Design, which has been modified in order to satisfy the WMP requirements. In what follows we refer to the STA implementation because it provides the simplest behavior of the high-level MAC.

However, both STA and AP behaviors are implemented in the CPU high and therefore it is mostly independent from the WMP implementation. The WMP API is the same used on the WMP implemented for Broadcom NICs [38]. The MAC programmer that designs XFSMs does not perceive the hardware-related complexity because this is hidden by the API implementation. For example, handling RX/TX buffers or recovering from HW error (phy block) are tasks that do not require any attention by the MAClet programmer.

In this section we discuss the high level architecture of WMP on WARP introducing at first the FPGA configuration changes in comparison with the original Reference Design and then the software changes that take advantage from the new hardware IP blocks.

Several details of the WMP on WARP implementation, including new memory blocks, FSM storage, the WMP engine workflow, the use of LUTs, and the MAClet Manager definition can be found in [111].

A.3.1 High Level Architecture

This section describes the hardware and software high level architecture of WMP. The hardware architecture of WMP is deeply based on the hardware architecture of the 802.11 Reference Design, so we limit the description to the new IP blocks. The PHY level is implemented by the original blocks developed by Mango Communication Inc. as part of the 802.11 Reference Design.

Current implementation includes an 802.11g compliant PHY whose primitives (Reception, Transmission, Power management, ...) are accessible through the software drivers that control those IP blocks. WMP software accesses the hardware primitive through the interface exposed by these drivers: for this reason, a change in one of these IP blocks will not affect WMP software layer unless the drivers interface changes.

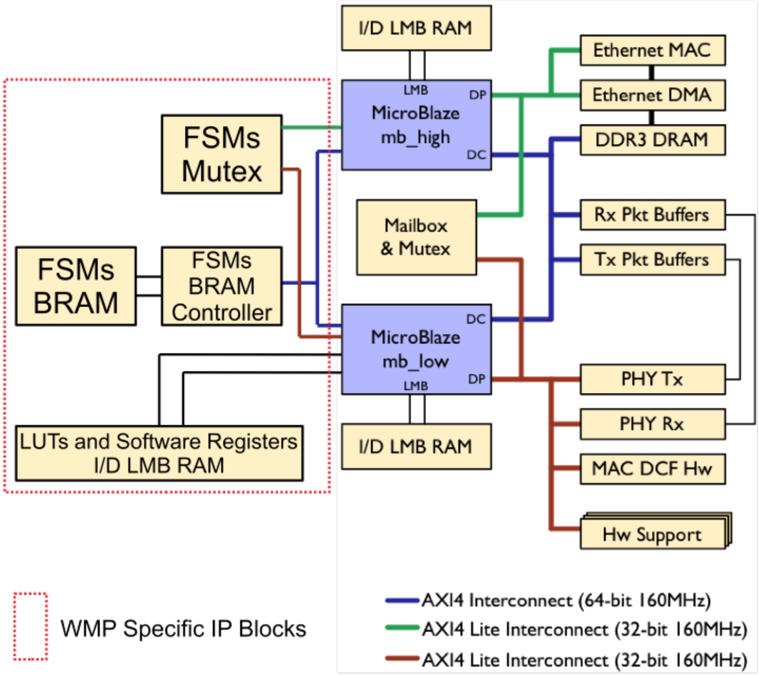


Figure A.12: WMP on WARP Hardware Architecture

After the WMP specific IP blocks description we introduce the WMP high level software and how each software module is connected to each other and to the hardware.

Hardware Architecture

Figure A.12 shows how the FPGA cores of the 802.11 Reference Design have been extended in order to implement the WMP framework. The Figure extends a previous one and the dotted red rectangle contains WMP specific IP blocks.

In figure it is reported a simplified version of the FPGA instantiated hardware. Clock generators and bus controllers are not depicted, as well as connections to and from the FPGA package PINs.

