Indoor Location : study on the IEEE 802.11 Fine Timing Measurement standard

Jérôme Henry

DOCTORAT BRETAGNE LOIRE MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE
MINES-TÉLÉCOM ATLANTIQUE BRETAGNE
PAYS-DE-LA-LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Jérôme HENRY

Indoor Location : Study on the IEEE 802.11 Fine Timing Measurement Standard

Evaluation of FTM Efficiency and Reliability for Indoor Deployments

Thèse présentée et soutenue à Cesson Sévigné, le 10 Décembre 2021
Unité de recherche : IRISA
Thèse N° : 2021IMTA0272

Rapporteurs avant soutenance :

Gentian Jakllari, Professeur, Toulouse INP-ENSEEIHT
Nathalie Mitton, Directrice de Recherche, INRIA Lille-Nord Europe

Composition du Jury :

Présidente :	Nathalie Mitton, Directrice de Recherche, INRIA Lille-Nord Europe
Examinateurs :	Romaric Ludinard, Maître de Conférence, IMT Atlantique
	Gentian Jakllari, Professeur, Toulouse INP-ENSEEIHT
	Franck Rousseau, Maître de Conférence, Grenoble INP-ENSIMAG / LIG
	Valérie Renaudin, Directrice de Recherche, Université Gustave Eiffel
Dir. de thèse :	Nicolas Montavont, Professeur, IMT Atlantique
Co-dir. de thèse :	Yann Busnel, Professeur, IMT Atlantique

Invitée :

Isabelle Guerin-Lassous, Professeure, Université Claude Bernard Lyon 1

Acknowledgement

Je tiens à remercier:

Jean-Loup Gélard, qui en 1998 me suggéra une idée simple: si tu en sais plus que ceux qui t'entourent, cherche d'autres compagnons. Cette volonté d'approfondir mon savoir et d'échanger avec ceux qui sont plus avancés que moi m'habite toujours et m'a conduit ici. Repose en paix, Jean-Loup.

Yann Busnel, Romaric Ludinard et Nicolas Montavont, de l'IMT Atlantique, qui ont patiemment dirigé cette thèse, ont su guider sans forcer, laisser la recherche se faire en balançant rigueur et patience, et ont su finalement me conduire jusqu'à cette page tout en me laissant penser que je conduisais moi-même.

Robert Barton, qui ne me laisse pas m'endormir sur mes lauriers, et est toujours là pour suggérer de nouvelles pistes, de nouveaux problèmes à résoudre.

I would like to thank:

Jean-Loup Gélard, who in 1998 suggested a simple idea: if you know more than anyone around you, then look for other people to surround yourself with. This will to deepen my knowledge, and exchange with those who have gone further than I have is still within me, and brought me here. Rest in peace, Jean Loup.

Yann Busnel, Romaric Ludinard et Nicolas Montavont, from IMT Atlantique, who have patiently directed this thesis, knew how to guide without pushing, let the research happen while balancing rigor and patience, and eventually drove me to the page while allowing me to think that I was the one driving.

Robert Barton, who never leaves me rest on my laurels, and is always near, suggesting new trails, new challenges and problems to solve.

Table of Contents

Résumé en Français

Il est souvent dit que le problème de localisation à l'extérieur a été résolu. Certes, il reste beaucoup à faire pour porter les techniques de localisations dans des territoires qui ne bénéficient pas de l'infrastructure dense des pays avancés. Dans ces derniers, beaucoup reste à faire pour éliminer les poches où la couverture radio (GPS et cellulaire) est marginale, et pour accroitre la fidélité et la précision des mesures dans les zones bénéficiant d'une bonne couverture. Cependant, partout où un signal GPS et cellulaire est disponible, l'état de l'art permet une localisation effective, avec une précision de l'ordre de quelques mètres ou moins.

A l'intérieur cependant, la donne est bien différente. D'abord, parce que les besoins de localisation sont apparus bien plus tard qu'à l'extérieur : si les premiers satellites GPS ont été lancés dans les années 1970, l'idée de localiser un objet à l'intérieur, en utilisant des ondes radios, n'a commencée à être effective que dans les années 2000. Ensuite, parce que ce besoin s'est manifesté de façon différente à l'intérieur. Dehors, le besoin central s'est rapidement concentré sur l'idée du 'blue dot', un objet radio représentant sa propre position sur un écran local. A l'intérieur, les premiers besoins ont été concentrés sur les verticaux industriels et médicaux, où le besoin dominant n'était pas un 'blue-dot', mais la capacité pour un administrateur de représenter sur une carte du bâtiment la position d'objets ou de personnes (porteurs d'un appareil émettant des signaux radios).

Les techniques radios requises pour répondre à l'un ou l'autre besoin sont pourtant essentiellement les mêmes. Une approche traditionnelle consiste à utiliser la puissance du signal reçu d'un transmetteur, et une estimation des caractéristiques du transmetteur, combinée à une approximation de l'environnement où le signal a été transmis, pour convertir une échelle de signal en une échelle de distance. Naturellement, le fait que la plupart des paramètres doivent être estimés est un obstacle majeur à l'efficacité de cette technique. Une autre approche consiste à utiliser plusieurs antennes, et estimer l'angle d'arrivée du signal. En combinant les angles de plusieurs sources dont la localisation est connue, le récepteur peut estimer sa propre position. La précision de la mesure des angles est dans ce cas le facteur déterminant la précision du résultat. Une autre approche, intéressante pour notre étude, est de mesurer directement la distance du récepteur à l'émetteur. La vitesse des signaux radio étant connue, un échange de points de référence temporels permet de mesurer la durée de l'échange et la distance entre les participants. Une fois que les distances ou les angles ont été estimés, une simple résolution matricielle, souvent combinée à un filtre (comme le Kalman filter) permet de convertir distances ou angles en positions. Ces techniques sont communément déployées à l'extérieur. GPS en particulier

utilise un réseau de satellites qui transmettent en continu leur position et un horodatage. Le récepteur convertit ces valeurs en pseudo-distances, puis en déduit la position du récepteur. Le système peut desservir un nombre infini de récepteurs, puisque chacun d'entre eux est entièrement passif. Le corollaire, bien sûr, est que l'infrastructure ne reçoit aucun signal, ce qui est un avantage pour assurer le respect de la vie privée du propriétaire du récepteur, mais aussi un inconvénient pour les cas où l'infrastructure a besoin de calculer la position de clients radios. Dans le cas des réseaux cellulaires par exemple, l'opérateur utilise cette information pour basculer un client vers l'une ou l'autre borne. De ce fait, les techniques GSM et LTE ont implémenté des techniques différentes, où l'infrastructure transmet des signaux au client cellulaire, qui retourne ensuite une réponse permettant cette localisation par et pour l'infrastructure. Le respect de la vie privée n'est plus l'élément central ici : c'est l'efficacité de la couverture qui est le critère principal. Cependant, ces dernières années, le respect de la vie privée est souvent apparu comme un élément crucial, en particulier aux États Unis, après plusieurs scandales ayant révélé que les opérateurs cellulaires vendaient la position de leurs clients à qui était prêt à l'acheter. Le client peut naturellement aussi utiliser des techniques autonomes, basées par exemple sur la puissance du signal des bornes, ou l'angle d'arrivée de leurs messages, et la consultation d'une base de données contenant la position de ces bornes pour déterminer sa propre position. Cependant, ces techniques ne sont pas efficaces partout, puisqu'elles dépendent de l'existence de bases de données fiables et la réception de signaux en quantité et qualité suffisantes pour opérer une triangulation efficace.

Il serait tentant d'utiliser ces techniques directement à l'intérieur. Malheureusement, les fréquences où elles opèrent ne traversent pas bien les murs, et l'efficacité du signal tant GPS que cellulaire tend à se réduire à mesure que l'opérateur s'enfonce dans les profondeurs du bâtiment. Si les techniques qui ont été développées pour la localisation en extérieur peuvent en principe être utilisées dans le cas de la localisation en intérieur, il faut trouver d'autre sources de signal.

Cette nécessité, combinée au fait que le besoin de résoudre le problème de la localisation à l'intérieur est apparu plus tard que son équivalent extérieur, a aussi permis aux différents acteurs de mieux qualifier les besoins et les modus operandi. Dans certains cas en effet, le besoin intérieur est équivalent au besoin dominant à l'extérieur : un 'blue dot', par exemple pour un consommateur armé d'un téléphone portable et cherchant son chemin dans les méandres d'un centre commercial. Dans ce cas, les critères d'efficacité incluent bien sûr la capacité du système à fournir une position précise et en temps réel, mais aussi le respect de la vie privée dudit consommateur. Dans d'autres cas, le besoin est inverse. Un objet, comme par exemple un badge attaché à un fauteuil roulant d'hôpital, doit être localisé par un opérateur central. L'objet n'est pas supposé avoir besoin de sa propre position. Il ne s'agit pas non plus d'un objet personnel, et le concept de vie privée ne s'applique pas directement. Mais dans des cas intermédiaires, un acteur humain transporte un objet émettant des signaux radios, et l'infrastructure peut tenter

de localiser cet objet. Un cas typique peut être un grand magasin analysant les volumes et trajets de ses visiteurs. Dans ce cas, il est crucial de déterminer si chaque visiteur doit être identifié individuellement ou non, et comment, dans le cas positif, le consentement de ce visiteur est recueilli.

Ces critères sont déterminants pour décider des techniques à retenir pour résoudre les cas de localisation à l'intérieur. La technique idéale pourrait répondre à tous les différents besoins. Il serait possible aussi de déployer une technique différente pour chaque scénario. Dans ce dernier cas, les chances d'adoption dépendent largement du rapport entre le coût de déploiement et le bénéfice attendu. Pour cette raison, les technologies qui nécessitent le déploiement d'un overlay, et le développement d'un écosystème entier, tant du côté du client radio que de l'infrastructure, ont eu du mal à s'imposer. Pour cette raison aussi, une technique basée sur Wi-Fi est très attractive, parce que cette technologie est omniprésente à l'intérieur. Et de fait, dans la décade 2010, le groupe 802.11 à l'IEEE a proposé une technique appelée Fine Timing Measurement (FTM). Son design était ambitieux, et se proposait de résoudre « le problème de la localisation en intérieur » en général. L'objet de notre étude était donc de comparer cette technique aux autres, et d'examiner en particulier comment elle pourrait s'intégrer dans une stratégie globale de localisation en intérieur qui respecterait les critères examinés plus haut.

En particulier, nous montrons dans cette étude que FTM souffrit dans sa genèse de limitations de design. Pensée principalement du point de vue de l'utilisateur (d'un téléphone portable par exemple), la technique ne fut pas conçue pour apporter un bénéfice à l'infrastructure qui l'aurait déployée. Le coût de cet oubli se transcrivit en une absence d'adoption par les vendeurs de points d'accès Wi-Fi. Nous avons dont proposé une augmentation du protocole qui permettrait, sous contrôle de l'utilisateur, de fournir une indication de position à l'infrastructure.

Nous montrons aussi que le protocole souffre du problème qu'il prétendait résoudre. En l'absence de signal GPS, un terminal client obtiendrait sa position en mesurant sa distance à plusieurs points radio Wi-Fi, qui partageraient aussi leur position GPS. Mais un point radio n'est pas en capacité de déterminer sa position GPS, puisque le signal GPS n'est pas disponible. Nous montrons cependant que les points radios, agissant tantôt comme des clients FTM, et tantôt comme des répondeurs FTM, peuvent construire une carte de leur distances relatives (les uns aux autres). Les approches requises pour convertir une matrice de distances en positions sont multiples, et utilisent souvent les techniques de Multi-Dimensional Scaling. Cependant, dans le cas de FTM, ces techniques se montrent inefficaces. Nous montrons que la source du problème est que les mesures d'un point radio à un autre n'ont pas une erreur uniforme. Or, la technique de résolution des matrices de distances tend à projeter l'erreur d'une mesure sur les autres. Quand les erreurs sont équivalentes d'une paire à une autre, cette projection est sans effet. Mais quand certaines paires souffrent d'une large erreur (ce qui est le cas pour FTM, parce qu'il y a peu de points d'accès, et parce que certains d'entre eux sont dans des configurations difficiles,

par exemple de part et d'autre d'un obstacle, qui fait que le signal principal est toujours un signal reflété et jamais un signal direct), ces larges erreurs se projettent sur les mesures obtenues entre les autres paires. Le résultat est une matrice de positions inutilisable, où les distances mesurées sont parfois juste, mais les positions calculées toujours fausses. Pour résoudre cette difficulté, nous présentons une technique en deux étapes. La première vise à repérer ces paires entachées d'une large erreur de mesure. Par un processus géométrique et itératif, ces distances sont réévaluées jusqu'à ce qu'elles redeviennent compatibles avec les mesures obtenues avec les autres points d'accès. Dans un second temps, les matrices de toutes les mesures de distances ainsi obtenues, entre tous les sous-groupes possibles de points d'accès, sont comparées entre elles. Toutes celles qui sont cohérentes les unes avec les autres sont groupées, et servent de base pour établir les positions de références à partir desquelles les positions des autres points d'accès sont déduites. Il ne reste qu'à injecter une position GPS (par exemple depuis un téléphone à l'extérieur du bâtiment) pour réorienter le graphe ainsi formé, et obtenir la position GPS de tous les points d'accès de l'étage.

Cette technique permet sans doute de réduire la difficulté de déploiement pour les opérateurs d'infrastructures. En revanche, elle ne résout pas toutes les limitations de FTM. En particulier, nous montrons aussi que l'échange FTM n'est pas protégé. La quantité gigantesque de paramètres possibles, et le traitement de la priorité de la mesure GPS dans les clients font (comme nous le montrons) qu'il est facile d'obtenir l'empreinte unique du client, en se basant exclusivement sur ses échanges FTM. Il s'agit là d'un problème majeur pour une technologie qui entend protéger la vie privée du client. Mais nous montrons aussi qu'il est possible, en limitant les paramètres choisis à un petit sous ensemble qui correspond aux cas d'utilisation typiques, et en uniformisant l'implémentation de FTM dans les clients, de rendre cette empreinte bien plus difficile à établir.

Un autre frein à l'adoption de FTM est que l'échange n'est pas sécurisé. Même en utilisant des paramètres identiques, même en supprimant l'empreinte unique de chaque client, la technique expose la sécurité du client. D'abord, parce qu'il est possible de suivre la position d'un client simplement en écoutant ses échanges FTMs. Ensuite, parce que nous montrons qu'il est possible d'insérer un point d'accès dans le dispositif (agissant comme un point d'accès supplémentaire, ou prétendant être l'un des points d'accès valides de l'infrastructure), et, en manipulant les réponses, de conduire un client à une destination au choix de l'attaquant. Cette possibilité rend bien sûr la technologie dangereuse dans bien des cas. Mais nous montrons aussi qu'il est possible de sécuriser partiellement ces échanges. FTM est par définition destiné à être utilisé en des endroits qui ne sont pas familiers pour l'utilisateur, et l'on ne peut donc pas s'attendre à ce que l'utilisateur ait des clefs permettant à son terminal d'établir des liaisons authentifiées et sécurisées avec chaque point d'accès. Mais nous montrons qu'en étendant d'autres techniques 802.11, il est possible de rendre difficile l'insertion d'un point d'accès pirate, en forçant les points d'accès participants à prouver qu'ils peuvent établir une liaison sécurisée entre eux, et donc qu'ils

font partie de la même infrastructure.

Ces éléments ne résolvent sans doute pas toutes les limitations de FTM, et nos futurs travaux exploreront notamment les scenarios hybrides, où une haute densité de clients porte à préférer un mode FTM passif, mais où l'infrastructure souhaite aussi obtenir des statistiques sur le flot des visiteurs. En revanche, nous espérons que les contributions de ce mémoire rendent déjà la technologie plus attractive en tant que candidat contribuant à résoudre le problème de la localisation à l'intérieur des bâtiments.

INTRODUCTION

This thesis probably resembles a journey that many professionals in the field of indoor location have undergone. In my particular case, the journey started in the early 2000s. At that time, Wi-Fi was new, and I was consulting and teaching infrastructure customers about a Wireless solution based on 802.11. Part of the portfolio included a Location Based Services engine. At that time, client devices were only laptops that could be disconnected from the power for 2 hours at most before the battery ran out. But enterprise customers were enjoying this newfound nomadic freedom, moving onto the floor from one working spot to another. In this context, it was becoming critical for IT to know where the laptops were, as most of them were Enterprise assets, and as users would frequently report drops in their connections.

Location back then was relying on the fact that laptops sent as often as possible broadcast probe requests at low data rate, in a feverish attempt to confirm the availability of these fleeting Wi-Fi access points (APs). Those requests could be heard loud and far, and APs would use a simple signal-to-distance conversion, then trilateration mechanism to provide an idea of where the client device could be. Accuracy was poor (10 to 15 meters), but IT customers were satisfied with the philosophy "if I am that close, I'll likely see it". The idea of providing location to the user of the end device was completely foreign.

The apparition of consumer GPS and smartphones changed the landscape. It was not a violent change, but rather a shift in expectations and behaviors. Users started to expect to see their location on their device. Although they understood that such technology was only available outside, a growing frustration started to crystallise, that public indoor venues were not doing much to port to the inside a blue dot that most people had come accustomed to outside. At the same time, the requirements for smartphone battery efficiency pushed vendors to send less and less of these 802.11 probe messages. Knowing that asset tracking had turned into a customer data collection system in public venues, end-device manufacturers also started attempts to hide the device identity and activity. By mid 2010, asset tracking had become a crippled solution refusing to die. In most settings, individual asset tracking had become sparse at best. In public venues, statistics about foot traffic based on Wi-Fi tracking had merely become a small and partial representation, to be used with caution.

By that time, I was traveling through Los Angeles airport. Like many travelers, I was using an airline app for my virtual boarding pass. As I was waiting for a connection flight, my phone vibrated to a notification from the app, reading: "your flight will be boarding in 30 minutes. You are 20 minutes away from the gate, you should leave soon". I was indeed about that distance

away from the gate, and surprised that my phone could determine my location with some level of accuracy, while I was inside the airport building (I thought that GPS did not work inside?) At the same time, the building had large windows, so I was not sure if my location was relying on an indoor technique (maybe the asset tracking system from the airport Wi-Fi infrastructure is tracking my phone and is sending the location back to the app?) or on GPS.

On my return trip, I transited this time through an airport in New York. As the building structure was thick concrete with no window in sight, I decided to check. The app displayed that I was near Gate 8. But in fact, I was near Gate 16, which was in a different wing of the airport complex. There was no way that GPS would provide location at such a poor accuracy level (and still provide a usable reading). What technology could provide location with such a mix of accuracy and inaccuracy? I later learned that it was based on 802.11, but using a crowd-sourcing technique to learn the position of individual access points. The technique is notoriously challenged when APs are moved or updated [1], which was the case for that airport.

Like many users of the technology, this appalling result left me frustrated. But as a wireless professional who had been teaching 802.11 localization for 15 years [2], it dawned on me that I was among the ones who should do something about it. At that time, the 2016 revision of the IEEE 802.11 Standard was in progress, and I was reviewing the many proposed additions to the text. Many of such proposals are often technical or editorial fixes, many others are good ideas that end-up never being implemented. Only a few belong to the "useful and popular" group. Among all the proposals, a ranging technique called Fine Timing Measurement offered to bring the blue dot to client devices based on 802.11 exchanges [3].

At the same period, many Enterprise customers were reporting that they concluded that the state of the art asset indoor location tracking was no longer providing acceptable results. My focus then became to study alternative options. Could GPS or cellular techniques be extended indoor? Other indoor technologies, like ultrasound, light or BLE were emerging, all promising solutions for indoor location. Could they replace the failing 802.11 techniques?