The WMP implementation required four new IP blocks: the FSMs Mutex, the FSMs BRAM, the FSMs BRAM Controller and the LUTs and Software registers I/D LMB RAM. None of these new IP block has been designed from scratch: they are all provided, in fact, by the Xilinx IP Standard Library. What makes them particular is the way they are used by WMP. The following list describes each of these new IP block, their interconnections with other IP blocks and their role in the WMP system.

FSMs Mutex: this is a Mutex IP and it is used for inter-processor communication and synchronization. In particular, it is used to synchronize the access to the FSMs BRAM between the two MicroBlaze processors. It is composed by eight mutex slots, each of which is used to synchronize the access to one of the eight FSM slots in the FSMs BRAM. If the process running on one of the two processors (for instance, mb_low) needs to access (read and/or write) the i-th FSM slot of FSMs BRAM, it must acquire the i-th mutex slot.

Only when the state register of the i-th mutex slot reports that it is owned by mb_low, the process running on this processor can access the i-th FSM slot.

Otherwise, if the state register of the i-th mutex slot reports that is has not been possible to acquire it (for example due to hardware error) or that the i-th mutex slot is busy because currently owned by the other processor, the process running on this processor can not access the i-th FSM slot.

The Mutex for FSMs is an FPGA core that can be used to implement inter-processor communication and synchronization, but it does not implement any hardware level synchronization. This means that a process running on one of the two processor can always access the content of the FSMs BRAM i-th FSM slot, but if it accesses the i-th FSM slot before acquiring the corresponding mutex slot then the coherence of the i-th FSM slot data is not guaranteed (e.g., the process running on mb_low reads data from the i-th FSM slot while the process running on mb_high is modifing it).

It is up to the software developer to follow the synchronization protocol: before any write or read operation on the i-th buffer he must be sure that the owner of the i-th mutex slot is the processor on which the software is executed.

Finally, the FSMs Mutex has two AXI4 interfaces that are used to connect the Mutex IP to the two MicroBlaze Processor. In particular, one interface is connected to the controller of the AXI4 bus whose master is the MicroBlaze mb_high and the other one is connected to the controller of the AXI4 bus whose master is the MicroBlaze mb_low. In both cases we have used existing AXI4 bus adding the FSMs Mutex as a new slave node.

FSMs BRAM Controller: this is an AXI BRAM Controller IP and it is used to connect one or more RAM blocks to an AXI BUS. It also represents the access interface to the controlled RAM blocks. Any read/write operation is performed by means of the BRAM controller and never directly on the RAM block. FSMs are stored in BRAM and are accessible by mb_low and mb_high. The FSMs BRAM Controller is connected as a slave node to the AXI4 Controller of a bus shared between the two MicroBlaze processors.

FSMs BRAM: this is a Block RAM (BRAM) IP. It acts as a configurable memory module that can be accessed by means of the FSMs BRAM Controller. The FSMs BRAM is used to store the bytecodes of the XFSMs that are ready to be executed. This memory is divided in eight different slots (FSM slots) each one able to contain one XFSM.

LUTs and Software registers I/D LMB RAM: it is a Block RAM (BRAM) IP core. Unlike the FSMs BRAM, it is connected to the controller of the LMB BUS whose master node is the MicroBlaze mb_low.

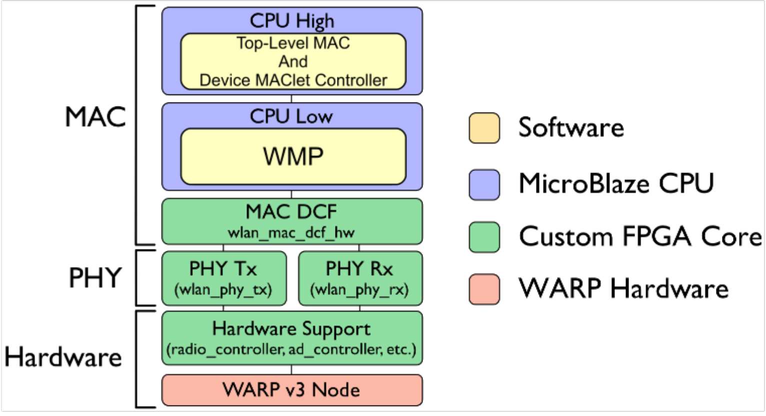


Figure A.13: WMP on WARP Software Architecture

The controller in turn is connected to the mb_low ILMB and DLMB ports. The peculiarity of this kind of BRAM is that they are reserved to contains the data section, the code section and the stack and heap sections of the running process. The WMP uses this kind of BRAM to store LUTs and software registers.