All options were open, but it seemed that replacing an 802.11 technology based on obsolete assumptions, with another 802.11 technology, but built on modern premises, was a logical direction. However, FTM in 802.11-2016 seemed to only represent a few pages, much of which seemed more theoretical than built on strong experimental foundations. The Wi-Fi Alliance had attempted to promote the technology with a Location R1 certification, but the adoption numbers were abysmal. 16 Wi-Fi systems were certified, mostly test boards. For comparison, more than 4000 devices were certified for the technology based on 802.11n. This seemed to indicate that no one really believed in the FTM technology, or that everyone was waiting for a large vendor to make the first move.

Therefore, I focused on FTM in more details, to research its ability to fulfill the requirements of a modern indoor localization technology, find out if it could really provide an accurate blue dot

on a client device, and if it could also replace the obsolete 802.11 techniques for asset tracking and foot traffic analytic.

This thesis reflects this study and its conclusions. The very notion of a "modern localization technology" is highly subjective, because it includes technical components, but also societal components (that reflect what users of each technology expect from it). Therefore, the first contribution of this thesis (chapter 1) is a review of the localization technologies based on wireless signals. Localization technologies are varied, but are all limited by the same laws of physics. Therefore, the general guiding principles that convert a set of signals into a location value are useful elements that are reused by most techniques.

The second contribution (chapter 2) is a review of the outdoor localization technologies. We look in detail into GPS and cellular localization techniques, first to understand their technical approach, not only to evaluate how far indoor they can be extended, but also to evaluate if an indoor complementary technology could use the same principles. We then look at the use case they fulfill, to better understand the user expectations of what these techniques should and should not achieve.

The next contribution (chapter 3) is then to look at the localization technologies that were proposed primarily with the indoor environment in mind. These techniques are many and very different from each other. Here again, their technical approach and the use cases they solve are invaluable to understand what a 'modern' localization technology is expected to achieve.

With this material in mind, we then study FTM in detail (chapter 4). The comparison with the other techniques allows us to see multiple benefits in the design choice, but also multiple challenges that were left unsolved. There had been very few studies on FTM, and it seems that the findings we made [4] [5] [6] contributed to push the industry to have a more acute look on the possibilities and risks of the FTM technology.

Among the challenges, FTM assumes that the AP geo-location is configured. However, getting this location information is very difficult indoor, and our next contribution (chapter 5) [7] [8] [9] is to design a machine-learning method to have APs automatically learn their geo-location through FTM exchanges with one another, and with external stations running a GPS client. This technique solves a fundamental obstacle to an infrastructure adoption of the FTM technique, allowing network owners to simply deploy APs, then let them learn their location by themselves.

Facilitating the support of FTM on the infrastructure is not sufficient to call the technology 'modern'. Another key requirement is to respect the user privacy. But our next contribution (chapter 6) shows that a neural network can be built to fingerprint devices solely based on their FTM pattern [10]. This work shows that, left unchanged, FTM would present dramatic risks for user privacy. In the same contribution, we design a method to limit these risks.

Our last contribution (chapter 7) is to look at FTM exposure to attacks that plagued other localization technologies: ranging and location attacks. We show that such attack is also very

possible with the default structure of FTM [11], allowing an attacker to lead an unsuspecting victim to the destination of the attacker's choosing. We also propose enhancements to the 802.11 Standard to mitigate the attack exposure.

We then examine if FTM, after these changes, could be the indoor localization technique of choice that the industry has been looking for.

Chapter 1

How to Localize an Object Using Wireless Signals

1.1 Introduction

There are multiple localization technologies based on wireless signals, but they all have to abide by the same laws of physics. Therefore, and although there are broad variations in the approaches from one technology to the other, it is interesting to first lay the groundwork and examine how wireless signals can be used to express location. We will see that localization techniques can attempt to establish ranges using signal strength, time or angles to find the relative position of two wireless objects. We will also see that, when one of the objects moves, an attractive augmentation technique is to compare each new evaluation to a predicted value, in an attempt to limit the effect of measurement noise.

1.2 Classification of Localization Techniques

This study is about indoor localization, which is an attempt to find location, that is answer the question: "where is this particular device"? There are many types of localization techniques, some working well outside and partially extending inside, some working exclusively inside, some providing centimeter-level accuracy while others achieve 15 meters at best. For our purpose, organising the field and placing FTM in a relevant group could be useful. However, there are countless ways of classifying localization techniques. In many ways, this is because the answer is a question of perspective: "where", is only meaningful in relation to some point of reference. Localizing (the act of localization) an object is first about finding the position of an object relative to the position of known reference points. Then, when the object position is translated into the coordinate system used with these reference points, position becomes location, changing from a "relative to" metric to a set of coordinates that do not need the known object anymore to be understandable.

This translation is only partially applied indoor. When a nurse searches for a blood pump in a hospital, the answer "in room 24" is technically a position, but it can serve as a location value if "room 24" is considered as a self-standing frame of reference (which means that the location

of that room 24 is well-known and does not need to be expressed into a larger or more general reference). Some localization techniques generate civil coordinates ("1725 King Street"), others will generate geographical coordinates (35.18380 North, -79.11860 West), others will generate coordinates that are relative to known landmarks or feature of a particular building ("in room 24"), and all could be labeled "locations" if the used frame of reference is known by the user of the information.

Therefore, a good way to organize the various types of location is to look at who uses the information, which questions where location is computed and displayed. The literature on sensors and location using wireless technologies (for example in [12] or [13]) commonly distinguishes the centralized and the distributed localization families of techniques.

1.2.1 Centralized Localization Techniques

In the centralized approach, a radio infrastructure collects information and signal from a mobile object. The infrastructure, in this context, may be one or more radio points, that we will generically call *anchors* until a technique-specific term is needed. The collected elements are then centralized into a location engine that computes the mobile object location [14] [15]. This location can then be made available to an operator having a direct relationship with the location server (*e.g.,* IT team managing the radio network, planning team in a cellular network operator, etc.), or can be sent to the mobile device and be displayed locally.

The purpose, and the place and time at which the location information will be consumed are very important for this type of techniques:

— In some cases, as will be detailed in chapter 3, the information is primarily intended to be returned back as fast as possible to the mobile device [16], for example to help with a navigation task [17]. In that case, the time it takes for the signals to be collected, concatenated in the location engine, the location to be computed and then returned to the mobile device may be a key factor. Because in most cases the device needs location information in near real time, the collection and concatenation delays, the localization algorithm and the distance between the location server and the access network (radio points and mobile device) are constrained by the maximum acceptable distance between the device current position and the displayed position on the screen (which in turns depends on the mobile movement speed).

— In some cases, the information is primarily intended to be consumed by a central system, but the delay between the time when the mobile device appears at a specific location and the time at which this location is computed is still of critical importance. For example, the location system may be set to trigger an alarm when the device enters a hazardous area [18]. In that case, short vs. long delay may mean the difference between life and death. In other verticals, the location system may be set to trigger an alarm if an asset moves

away from some reference point (or goes through a door). Here again, short vs. long delay may mean the difference between succeeding or failing to prevent theft. The system will therefore be constrained as well by the delay variable, limiting as above the acceptable collection and computation techniques.

— In some cases, a single mobile location is still needed, delay may be somewhat important, but is no longer a critical factor. A typical example is network management, where individual device location is needed to provide proper IT support and understand why an issue occurred (*e.g.,* connection issues while on a teleconference) based on the device location and other factors [19]. In that case, the continuous location of the device is important (so the movement can be followed, for example up to a coverage gap area), along with individual device identification. This requirement has consequences on the design of a "good" localization protocol, as will be detailed in chapter 4. The collection and computation delays are less important than in the previous cases, because troubleshooting either occurs after the facts (giving ample time for location to be computed), or IT support is in direct communication with the user (who can verbally update the location information).

— In some cases, individual devices locations are not important, and the system looks at trends (*e.g.,* foot traffic analysis in a public venue [20] [21]). In that case, the collection, travel and computation delays are no longer a constraint, thus opening the door to heavier localization computation algorithms that present additional properties of interest for the particular use-case at hand, as will be seen later in this chapter. Individual device unique identification is often no longer necessary for this case. The system also often accepts only partial location estimates (*e.g.,* a device is located for a short duration, for example 10 to 15 seconds, then is no longer detected for some time), and can still surface useful patterns at the scale of hours or days.

— In some cases, the mobile devices are not really mobile, but nomadic. Nomadic devices are used in one location, and can be moved to another location. However, the displacement from one location to another is mostly not relevant to the use case where they are inscribed. By contrast, mobile devices move, and their location during the movement is important. A typical example of a mobile device is a smartphone or a tablet used in the third bullet above. When the user complains about poor connection experience, the position of the device during the movement is the critical piece of information needed by IT support. A typical example of a nomadic device is the blood pump in our "room 24" example. What matters is where the pump is used now. The details of its past movement from room 31 to room 24 are in most cases secondary considerations. The computation of the position of nomadic devices also relaxes the collection and algorithmic constraints, because a large set of samples can be taken before location computation is attempted. This relaxation opens the door to additional techniques, such as large matrices with Euclidean Distance Matrix

(EDM) resolutions, which we will detail and use in our contribution in chapter 5.

Therefore, the fact that location is computed on a central system does not mean that there is a primary technique that should be assumed. However, such central system commonly has two characteristics:

1. Power is not a problem, as such location server is typically connected to a permanent power source. As such, power-hungry algorithms are perfectly acceptable.
2. The computing resources can also be tailored to the use case. Based on the number of devices to locate, and based on the location requirements detailed above the the accuracy needs, the computing resources can be increased or reduced.

1.2.2 Distributed Localization Techniques

In the distributed approach, the location is computed by individual participating nodes. In the case a mobile device location, an example would be the device collecting signals from anchors, and using that information to determine its own position then location. In this case, the locations of the anchors are known (either shared in a radio signal, or pre-configured on the mobile device). The scenarios can be more complex. In sensor networks, multiple nodes can be positioned in a field, and each node computes its relative position to the other. Most of the nodes have no information about their own or their neighbors' positions. Multiple techniques can then be used to build a map of the relative positions, like sampling and fingerprinting [22] or the search of an error minimum [23] [24]. These techniques will be explored further in our contribution in chapter 5.