Software Architecture

Figure A.13 reports the high-level software architecture of the WMP implementation for the WARP platform.

It is similar to the architecture shown in Figure A.5 because the WMP keeps using the same drivers (MAC DCF, PHY TX, PHY RX and Hardware Support) as in the original 802.11 Reference Design and maintains the two processors: the CPU Low runs software that interacts with the PHY level and the CPU High runs software that implements higher level functionalities (e.g, association/disassociation logic, Ethernet frame handling,...).

The main difference with the original 802.11 Reference Design concerns the software that is executed by the two processors. We use as a starting point the STA implementation of the top-level MAC provided by the 802.11 Reference Design in order to simplify the debug process in the first phase of development. In fact, the AP implementation offered by the 802.11 Reference Design starts generating traffic at the end of the boot process (Beaconing) and this behavior makes the test/debug activity more difficult. Instead, with the STA implementation we have a complete control over the traffic generated and this simplifies the design and the execution of test cases.

The STA implementation has been extended to include what we called the Device MAClet Controller (DMC). The DMC has three responsibilities: 1) it manages incoming frames containing XFSMs and WMP commands (e.g., run the XFMS identified by label x); 2) it handles the XFSM slots in FSMs BRAM and writes new XFSMs in the appropriate slots; 3) it manages the switches to new XFSMs.

Regarding the CPU Low, the original DCF MAC implementation was completely replaced by the WMP software architecture.

The new code implements the MAC engine. It is a protocol-agnostic element that may run different protocols using the primitives offered by the PHY level. Moreover, several actions implemented in the 802.11 Reference Design DCF contained loops. For instance, the reception action waited for the reception of a frame to complete and the transmission action waited for the transmission of the frame to successfully complete. In the new WMP code, only states can contain loops waiting for something, whereas actions are atomic.

‘

Bibliography

[1] A. Küpper, "Front Matter," in *Location-Based Services.* John Wiley & Sons, Ltd, 2005, pp. i–xix.

[2] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[3] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th annual international conference on Mobile computing and networking.* ACM, 2009, pp. 261–272.

[4] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2004, pp. 1012–1022.

[5] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.

[6] R. Akl, "Indoor propagation modeling at 2.4 GHz for IEEE 802.11 networks," Ph.D. dissertation, UNIVERSITY OF NORTH TEXAS, 2005.

[7] K. El-Kafrawy, M. Youssef, A. El-Keyi, and A. Naguib, "Propagation modeling for accurate indoor WLAN RSS-based localization," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd.* IEEE, 2010, pp. 1–5.

[8] MaXentric Technologies LLC, "Calradio," 2014. [Online]. Available: http://calradio.calit2.net/calradio1.htm

[9] Ettus Research, "Universal software radio peripheral." [Online]. Available: http://www.ettus.com

[10] K. Tan, H. Liu, J. Zhang, Y. Zhang, J. Fang, and G. M. Voelker, "Sora: high-performance software radio using general-purpose multi-core processors," *Communications of the ACM*, vol. 54, no. 1, pp. 99–107, 2011.

[11] WARP Project - Wireless Open-Access Research Platform. [Online]. Available: http://warpproject.org/trac

[12] M. C. Ng, K. E. Fleming, M. Vutukuru, S. Gross, H. Balakrishnan *et al.*, "Airblue: A system for cross-layer wireless protocol development," in *Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems.* ACM, 2010, p. 4.

[13] G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, and I. Tinnirello, "MAClets: active MAC protocols over hard-coded devices," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies.* ACM, 2012, pp. 229–240.

[14] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "Softmac-flexible wireless research platform," in *Proc. HotNets-IV*, 2005.

[15] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling mac protocol implementations on software-defined radios." in *NSDI*, vol. 9, 2009, pp. 91–105.

[16] J. Ansari, X. Zhang, A. Achtzehn, M. Petrova, and P. Mahonen, "A flexible mac development framework for cognitive radio systems," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE.* IEEE, 2011, pp. 156–161.

[17] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Wireless MAC processors: Programming MAC protocols on commodity hardware," in *INFOCOM, 2012 Proceedings IEEE.* IEEE, 2012, pp. 1269–1277.

[18] P. De Mil, B. Jooris, L. Tytgat, J. Hoebeke, I. Moerman, and P. Demeester, "snapmac: A generic mac/phy architecture enabling flexible mac design," *Ad Hoc Networks*, vol. 17, pp. 37–59, 2014.

[19] P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, I. Tinnirello, and G. Bianchi, "Wireless MAC processor networking: A control architecture for expressing and implementing high-level adaptation policies in WLANs," *IEEE Vehicular Technology Magazine*, vol. 8, no. 4, pp. 81–89, Dec. 2013.

[20] I. Tinnirello, P. Gallo, P. Loreti, and al., "Deliverable 2.2.1 - Architecture Specification," The FLAVIA Consortium, Tech. Rep., 2011.

[21] J. Medved, A. Tkacik, R. Varga, and K. Gray, "OpenDaylight: Towards a Model-Driven SDN Controller architecture," in *2014 IEEE 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun. 2014, pp. 1–6.

[22] T. Gu, H. Pung, and D. Zhang, "Toward an OSGi-based infrastructure for context-aware applications," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 66–74, Oct. 2004.

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

[24] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[25] D. Erickson, "The beacon openflow controller," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013, pp. 13–18.

[26] "Floodlight: Open Source Software for Building Software-Defined Networks."

[27] "NOX." [Online]. Available: http://www.noxrepo.org/nox/about-nox/

[28] "POX." [Online]. Available: http://www.noxrepo.org/pox/about-pox/

[29] Ryu SDN Framework Community, "Ryu - component-based software defined networking framework." [Online]. Available: http://osrg.github.io/ryu/

[30] Open Networking Lab, "Open Network Operating System (ONOS)." [Online]. Available: http://onosproject.org/

[31] Cisco, "Cisco Extensible Network Controller - An OpenDaylight-based controller," Cisco, Tech. Rep., 2011.

[32] X. Zhang, J. Ansari, G. Yang, and P. Mahonen, "Trump: Supporting efficient realization of protocols for cognitive radio networks," in *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*. IEEE, 2011, pp. 476–487.

[33] F. Vázquez Gallego, J. Alonso-Zarate, C. Liss, and C. Verikoukis, "OpenMAC: A new reconfigurable experimental platform for energy-efficient medium access control protocols," *IET Science, Measurement Technology*, vol. 6, no. 3, pp. 139–148, May 2012.

[34] P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, I. Tinnirello, and G. Bianchi, "Wireless MAC Processor Networking: A Control Architecture for Expressing and Implementing High-Level Adaptation Policies in WLANs," *IEEE Vehicular Technology Magazine*, vol. 8, no. 4, pp. 81–89, Dec. 2013.

[35] R. S. Hall, K. Pauls, S. McCulloch, and D. Savage, *OSGi in Action - Creating Modular Applications in Java*. Manning, 2011.

[36] M. Youssef, M. Mah, and A. Agrawala, "Challenges: device-free passive localization for wireless environments," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 2007, pp. 222–229.

[37] J. Wilson and N. Patwari, "Radio Tomographic Imaging with Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 5, pp. 621–632, May 2010.

[38] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Wireless mac processors: Programming mac protocols on commodity hardware," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1269–1277.