In all cases, the system is self-contained, in the sense that only positions are determined, until a system of reference coordinates is inserted. This addition is usually achieved by inputting one or more reference locations (*e.g.,* one or more contributing node has a declared known location, or can compute its location through an external method, for example GPS). The known locations are then used as seed to orient the graph of relative positions and inscribe it into the larger coordinate reference framework.

1.2.3 Hybrid Localization Techniques

The above containers found in the literature focus on the place where computation happens, but ignore the structure of the radio signal exchanges, which is nevertheless a key dimension to a proper technique classification. In the centralized approach, the mobile could be the only source of radio signals, and the infrastructure could be entirely passive. In most approaches, infrastructure anchors still exchange messages over the wireless medium, but these exchanges may not have a direct usage for the location determination (see analysis in section 1.4.1 and example in section 3.3.3).

In reverse, in the distributed model focusing on a single device computing its location, the mobile may be passive, and only the anchors could be sending signals. This approach has useful properties (privacy, unlimited mobile device count, *etc.*) that will be examined in chapter 4.

Thus a classification could also be designed where the direction and the quantity of signals is a key classifier. In this model, the passive side does not consume any airtime (that side can thus scale to infinity), and does not consume energy for transmissions (limiting the radio energy expenditure to receiving signals when measurement collection is started). Because the passive side is not transmitting, it is also not detected by the other side. This design decision has a lot of consequences that will be detailed throughout this thesis. One of them is that the passive side may compute its location, but the active side then does not obtain the passive side location value (and is not even aware of the passive side presence), unless an additional system is put in place to send this information. The method can be a subsequent message or an out-of-band technique (*e.g.,* another radio technology).

With this approach in mind, it is therefore useful to also list hybrid technologies, where each side contributes to the location determination [25]. This contribution can take the form of signals, either where both sides' input is needed for one side to compute position or location, or because the technique is designed to make sure that both sides can compute the position or location output. As we navigate through the different techniques, the advantages and limitations of these hybrid approaches will become apparent.

1.2.4 Range or Range-Free

The literature on sensor positioning also makes a clear distinction between range-based methods and range-free methods [12]. This terminology may be misleading. In this context, "range" refers to a form of measurement, either distances (in which case "range" is indeed a logical term choice), but also possibly angles (and sometimes both distances and angles). The combination of several of such measurements is then used to determine a position or a location value. These range-based techniques will be examined in turn below, as they are the techniques of interest for this thesis.

However, such measurement and the related location computation consume radio and CPU resources that constrained sensors may not have (or cannot expense liberally). For example, measuring distances may require measuring time of flight, or signal strength, or exchanging multiple timestamped messages *etc.*, all operations requiring specific programming and access to specific hardware elements (accurate clock, received signal strength estimator, *etc.*) Therefore, in the sensor world, an additional set of techniques, called "range-free", are widely studied. In this setting, a set of fixed anchors are deployed, and their number may be large. The task is then to compute the location of a particular sensor present in this set of anchors. As the goal is to minimize energy and hardware expenditures, this exercise is often limited to estimating the

relative position of the sensor (*e.g.,* "within the triangle formed by anchors 1, 32 and 17").

These techniques are undoubtedly valuable and [26] provides a good introduction to the main families. In most cases, the algorithm attempts to draw geometric figures representing the relative positions of detected anchors (*e.g.,* triangles formed by each set of 3 detected anchors, with the Approximate Point in Triangle, or APIT, technique) then find the most logical position of the sensor based on these figures (triangle intersection, centroid determination, hop count and lines intersections, *etc.*)

One key limitation of range-free techniques is that the expenditure savings achieved by this approach comes at the price of accuracy challenges. Measurements do not occur, so the accuracy is bound by the density and relative positions of the anchors. Additionally, a smartphone in a public venue may have power, radio and CPU constraints, but they are not so dire that measuring distances or angles, and computing a matrix error reduction from these elements would be an impossible task. Therefore, the scenarios envisioned in this thesis support an end-device resourceful enough to measure and compute elements, and range-free techniques are not a primary consideration.

1.3 Distance Estimation Techniques

There are many "range-based" techniques, as the Sensor networking literature would classify them. An intuitive approach to finding a mobile device location is to measure the distance from that device to one or more anchors used as reference points and which locations are known, combine these estimated distances to deduce the most likely position of the device relative to these anchors, then translate the position into a set of reference coordinates. Although many algorithm perform all three operations, they are distinct steps in the localization process. The first requirement is therefore to estimate a distance (in the context of this thesis) using radio signals.

1.3.1 Estimating Distance from Signal Strength

Multiple protocols use the notion of received signal strength (commonly called Received Signal Strength Indicator, RSSI), for purposes unrelated to location. This value is a useful metric for mobile devices that exchange data with other nodes (fixed or not). By estimating the strength of the signal received downstream from another node, the mobile device (assuming link symmetry) can estimate which upstream modulation technique might be the best compromise between maximizing transmission efficiency (which leans toward choosing a complex modulation and encoding technique, that will maximize the information density by unit of time and therefore minimize the transmission time and its associated energy and airtime consumption) and maximizing resiliency (which leans toward choosing a simple modulation and encoding technique,

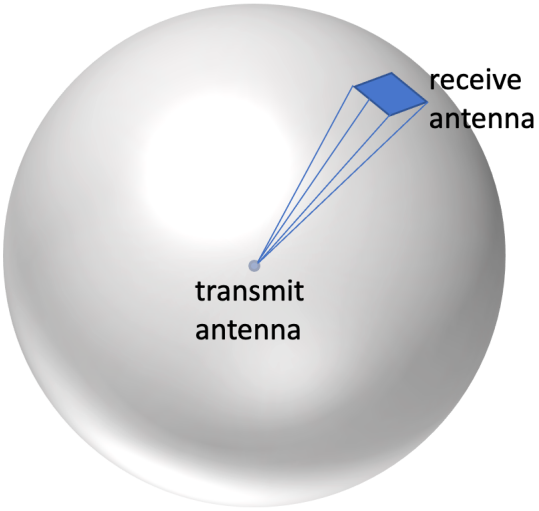


Figure 1.1 – Signal propagation in free space

that will maximize the chances that the first transmission will be successful in a given noise and distance setting).

Therefore, the RSSI is often available, and using that metric to convert the signal value into an estimated distance is an attractive approach [27] [28]. Converting RSSI to distance is possible [29] because, in free space, the signal transmitted by a theoretical omni-directional antenna expands in a sphere centered on the emitting antenna, as shown in Figure 1.1. The surface area of the sphere, at a distance d from the antenna, is $4\pi d^2$. As the distance increases, the resulting sphere is larger, and the same amount of signal is spread along a larger surface (and is thus weaker at any point of the surface of the sphere). Therefore, the amount of signal loss (the free space path loss, $FSPL$), as the signal expands, is a direct factor of the size of the sphere. It also depends on the wavelength λ of the signal. The reasons for this additional relationship include multiple physical parameters, but it is also intuitively logical that a signal that undergoes multiple cycles by unit of distance (high frequency) would incur a higher loss than a signal that undergoes only a few cycles (low frequency) in that same distance. The path loss can then be expressed as follows:

$$FPSL = \left(\frac{4\pi d}{\lambda}\right)^2 = \left(\frac{4\pi df}{c}\right)^2 \tag{1.1}$$

where f is the frequency of the signal and c the speed of light. Converting equation 1.1 to decibels, and thus to a logarithmic scale, provides the following equivalence:

$$FPSL(dB) = 10log_{10}\left(\left(\frac{4\pi df}{c}\right)^2\right) = 20log_{10}\left(\frac{4\pi df}{c}\right) \tag{1.2}$$

From this equation, and if the infrastructure node sends its signal at a known power p_T, the mobile device should be able to measure the power of the received signal (the RSSI), deduce the path loss, and revert equation 1.2 to find the likely distance to the node. This reversion relies on the principle that the power density w received at any point of the sphere is also a direct factor of the distance, and thus can be expressed as $w = p_T/4\pi d^2$.

However, in real systems, the infrastructure node's antenna may not be omni-directional, and may therefore focus the signal in a particular direction. This focus translates into an additional amount of energy (the transmit gain, G_T) toward the direction of the main lobe of the signal beam (energy that is removed from the directions where the signal does not travel). Similarly, the receiver also has an antenna that receives the signal on a surface larger than a point on the sphere, and thus receives an amount of energy proportional to the receive antenna surface A. This ability related to the antenna surface is often expressed as the receive gain, G_R, which maps to the antenna area with the relationship: $A = \frac{G_R \lambda^2}{4\pi}$. The received energy also depend on the wavelength of the signal (λ), for the same reasons as above. Therefore, the power density estimation can be transformed into a measure of the available power p_R at the receiver antenna, as follows:

$$p_R = \frac{p_T G_T}{4\pi d^2} A \tag{1.3}$$

The product $p_T G_T$ is called the effectively radiated power (ERP) of the transmitter. It will come as an important quantity for FTM. Therefore, by measuring the power received at the antenna (expressed as the RSSI, *i.e.* commonly expressed as an integer in the $(1, 255)$), the mobile can revert equation 1.3 to estimate its distance to the transmitting node:

$$d = \sqrt{\frac{p_T G_T}{4\pi p_R} A} \tag{1.4}$$

The technique is simple. As the receiver moves away from the sender in a straight line, the signal degrades. When the RSSI is expressed in dBm, the degradation forms a negative log curve, as represented in Figure 1.2. One characteristic of this degradation is that it is stiff close to the emitter, then becomes less pronounced as distance increases. The effect is that it is rather easy to tell the difference between, for example, 2 and 4 meter distance, but it is difficult to tell the difference between 60 and 70 meter distance, unless conditions are ideal (lab measurements and low noise).

However, this model makes multiple assumptions that do not apply well in the real world. One of them is that the power of the transmitting node is known. Some protocols mention the transmit power in the transmitted frame, but many do not. The receiver may then have to estimate (or guess) the transmit power. Other techniques use a fingerprinting approach, where measurements of FPSL are taken at one or more known distances from the node and used to

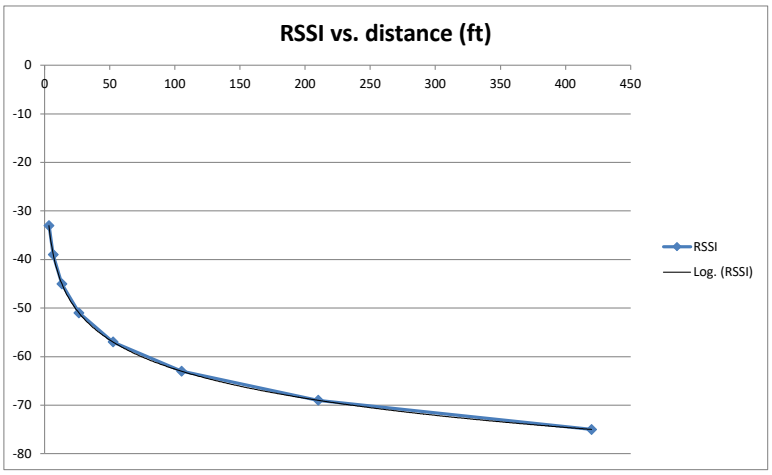


Figure 1.2 – Example RSSI over distance (dBm scale)

estimate the correct distance when other signal values are measured.

The technique also assumes that the antenna of the transmitter and its gain in the direction of the device are known. These are also often impractical assumptions. Figure 1.3 shows the radiation pattern of a bi-directional antenna used in 802.11 access points. The antenna is supposed to be at the center of the graph. The green continuous line represents the signal level at which the signal emitted from the antenna would be received at a given position. The graph provides multiple levels of information. As can be seen, a first level is showing that the antenna has a strong gain toward the right of the graph (270 degree direction) and toward the back of the graph (roughly toward the 70 degree direction). By contrast, the signal at directions 350 degrees or 150 degrees is much weaker. The second level is that the green line also represents the points of iso-signal. In other words, a device turning around the antenna, while staying on the green line would get the same signal strength at any point of the path (the exact value depends on the transmitter power). The path would of course bring the device much closer to the antenna at some positions than at some others.

The concentric scale (concentric circles) also represents the differences of gain (in dB) at various angles. Formally, the scale compares this antenna to a perfectly isotropic antenna (a theoretical antenna that would radiate signal in a perfect sphere, *i.e.* equally in all directions). By looking at the position of the green line for, say, direction 270 degrees (the line is at about 4 dB in the concentric axis, which means that this antenna radiates 4dB more in this direction than an isotropic antenna would) and 150 degrees (the line is at about -24 dB in the concentric axis), one can also infer that a device performing a perfectly circular rotation around the antenna will get a $(|24 + 4|)$ 28 dB stronger signal at position 270 than in position 150. However, in real settings, the orientation of the receiver to the antenna is usually not known. The radiation pattern is also unique to each antenna. The pattern has a vertical and a horizontal component

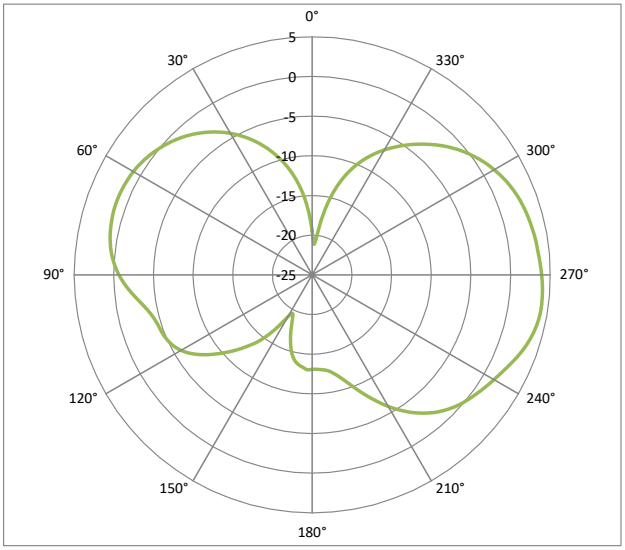


Figure 1.3 – Horizontal view of the radiation pattern of an omni-directional antenna

(we only represented one of the views in the figure, as the goal is to express the limitations of the method, not to characterize a particular antenna), and most vendors do not publish widely their antennas patterns. Here again, estimations or sampling have to compensate for the unknown. Alternatively, the measurement can be considered noisy and dealt with at the position computation phase. However, this second approach is certain to insert inaccuracies that will reduce the value of the position estimation. This technique is still useful to our purpose and for FTM, as will be seen in sections 5.5 and 7.4. However, the large quantity of unknown makes that great accuracy cannot be expected from this technique alone.

1.3.2 Estimating Distances with Time of Flight

A good way to remove the uncertainties of of RSSI-to-distance conversion techniques is to attempt to measure the travel time of the signal from one device to another [30] [31]. All electromagnetic signals travel at the same speed c.

c is commonly called "speed of light", but it is the speed of any electromagnetic transmission in the RF spectrum, light or other. This fact is well understood since Maxwell, but in the 19th century, the scientific community, led by Lorentz, then Poincarré and later Einstein, was primarily interested in the light transmission case, and the term speed of light stuck. We will use it as a general term throughout this thesis.

This speed changes slightly with the medium, and is well known for the air (299,702,547 meters per second in dry air condition at standard sea-level pressure). Therefore, if a signal leaves the antenna of a transmitter at time t_1 and reaches the antenna of a receiver at time t_2,

the distance between the transmitter and the receiver is simply set by the equation:

$$d = (t_2 - t_1)c \tag{1.5}$$

Replacing the RSSI method with an estimation derived from the signal travel time (time of Flight, ToF) is tempting. However, this technique also encounters a certain number of difficulties, that are solved in different ways by the various protocols that use it (the particular strategies will be examined below and through chapter 3). The first difficulty is to precisely measure the departure and arrival times at the antenna. This challenge will be detailed further in the case of FTM in section 4.4.1. In essence, the antenna itself is a piece of copper and does not include a clock (the clock is in an electronic circuitry inside the object), so the systems have to be able to compensate for circuit and measurement delays to evaluate t_1 and t_2.

A second and larger problem is that each side only has one of the values. The transmitter knows t_1 and the receiver knows t_2. To measure the signal ToF, both numbers must be known. This difficulty means that both timestamps have to be sent to the system that will compute the distance value. When this system is one of the radio devices, a second transmission is required to pass the local timer to the other side.

This requirement leads to a third difficulty, which is that the timestamp value is expressed in the local clock reference. Unless both clocks use the exact same time reference, one side timestamp value has no useful meaning for the other side. Some solutions attempt to synchronize the clocks. This system is efficient, but faces the additional challenge that the crystals on each side may not vibrate at the exact same rate, and therefore the clocks may start drifting from each other as soon as they are synchronized (see sections 3.3.2 and 3.3.3). This effect is unavoidable in consumer silicon, where the price of the chipset is a key consideration. Other systems (*e.g.,* satellites used for GPS) solve the problem by including a high accuracy clock (with the limitation of a high cost). But systems that use consumer-grade chipsets resort to synchronizing often to set a common reference and limit the drift problem. During the manufacturing process, the mean drift of a particular chipset is measured, and this value derives in recommendations for the synchronization interval that constrains the drift within an acceptable range. These recommendations sometimes are translated in inter-vendor interoperability specifications.

The acceptable range naturally depends on the range measurement accuracy that is sought. Light travels approximately 30 cm in a nanosecond. Therefore, any 1 nanosecond drift between clocks introduces an additional inaccuracy of about 30 cm in the distance evaluation. Maintaining synchronization between clocks may therefore require a large quantity of exchanges (thus airtime consumption). These exchanges are not useful to distance measurement by themselves, they are merely overhead requirements that ensure that measurements can be performed within the same time reference. As can be expected, more frequent measurements reduce the risk of clock differences, but also leave less time for the distance measurements themselves, causing a catch

22 (self-contradicting) scenario.

To limit this difficulty, other solutions attempt to get rid of the clock synchronization problem. This can be achieved by simply causing each side to express an interval instead of a time. A second set of messages is sufficient to achieve that goal. This phenomenon is easily understood with an example. Suppose two devices A and B. A sends a message at t_1, which is received by B at t_2. If A then communicates $t_1 = 913289187$, then B needs to align its clock to A's so that t_1 makes sense in B's frame of reference. But if B's local clock is not synchronized, and B's t_2 is in a different range (*e.g.,* $t_2 = 154954644$), then the distance cannot be evaluated (in this example, $t_2 - t_1 = -758334543$ is not only a negative number, but its absolute value would also translate in a distance of 2.10^{17} meters, which is nonsensical).

However, suppose that B then sends a message at time t_3, received at time t_4 by A. Then the distance between A and B simply becomes:

$$d = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} c \tag{1.6}$$

This technique is widely used, for example by FTM and UWB. Because $(t_4 - t_1)$ and $(t_3 - t_2)$ are now intervals, the absolute values of the clocks do not matter anymore. In the example above, A could determine that $t_4 = 913289293$ and B that $t_3 = 154954662$. Regardless of the clocks misalignment, the intervals can still be computed without depending on the local clock value, and thus $(t_4 - t_1) = 106$ and $(t_3 - t_2) = 18$. Using equation 1.6, and knowing the unit of the time intervals (*e.g.,* nanoseconds), a distance of 13.19 meters can be found. Quite naturally, this technique implies that all times are shared with the device that will compute the distance.

The absolute reference issue is solved, but the clock drift issue is still not completely eliminated. A computes t_1 and t_4 using its own clock (likewise for B and t_2 and t_3). If the crystals on both sides do not oscillate at the exact same rate, 18 or 106 nanoseconds measured on one side would not be measured to the same duration by the other side. Therefore, drift is still an issue to consider, with the awareness that the drift is constrained to the interval measured (in our example, the drift occurs at most during an interval of 106 nanoseconds). We will see in chapters 3 and 4 how individual protocols (UWB and FTM respectively) address this issue.