[39] P. R. Grønsund, “Cognitive Radio from a Mobile Operator’s Perspective: System Performance and Business Case Evaluations,” Ph.D. dissertation, University of Oslo, 2013.

[40] A. Varshavsky, E. de Lara, J. Hightower, A. LaMarca, and V. Otsason, “GSM indoor localization,” *Pervasive and Mobile Computing*, vol. 3, no. 6, pp. 698–720, Dec. 2007.

[41] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, “Zero-configuration, robust indoor localization: Theory and experimentation,” University of Illinois, Tech. Rep., 2005.

[42] IEEE, “P802.11v - amendment to standard for information technology – telecommunications and information exchange between systems – local and metropolitan networks – specific requirements – part ii: Wireless lan medium access control (mac) and physical layer (phy) specifications: Ieee 802.11 wireless network management,” IEEE, Tech. Rep., 201.

[43] F. Institute, “Awiloc positioning system.” [Online]. Available: http://www.iis.fraunhofer.de/en/ff/lok/tech/feldstaerke/rssi.html

[44] “Cisco wireless control system.” [Online]. Available: http://www.cisco.com/c/en/us/products/wireless/wireless-control-system/index.html

[45] M. Ciurana Adell *et al.*, “Contributions to toa-based location with wlan,” Ph.D. dissertation, Departament d’Enginyeria Telematica, 2010.

[46] D. Humphrey and M. Hedley, “Super-resolution time of arrival for indoor localization,” in *Communications, 2008. ICC’08. IEEE International Conference on.* IEEE, 2008, pp. 3286–3290.

[47] A. Günther and C. Hoene, “Measuring round trip times to determine the distance between wlan nodes,” in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems.* Springer, 2005, pp. 768–779.

[48] K. Ahmed and G. Heidari-Bateni, “Wsn06-3: Improving two-way ranging precision with phase-offset measurements,” in *Global Telecommunications Conference, 2006. GLOBECOM ’06. IEEE*, Nov 2006, pp. 1–6.

[49] D. Giustiniano and S. Mangold, “Caesar: carrier sense-based ranging in off-the-shelf 802.11 wireless lan,” in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies.* ACM, 2011, p. 10.

[50] S. Bancroft, “An algebraic solution of the gps equations,” *Aerospace and Electronic Systems, IEEE Transactions on*, no. 1, pp. 56–59, 1985.

[51] S. A. Golden and S. S. Bateman, “Sensor measurements for wi-fi location with emphasis on time-of-arrival ranging,” *Mobile Computing, IEEE Transactions on*, vol. 6, no. 10, pp. 1185–1198, 2007.

[52] C. Hoene and J. Willmann, "Four-way toa and software-based trilateration of ieee 802.11 devices," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on.* IEEE, 2008, pp. 1–6.

[53] M. Ciurana, F. Barceló, and S. Cugno, "Multipath profile discrimination in toa-based wlan ranging with link layer frames," in *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization.* ACM, 2006, pp. 73–79.

[54] R. Krishna and C. Hoene, "Calculating relative clock drifts using ieee 802.11 beacons," in *India Conference (INDICON), 2009 Annual IEEE.* IEEE, 2009, pp. 1–4.

[55] B. Sieka, "Active fingerprinting of 802.11 devices by timing analysis," in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, vol. 1. IEEE, 2006, pp. 15–19.

[56] G. Lackner, U. Payer, and P. Teufl, "Combating wireless lan mac-layer address spoofing with fingerprinting methods," *International Journal of Network Security*, vol. 9, pp. 164–172, Sept 2009.

[57] G. Lackner and P. Teufl, "Ieee 802.11 chipset fingerprinting by the measurement of timing characteristics," in *Proceedings of the Ninth Australasian Information Security Conference-Volume 116.* Australian Computer Society, Inc., 2011, pp. 41–50.

[58] The WMP team. Wireless mac processor.

[59] C. Zhang, M. Kuhn, B. Merkl, M. Mahfouz, and A. E. Fathy, "Development of an uwb indoor 3d positioning radar with millimeter accuracy," in *Microwave Symposium Digest, 2006. IEEE MTT-S International.* IEEE, 2006, pp. 106–109.

[60] T. C. Karalar, "Implementation of a localization system for sensor networks," DTIC Document, Tech. Rep., 2006.

[61] K. I. Ahmed and G. Heidari-Bateni, "Wsn06-3: Improving two-way ranging precision with phase-offset measurements," in *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE.* IEEE, 2006, pp. 1–6.

[62] P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, and I. Tinnirello, "Wmps: A positioning system for localizing legacy 802.11 devices," *Transactions on Smart Processing and Computing*, 2012.

[63] R. Dobbins, "Software defined radio localization using 802.11-style communications," Ph.D. dissertation, WORCESTER POLYTECHNIC INSTITUTE.

[64] D. Niculescu and B. Nath, "Vor base stations for indoor 802.11 positioning," in *Proceedings of the 10th annual international conference on Mobile computing and networking.* ACM, 2004, pp. 58–69.

[65] M. Ciurana, F. Barceló-Arroyo, and I. Martín-Escalona, "Comparative performance evaluation of ieee 802.11 v for positioning with time of arrival," *Computer Standards & Interfaces*, vol. 33, no. 3, pp. 344–349, 2011.

[66] I. S. for Information technology, "Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," IEEE, Tech. Rep., February 2012.

[67] B. R. Mahafza, *Radar systems analysis and design using MATLAB*. CRC press, 2000, ch. Chapter 6, Matched filter and Radar ambiguity function, p. 552.

[68] S. Stein, "Algorithms for ambiguity function processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 3, pp. 588–599, 1981.

[69] P. Gallo and S. Mangione, "RSS-eye: Human-assisted Indoor Localization without Radio Maps," in *To appear in Proceedings of International Conference on Communication ICC*, 2015.

[70] P. Gallo, S. Mangione, and G. Tarantino, "Widar: Bistatic wi-fi detection and ranging for off-the-shelf devices," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, June 2013, pp. 1–6.

[71] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proceedings of the International Conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 787–790.

[72] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2000, pp. 775–784 vol.2.

[73] Z. Yang, Z. Zhou, and Y. Liu, "From rssi to csi: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 25:1–25:32, Dec. 2013.

[74] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system." in *NSDI*, 2013, pp. 71–84.

[75] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA '02. New York, NY, USA: ACM, 2002, pp. 105–111.

[76] M. Malajner, P. Planinsic, and D. Gleich, "Angle of arrival estimation using rssi and omnidirectional rotatable antennas," *Sensors Journal, IEEE*, vol. 12, no. 6, pp. 1950–1957, June 2012.

[77] S. Sen, R. R. Choudhury, and S. Nelakuditi, "Spinloc: Spin once to know your location," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*. ACM, 2012, p. 12.

[78] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng, "I am the antenna: accurate outdoor ap location using smartphones," in *Proceedings of the 17th annual international conference on Mobile computing and networking.* ACM, 2011, pp. 109–120.

[79] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 53–53, Jan. 2011.

[80] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14. New York, NY, USA: ACM, 2014, pp. 483–494.

[81] B. Li, B. Li, J. Salter, A. G. Dempster, and C. Rizos, "Indoor positioning techniques based on wireless lan," *LAN, FIRST IEEE INTERNATIONAL CONFERENCE ON WIRELESS BROADBAND AND ULTRA WIDEBAND COMMUNICATIONS*, pp. 13–16, 2007.

[82] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piche, "A comparative survey of wlan location fingerprinting methods," in *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, March 2009, pp. 243–251.

[83] Wigle (wireless geographic logging engine). [Online]. Available: https://wigle.net/

[84] Evarilos eu project. evaluation of rf-based indoor localization solutions for the future internet. www.evarilos.eu.

[85] N. Roy, H. Wang, and R. R. Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Proceedings of the 12th international conference on Mobile systems, applications, and services, ACM*, 2014.