The above estimation relies on the speed of light in the air. However, buildings are often filled with obstacles (plaster walls, glass *etc.*) There seems to be a urban legend in the indoor location engineering community that these obstacles necessarily affect dramatically the distance measurements, because the RF signal will slow down in material denser than the air. The propagation of radiowaves is well-known [32], and the effect of building material also extensively modeled [33] and documented [34]. Radiowaves travel slower in denser materials. In a building, a plaster wall is a typical obstacle. Propagation speed depends on the signal frequency, the plaster humidity and a few other factors, but a speed of 197,000 km/s is a good approximation. Compared to a travel speed in the air of 299,703 km/s, radiowaves do travel considerably slower

in the plaster wall than in the air. These raw numbers may be where the urban legend takes its source.

However, a valid comparison is not in the raw speeds, but in the differential results between open space and a building with walls. A plaster wall is typically composed of two plaster sheets (construction details vary widely from one country to the next, but 1.5 cm thickness for each layer is a good approximation), with dampening material in between (rock wool or equivalent, which primary component is air). For this thought experiment, consider a series of walls representing together a one meter thick plaster obstacle. This quantity represents more than 30 walls. Such thickness is unlikely in real life, as a plaster wall can attenuate the signal by 10 to 20 dB (again, the attenuation depends on the thickness, humidity and other factors). Therefore, it is unlikely that a signal would actually survive crossing a meter of plaster and be usable on the other side. But we suppose it does for the thought experiment. In the air, the signal would take approximately 3.34 nanoseconds (depending on the frequency) to travel one meter. But the signal would take approximately 5.076 nanoseconds to cross the full one meter of plaster. Thus, the series of walls causes the signal to be delayed by 1.736 nanoseconds. This added delay makes that the operator, thinking that the signal was transmitted in open space, would over-evaluate the distance by 60 cm. Each wall (supposing that two-sheet structure) adds 1.8 cm to the measured distance. As will be seen in the upcoming chapters, the accuracy expected by the techniques for strict over the air transmission makes that the wall delay is negligible. For example, UWB can merely cross a single wall, and offers an accuracy expectation in the order of 10 to 20 cm. FTM signals may go through half a dozen walls (10 cm increased perceived distance), but offers and accuracy of about 1.5 meters (with an 80 MHz signal, see 4.4). Thus it is reasonable to consider that obstacles do play a role in the distance estimation, but their effect is not major at the scale of the other factors that affect the range accuracy.

1.3.3 Determining Location from Distances

At this point, we have two methods to evaluate distances. As noted above, measuring the distance to an anchor is the first step, but is insufficient to conclude on a mobile location (unless the goal is proximity detection as will be detailed in section 3.2.2). In most cases, an iterative series of measurements is conducted against multiple anchors. Then, the set of estimated distances needs to be converted into a location or a position value. Several general techniques are available for such conversion ([35] provides a complete analysis), that are so widely used by many localization technologies that they are worth mentioning here, before protocol-specific modifications are examined in subsequent chapters.

Simple 3-sphere intersection

If the distances to multiples anchors are known, one tempting method to derive a position is to draw circles around the anchors (which radius is the distance to that particular anchor), and look where the circles intersect. In fact, the mobile and the anchors are not necessarily on the same plane. For example, the mobile may be on the second floor and detect an anchor on the fourth floor. Even when both anchor and mobile are on the same floor, the mobile may be 1 meter above the floor (a phone in someone's hand) and the anchor may be attached to the ceiling. As such, the problem of resolving the position has to be thought in $\mathbb{R}^3$.

Therefore, when the mobile exchanges with a single anchor, the output is a single distance positioning the mobile anywhere on a sphere (and not a circle) around the anchor. With 2 anchors, 2 spheres are formed and their intersection is a curve lying on a plane perpendicular to the axis formed by the sphere centers. Such result is of course too inaccurate for proper location. 3 spheres offer a better solution, as they intersect on two points (if all three intersect). There remains a two-way ambiguity. Naturally, if all three anchors are on the same floor and at the same height, the two possible intersection points are at different heights, and the mobile may have internal additional sensors that can be used to resolve the ambiguity (*e.g.,* additional readings from GPS or an internal barometer to establish a possible height and discard one of the solutions). In many cases, the question of the height is also not posed by the user. In a "blue-dot" scenario, the user opens a particular floor plan, and asks the device to display its position on that floor (so the verticality is solved before the computation starts). Therefore, in many cases, using the distances to three anchors is a good approach. Determining the position from these distances is a simple transformation.

Formally, if x_i is an anchor of coordinates $x_i = (u_i, v_i, w_i)$ and $y = (u_y, v_y, w_y)$ is the mobile, the correct distance d_i from the mobile to the anchor can be written as:

$$d_i^2 = (u_y - u_i)^2 + (v_y - v_i)^2 + (w_y - w_i)^2 \tag{1.7}$$

When the mobile ranges against three anchors (i, j, k), the system becomes:

$$\begin{cases} d_i^2 = (u_y - u_i)^2 + (v_y - v_i)^2 + (w_y - w_i)^2 \\ d_j^2 = (u_y - u_j)^2 + (v_y - v_j)^2 + (w_y - w_j)^2 \\ d_k^2 = (u_y - u_k)^2 + (v_y - v_k)^2 + (w_y - w_k)^2 \end{cases} \tag{1.8}$$

Solving this system provides the mobile position. In its simplest form, an algebraic solution starts by defining a plane formed by the position of the three anchors (the spheres centers). One anchor is positioned at the origin, thus $x_i = (0, 0, 0)$. Another anchor is positioned along the v axis, thus $x_j = (u_j, 0, 0)$. The last anchor is positioned on the same plane, thus $x_k = (u_k, v_k, 0)$.

In such coordinates, equation 1.8 becomes:

$$\begin{cases} d_i^2 = & u_y^2 + v_y^2 + w_y^2 & (a_1) \\ d_j^2 = & (u_y - u_j)^2 + v_y^2 + w_y^2 & (a_2) \\ d_k^2 = & (u_y - u_k)^2 + (v_y - v_k)^2 + w_y^2 & (a_3) \end{cases} \tag{1.9}$$

By subtracting the second equation from the first $(a_2 - a_1)$, then the third from the first $(a_3 - a_1)$ and using the first equation, u_i, v_i and w_i are found as:

$$\begin{cases} u_y = \dfrac{d_i^2 - d_j^2 + u_j^2}{2u_j} \\ v_y = \dfrac{d_i^2 - d_k^2 + u_k^2 + v_k^2 - 2u_k u_y}{2v_k} \\ w_y = \pm\sqrt{d_i^2 - u_y^2 - v_y^2} \end{cases} \tag{1.10}$$

Naturally, other approaches exist, including methods based on vectors and derived from Powell's work [36], such as the one suggested by Gibson and Scheraga [37]. Their outcome is still a position derived from the estimation of 3 distances. This three-sphere method is light and simple. However, one clear limitation is that measured distances are estimated, thus noisy, and as such, the resulting position is an approximation. Additionally, y is not always found within the triangle formed by the three anchors. The anchors may all be in the same direction, thus causing dilution of precision [38] issues (see section 2.4.2).

1.3.4 Multi-sphere intersection

Because of the limited precision resulting from the 3-sphere intersection technique, it is tempting to increase the number of anchors ranged against, and generalise the formula. As soon as there are more than 3 anchors, the system has more equations than unknowns and appears to be overdetermined. However, adding more anchors, and thus more distances, does not always result in a better position estimation in a noisy context. This counter-intuitive result is easily understood with an example, and illustrated for other cases of projective geometry in [39] and [40].

Starting from measured distances (denoted $\tilde{d}$ to differentiate the noisy, measured distance from the actual distance d), the expected position $\tilde{y} = (\tilde{u_y}, \tilde{v_y}, \tilde{w_y})$ of the mobile can be extracted from (1.7) as follows:

$$(\tilde{u_y}^2 + \tilde{v_y}^2 + \tilde{w_y}^2) - 2(\tilde{u_y}u_i + \tilde{v_y}v_i + \tilde{w_y}w_i) = \tilde{d_i}^2 - (u_i^2 + v_i^2 + w_i^2) \quad (a_i) \tag{1.11}$$

We note that the location of the anchor i may be an approximation and add to the noise, but we consider for this chapter that the implementer configured the anchors locations with precision, to demonstrate that addition more measurements does not improve the location accuracy, even when the anchors locations are correct and known.

Thus, when ranging against four anchors (i, j, k, l), there are more equations than unknowns. By operating the pairwise subtraction described above $((a_i - a_j), (a_j - a_k), (a_k - a_l))$, the element $(\tilde{u_y}^2 + \tilde{v_y}^2 + \tilde{w_y}^2)$ disappears from each combined equation, and the system becomes:

$$\begin{aligned}
&\tilde{u_y}(u_i - u_j) + \tilde{v_y}(v_i - v_j) + \tilde{w_y}(w_i - w_j) \\
&= \frac{1}{2}((\tilde{d_i}^2 - \tilde{d_j}^2) + u_i^2 + v_i^2 + w_i^2 - u_j^2 - v_j^2 - w_j^2)\tilde{u_y}(u_j - u_k) + \tilde{v_y}(v_j - v_k) + \tilde{w_y}(w_j - w_k) \\
&= \frac{1}{2}((\tilde{d_j}^2 - \tilde{d_k}^2) + u_j^2 + v_j^2 + w_j^2 - u_k^2 - v_k^2 - w_k^2)\tilde{u_y}(u_k - u_l) + \tilde{v_y}(v_k - v_l) + \tilde{w_y}(w_k - w_l) \\
&= \frac{1}{2}((\tilde{d_k}^2 - \tilde{d_l}^2) + u_k^2 + v_k^2 + w_k^2 - u_l^2 - v_l^2 - w_l^2)
\end{aligned} \tag{1.12}$$