[86] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '13. New York, NY, USA: ACM, 2013, pp. 249–262.

[87] D. Croce, P. Gallo, D. Garlisi, L. Giarre, S. Mangione, and I. Tinnirello, "ARIANNA: A smartphone-based navigation system with human in the loop," in *2014 22nd Mediterranean Conference of Control and Automation (MED)*, Jun. 2014, pp. 8–13.

[88] The cyanogenmod aftermarket firmware. [Online]. Available: http://www.cyanogenmod.org/

[89] D. Potts, G. Steidl, and M. Tasche, "Fast fourier transforms for nonequispaced data: A tutorial," in *Modern sampling theory.* Springer, 2001, pp. 247–270.

[90] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *Proceedings of the 18th annual international conference on Mobile computing and networking.* ACM, 2012, pp. 305–316.

[91] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, “ARIADNE: a dynamic indoor signal map construction and localization system,” in *Proceedings of the 4th international conference on Mobile systems, applications and services*. ACM, 2006, pp. 151–164.

[92] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, “The user as a sensor: navigating users with visual impairments in indoor spaces using tactile landmarks,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 425–432.

[93] S. Willis and S. Helal, “Rfid information grid and wearable computing solution to the problem of wayfinding for the blind user in a campus environment,” in *Proceedings of the ninth annual IEEE International Symposium on Wearable Computers, Osaka, Japan*. Citeseer, 2005.

[94] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, “Indoor human navigation systems: A survey,” *Interacting with Computers*, vol. 25, no. 1, pp. 21–33, 2013.

[95] S. Chen, “Kalman filter for robot vision: a survey,” *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 11, pp. 4409–4420, 2012.

[96] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin, “Cyber Physical System Challenges for Human-in-the-Loop Control,” in *USENIX*, vol. 8th International Workshop on Feedback Computing, 2013.

[97] J. M. Loomis, R. G. Golledge, and R. L. Klatzky, “Navigation system for the blind: Auditory display modes and guidance,” *Presence: Teleoperators and Virtual Environments*, vol. 7, no. 2, pp. 193–203, 1998.

[98] S. Holland, D. R. Morse, and H. Gedenryd, “Audiogps: Spatial audio navigation with a minimal attention interface,” *Personal and Ubiquitous Computing*, vol. 6, no. 4, pp. 253–259, 2002.

[99] R. Etter and M. Specht, “Melodious walkabout: Implicit navigation with contextualized personal audio contents,” *Adjunct Proceedings of the Third International Conference on Pervasive Computing,*, 2005.

[100] V. Hayward, O. R. Astley, M. CruzHernandez, D. Grant, and G. RoblesDeLaTorre, “Haptic interfaces and devices,” *Sensor Review*, vol. 24, no. 1, pp. 16–29, Mar. 2004.

[101] E. Samur, “Systematic evaluation methodology and performance metrics for haptic interfaces,” Ph.D. dissertation, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2010.

[102] J. M. Loomis, “Digital map and navigation system for the visually impaired,” *Unpublished manuscript, Department of Psychology, University of California, Santa Barbara*, 1985.

[103] The MoBIC Project. [Online]. Available: http://isgwww.cs.uni-magdeburg.de/projects/mobic/mobicuk.html

[104] M. Solazzi, W. Provancher, A. Frisoli, and M. Bergamasco, “Design of a SMA actuated 2-DoF tactile device for displaying tangential skin displacement,” in *2011 IEEE World Haptics Conference (WHC)*, Jun. 2011, pp. 31–36.

[105] T. Amemiya, J. Yamashita, K. Hirota, and M. Hirose, “Virtual leading blocks for the deaf-blind: a real-time way-finder by verbal-nonverbal hybrid interface and high-density RFID tag space,” in *IEEE Virtual Reality, 2004. Proceedings*, Mar. 2004, pp. 165–287.

[106] G. Park and S. Choi, “Perceptual space of amplitude-modulated vibrotactile stimuli,” in *2011 IEEE World Haptics Conference (WHC)*, Jun. 2011, pp. 59–64.

[107] W. Heuten, N. Henze, S. Boll, and M. Pielot, “Tactile wayfinder: A non-visual support system for wayfinding,” in *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges*, ser. NordiCHI ’08. New York, NY, USA: ACM, 2008, pp. 172–181.

[108] M. Pielot and R. d. Oliveira, “Peripheral vibro-tactile displays,” in *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services.* ACM Press, 2013, pp. 1–10.

[109] “Martin pielot - how the phone’s vibration alarm can help to save battery.” [Online]. Available: http://pielot.org/2012/12/how-the-phones-vibration-alarm-can-help-to-save-battery/

[110] Xilinx, “Virtex-6 Family Overview, Product Specification,” Tech. Rep., January 2012.

[111] Facchi, N., Gallo, P., and Gringoli, F., “Project Deliverable D7.6.3-intermediate Final report on results of CNIT experiment and user experience (Enhanced layer-2 functionalities for experiment design),” Sep. 2013.

Glossary

ADC Analog to Digital Converter. 105

AGC Automatic Gain Control. 22

AoA Angle of Arrival. 2, 9, 10

AP Access Point. 2, 5, 14, 21, 70, 71, 93, 98

API Application Programming Interface. 2, 12, 15, 17, 20, 22

ASIC Application-Specific Integrated Circuit. 97

BRAM Buffer Random Access Memory. 99, 101, 105, 110–112

CPS Cognitive Positioning Systems. 22

CR Cognitive Radio. vii, 19, 22, 23

DAC Digital to Analog Converter. 105

DCF Distributed Coordination Function. 19, 21, 102

DfPL Device-free Passive Localization. 21

DIFS Distributed (coordination function) Inter-Frame Space. 99

DoA Direction of Arrival. 9

DToA Differential Time of Arrival. 2, 8

FPGA Field Programmable Gate Array. 93–99, 102, 105, 107, 109–111

GPS Global Positioning System. 1

HDL Hardware Description Language. 97

IPS Indoor Positioning System. 2, 10, 13, 15, 19, 91

ISP Internet Service Provider. 19

LoS Line of Sight. 20

LUT LookUp Table. 108–112

MAC Medium Access Control. 2, 10–12, 14, 15, 17–20, 22, 91

MAFLIP Modular Architecture for FLexible Wi-Fi Networking and Indoor Positioning systems. 14

MDNM Model Driven Network Management. 15

MDSE Model Driven Software Engineering. 15

NLoS Non Line of Sight. 20

nLoS near Line of Sight. 20

OFDM Orthogonal Frequency-Division Multiplexing. 93, 96, 99, 105–108

OSGi Open Service Gateway initiative. 2, 17, 20, 91

PHY Physical Layer Protocol. 2, 10, 11, 15, 18, 19, 91

PPI Panoramic Power Image. viii, 70, 71, 73

PTC Positioning and Tracking Controller. 15

REST REpresentational State Transfer. 17

RSSI Received Signal Strength Indication. 10, 21, 69–73, 77

RToF Roundtrip Time of Flight. 7

SAL Service Adaptation Layer. 17

SDN Software-defined Network. vii, 2, 14–17, 23, 91

SDR Software-defined Radio. 17, 19, 22, 91

SIFS Short Inter-Frame Space. 99

SOCAM Service-Oriented Context-Aware Middleware. 15, 20

STA Station. 93

TBTT Target Beacon Transmission Time. 21

TDM Time Division Multiplexing. 19

ToA Time of Arrival. 2, 7, 17, 22

VoIP Voice over Internet Protocol. 19

WARP Wireless Open-Access Research Platform. 19

WLAN Wireless Local Area Network. 5, 11

WMP Wireless MAC Processor. 11, 12, 14–16, 18–22, 91, 97, 109

WSN Wireless Sensor Network. 16

XFSM eXtended Finite State Machine. 12

XNC Extensible Network Controller. 17