By organising the different factors in a matrix, these equations can be represented as follows:

$$\begin{bmatrix} (u_i - u_j) & (v_i - v_j) & (w_i - w_j) \\ (u_j - u_k) & (v_j - v_k) & (w_j - w_k) \\ (u_k - u_l) & (v_k - v_l) & (w_k - w_l) \end{bmatrix} \begin{bmatrix} \tilde{u_y} \\ \tilde{v_y} \\ \tilde{w_y} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}((\tilde{d_i}^2 - \tilde{d_j}^2) + u_i^2 + v_i^2 + w_i^2 - u_j^2 - v_j^2 - w_j^2) \\ \frac{1}{2}((\tilde{d_j}^2 - \tilde{d_k}^2) + u_j^2 + v_j^2 + w_j^2 - u_k^2 - v_k^2 - w_k^2) \\ \frac{1}{2}((\tilde{d_k}^2 - \tilde{d_l}^2) + u_k^2 + v_k^2 + w_k^2 - u_l^2 - v_l^2 - w_l^2) \end{bmatrix} \tag{1.13}$$

Such structure is well-known in linear algebra and has the form:

$$Ax = b$$

where A is an invertible matrix representing m equations with n unknown, and x is the vector in $\mathbb{R}^n$ that minimizes $||b - Ax||$ with respect to the inner product in $\mathbb{R}^m$. The optimal solution for this system, starting from the system of linear equations built above, is the vector x so that:

$x = (A^T A)^{-1} A^T b$

Therefore, finding x requires finding the determinant of A (this is needed, as $(A^T A)^{-1} = \frac{1}{det(A^T A)} adj(A^T A)$). In our case, this determinant is expressed as:

$$\begin{aligned}
d_A = (u_i - u_j)((v_j - v_k)(w_k - w_l) - (w_j - w_k)(v_k - v_l)) - (u_j - u_k)((v_i - v_j)(w_k - w_l) \\
-(w_i - w_j)(v_k - v_l)) + (u_k - u_l)((v_i - v_j)(w_j - w_k) - (w_i - w_j)(v_j - v_k))
\end{aligned} \tag{1.14}$$

An examination of the coefficient matrix clearly shows that each row represents the distance

between the points (i, j), (j, k) and (k, l) respectively, along each dimension (u, v, w). A direct consequence of this structure is that the determinant is zero if the points (and therefore the anchors) are on a straight line. This configuration would not be surprising in a public venue (*e.g.,* shopping mall main aisle with Wi-Fi access points aligned along a straight walking path), and the mobile has no direct mechanism to detect in advance the anchors relative positions. In an indoor setting where most anchors are commonly at the same height, the third column of the coefficient matrix also becomes a 0 vector, here again causing the determinant to be 0.

However, the measurements in a real environment are noisy, and the mobile may find a non-zero set of values for the w dimensions. These values may be close to 0 (but not null). Then, the determinant may be non-zero and small if the anchors are close to a straight line. In all cases, the non-zero determinant value is very sensitive to measurement errors. As the vector x is computed using the $\frac{1}{det(A^T A)}$, a small error, resulting in a small non-null determinant may have a dramatic effect on the coordinates found (as if $det(A^T A)$ is small, then $\frac{1}{det(A^T A)}$ is large).

This difficulty makes that there is often no great advantage in using more than 3 anchors in noisy environments when using the $Ax = b$ form for the multi-sphere intersection resolution.

The above does not mean that using more than 3 anchors is never a good choice. It means that when computation cost matters, 3 anchors is a "good enough" choice. When more than 3 anchors are used, a linear method cannot be used. Instead, a minimization technique is needed, with the downside of a higher computation cost.

A simple technique can still leverage equation (1.7) [41], by attempting to minimize the error (e) between the positions computed from the distance to each anchor. Thus for each set of n anchors ranged against, the mobile's goal is to compute:

$$\min_{y} \sum_{i=1}^{n} (||y - i|| - \tilde{d}_i)^2 \tag{1.15}$$

There are three different ways to compute the solution [42]. The most common approach is to use calculus, by computing the derivative of the error expression in (1.15) as:

$$\delta(e) = 2 \sum_{i=1}^{n} (||y - i|| - \tilde{d}_i) \frac{y - i}{||y - i||} \tag{1.16}$$

This regression can be computed with standard gradient descent techniques, by moving by steps the estimated position of y based on the gradient value. This equation will become important in section 7.4.

1.4 Comparison techniques

Measuring distances or ToF can be inaccurate, but also presents specific difficulties. RSSI methods rely on knowledge of the radio parameters of the transmitter (that the receiver does not always have). ToF methods consume airtime to pass values around and limit clock drifts, which causes scale issues in high-density environments. Therefore other methods were designed that rely on a single side transmission (thus offering higher scalability that ToF techniques), but without the need to know the transmitter parameters.

1.4.1 Time Difference of Arrival

These alternative techniques are especially useful in high-density environment, for example a factory or warehouse where thousands of assets (parts or boxes) need to be located to maintain operational efficiency. In this context, the asset typically does not need its own location. It is often a constrained device, and the primary requirement is a process by which the asset sends a single message (at intervals) that can be used to deduce the asset position.

If anchors are positioned around the asset location (*e.g.,* at the edge of the factory floor), then they may be able to deduce the asset position from the time it takes for the asset message to reach each of them [43] [44]. The strict ToF may not be known (as this would require the asset to send an interval or a clock value that can be used by the anchors). However, the anchors can compare the time of arrival of the asset signal, and use this information to deduce the relative position of the asset to each of them.

In other words, if the signal takes 10 nanoseconds more to reach anchor 2 than to reach anchor 1, then the asset is 3 meters (the distance traveled by the signal in 10 nanoseconds) farther to anchor 2 than it is to anchor 1. A precise distance to the asset is not determined, only a relative distance comparison, forming an hyperbola of possible positions that are 3 meters farther to anchor 2 than to anchor 1. This comparison is repeated for each available anchor pair, as illustrated in Figure 1.4.

Formally, in a two-dimensional domain, the asset at position $y = (u_y, v_y)$ sends a signal received by anchor i sitting at position $i = (u_i, v_i)$ and by anchor j sitting at position $j = (u_j, v_j)$. The distance d_{yi} from y to i can be expressed with the simple Pythagorean expression illustrated in Figure 1.4:

$$d_{yi} = \sqrt{(u_i - u_y)^2 - (v_i - v_y)^2} \tag{1.17}$$

The same logic is applied to express the asset distance to anchor j. The difference in distance between the asset and both anchors follows the same reasoning structure as the ToF method. If the asset signal reaches anchor i at time t_i and anchor j at time t_j, and knowing that the signal travels at the speed of light (c), then the difference between the distance of y to i and j

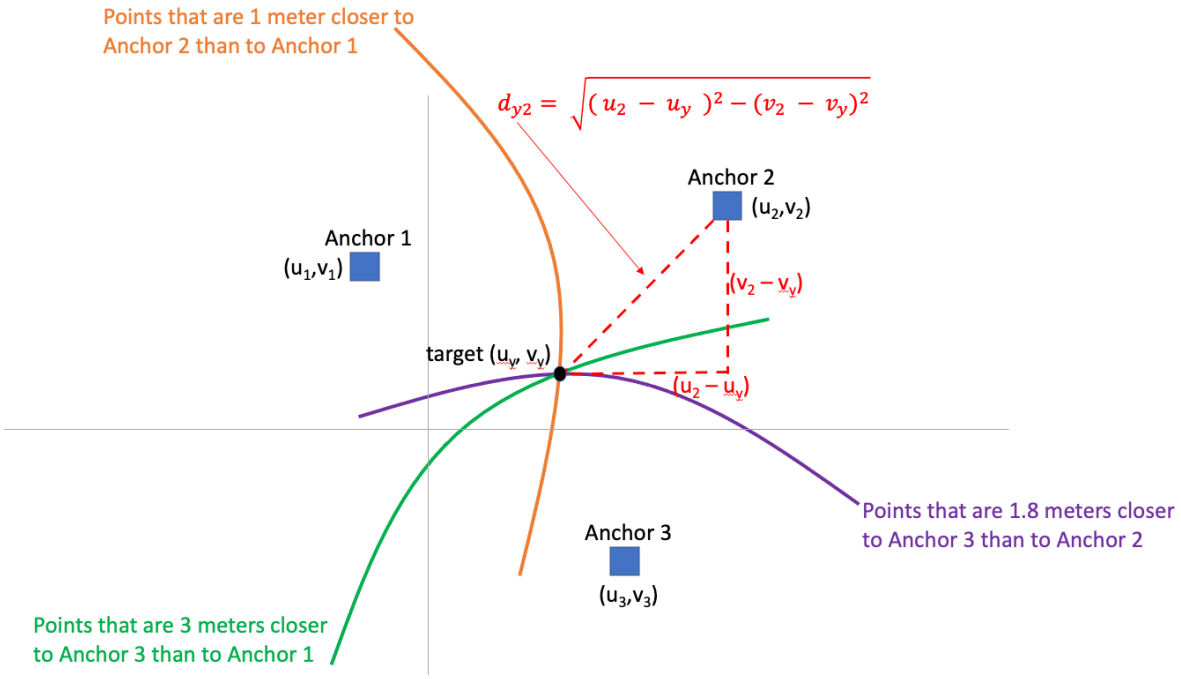


Figure 1.4 – Location determination with hyperbolic intersection in TDoA

is simply the difference of arrival time multiplied by the speed of light:

$$d_{yj} - d_{yi} = (t_j - t_i)c \tag{1.18}$$

As these distances are expressed in equation (1.17), a TDoA equation can be written that expresses this difference between these distances:

$$d_{yj} - d_{yi} = \sqrt{(u_j - u_y)^2 - (v_j - v_y)^2} - \sqrt{(u_i - u_y)^2 - (v_i - v_y)^2} \tag{1.19}$$

This equation is the squared-root form of the standard $\frac{(x-h)^2}{a^2} - \frac{(y-k)^2}{b^2} = 1$ equation for a hyperbola.

With 2 anchors, one hyperbola is formed and naturally the position of the asset is anywhere on the hyperbolic curve. A third anchor k allows the determination of the hyperbolas to the anchors i and k and to the anchors j and k. As the equations are of the second degree, the intersection of each hyperbola pair (and two conics in general) provides 4 solutions (therefore 4 intersection points). These points may all be real and distinct, two real and two imaginary or all imaginary. Two or more points may also coincide. Therefore a set of 3 anchors may be sufficient to find the position of y on a plane, but in other cases a fourth anchor is needed, and a fifth if the problem is presented in $\mathbb{R}$.

The solution of such system can be found with the method of simultaneous equations. How-

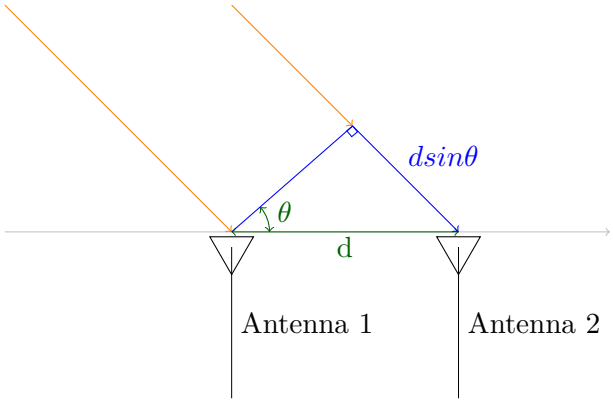


Figure 1.5 – Principles of AoA measurements

ever, one difficulty that we will examine further in chapter 3 is that the collected times of arrival are noisy in real deployments. Just like for the ToF approach, a comparison of the times of arrival (ToA) means that the location engine must be able to observe the ToA on each anchor within the same global time reference. As each anchor reports the ToA from its own clock, the same local significance issue arises as with ToF. Therefore, the common practice is to designate a primary anchor, that sends a synchronization message to the others (over the air or over the wire) to set a reference time. This technique sets a common clock, and limits individual clock drifts to the synchronization interval. An example is provided in Section 3.3.3.

1.4.2 Angle of Arrival

Each method that attempts to measure the times (difference of arrival, time of flight) seems to face the problem of clock synchronization. Therefore another technique of interest is one where no times are measured [45]. Rather, the method simply measures the angles at which a signal arrives (thus the name of the method, Angle of Arrival, AoA) to one receiver with multiple antennas or radio chains. This method presents this important requirement, that the receiver needs to incorporate more than one antenna. The signal from a transmitter, at the scale of the antenna separation, is seen as arriving at approximately the same angle to both antennas. However, one antenna will be closer to the transmitter than the other, and will receive the signal first. As the signal reaches the second antenna, a moment later, a phase change should be observed as shown in Figure 1.5, because the signal needs to travel an additional distance of $dsin(\theta)$ to reach the second antenna (where d is the distance between antennas, a known quantity, and θ the angle of arrival of the signal).

The idea behind the determination of the angle of arrival is easily understood as follows. As the angle of arrival between antennas is considered to be the same, the difference between both signals (besides a difference in attenuation due to the difference in distance and noise between

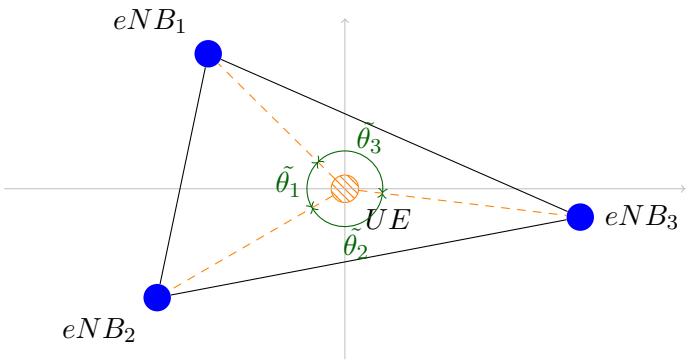


Figure 1.6 – Finding the UE position from angle measurements to known ENB positions

receiver figures), will be the phase shift Δ_ϕ between the first and the second antennas, that can be expressed as follows:

$$\Delta_\phi = -2\pi \frac{dsin(\theta)}{\lambda} \tag{1.20}$$

where λ is the frequency of the signal (which is also known). As the shift in signal can directly be measured by comparing the phase on each receiver, deducing the common angle θ can simply be done by reversing the equation:

$$\theta = arcsin\left(\frac{\Delta_\phi \cdot \lambda}{-2\pi d}\right) \tag{1.21}$$

In practice, the signal is noisy, and a difficulty is to determine the exact time, and the exact signal component, that should be retained as the main signal. A common technique is to use the MUltiple SIgnal Classification algorithm (MUSIC [46]). This technique and the issue of time of arrival determination will be examined further in Section 4.4.2.

Once the angle of arrival of the signal coming from one anchor has been found, the same operation can be repeated with one or more signals coming from other anchors. If the position of each anchor is known, the location of the mobile can then be deduced by a simple geometric method, illustrated in Figure 1.6. This method compares angles to different known sources, and is therefore called angulation. Measurements to at least three anchors are needed to evaluate the location of the mobile, even in the absence of noise (making triangulation the common embodiment and name for this method). Between the mobile and each set of two anchors, a triangle can be drawn where the mobile occupies one apex and each anchor each other two apices. From any arbitrary direction used as angle of value 0 in the measure of AoA, the mobile has determined the angle to each anchor, and the sum of both angles is easily found. When the mobile is within the triangle formed by the three anchors, the sum of the angles from the mobile to all anchors should equal π radians (modulo 2π). In a noisy environment, the angles

measurements are approximations, and a simple scalar method can be used to reduce the error, converting each measured angles $\tilde{\theta_i}$ to a reasonable estimation $\mathring{\theta}_i$:

$$if\ \tilde{\theta_1} + \tilde{\theta_2} + \tilde{\theta_3} \equiv m\pi,\ then\ \mathring{\theta}_1 \equiv \frac{\tilde{\theta_1}}{m},\ \mathring{\theta}_2 \equiv \frac{\tilde{\theta_2}}{m},\ \mathring{\theta}_3 \equiv \frac{\tilde{\theta_3}}{m} \pmod{2\pi} \tag{1.22}$$

If the positions of the anchors are known, the triangle they form can be drawn, and the task is merely to position the mobile in a position where the vertices formed from the evaluation of the angles θ_i are closest to each anchor.

The accuracy of this method depends on the precision of the estimation of each angle of arrival θ_i, and naturally on the knowledge of the positions and distances between the anchors [47].

1.5 Adding a Filter

The Kalman filter [48] is also a very popular technique for noisy measurements and computations aiming at finding the position (and possibly other elements of movement, such as speed and acceleration) of an object, especially when the object moves, which is the case for FTM. The efficiency of the Kalman filter comes from its ability to include the prediction of a new state (*e.g.,* the object position, speed, acceleration) with the observed (but noisy) state, and combining the two to output the best estimation of the actual state. In a noisy environment with stochastic measurements, the Kalman filter helps cut through the noise and converge rapidly on correct values. The process of the Kalman filter depends very much on what elements are measured, what modifies them, and how a state prediction is made. All these elements can be fairly confusing if only named generically. Therefore, and for better clarity, we will use FTM as a driving example in this section. But the principles can be applied to any case where both a prediction and a measurement can be used to evaluate a state.

1.5.1 Standard Kalman Filter

The Kalman filter is an iterative process. In the case of FTM, it occurs each time the ISTA obtains a new range from exchanges with an RSTA, *i.e.* at the end of each ranging burst. The same logic can be applied to any other technique that makes an estimation of distance to a reference anchor, then determine a new distance as the object moves. At that time, the measured distance $\tilde{d}$ to the client and the anchor known position are used to compute a circle at distance $\tilde{d}$ from the anchor. The role of the filter is then to determine if this new distance (or position) measurement matches the distance or position that would be expected from the previous series of measurements.

The filter is instantiated through several steps. In a first step, a new predicted state at

iteration k, X_{k_p}, is predicted from the previous state determined through the Kalman filter process, X_{k-1} (or from a seed initial state for the first iteration). A state matrix X_k can therefore be written, that represents the values of the various variables (*e.g.,* position x, y, z and associated speeds in each dimension, $\dot{x}$, $\dot{y}$, $\dot{z}$ at iteration k, as follows:

$$X_{k_p} = AX_{k-1} + Bu_k + w_k \tag{1.23}$$

where A is an adaptation matrix, used to modify X_{k-1} into a format compatible with the addition of the other values, while incorporating the time interval since the previous iteration, that quite logically will modify the position. For example, for a simple position and speed matrix at interval ΔT, $X_{\Delta T}^T = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}]$, A could be in the form:

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In equation (1.23), u_k is the control variable matrix, *i.e.* the elements that modify the position and speed beyond the effect of the time interval. In the case of a user moving on a floor, u_k is typically a 1x1 vector representing the user acceleration at iteration k. B is therefore a matrix representing how this acceleration modifies the position and the speed. For example, a moving vehicle accelerating at a constant rate a along a single axis will have a position at time t defined by $\frac{at^2}{2}$ and a speed defined by at, thus u_t would be defined as $u_t = [a]$ and after interval ΔT, $B^T = [(1/2)a\Delta T^2\ a\Delta T]$. In the case of FTM, acceleration is often expected to be null for a walking user, but we mention this component as it may appear in other cases where acceleration plays an important role (*e.g.,* localization technologies integrated into a vehicle). w_k is the predicted state noise matrix, *i.e.* the noise in the prediction of the new state. Thus the predicted state matrix X_{k_p} is determined by the previous state, modified by the elapsed time and the variables affecting the object position and speed.

The next step is to determine the state covariance matrix for iteration k, P_{k_p}. This matrix is computed as follows:

$$P_{k_p} = AP_{k-1}A^T + Q_k \tag{1.24}$$

where Q_k is the process noise covariance matrix, *i.e.* covariance of the expected noise in the measurement process (not the measurement itself, but the way we measure, in the case of FTM or another technique measuring ToF, this can represent calibration noise between the signal at

the antenna and the timestamp estimator, for example), and P_{k-1} the process error at step $k-1$. In a simplified model where we track a position x and a velocity $\dot{x}$, P_k can be in the form:

$$P_k = \begin{bmatrix} \sigma_x^2 & \sigma_{x\dot{x}} \\ \sigma_{\dot{x}x} & \sigma_{\dot{x}}^2 \end{bmatrix}$$

In the case of FTM and many others where there is no relationship between the errors in the process of measuring each variable (the position and the velocity are measured independently, for example position with FTM and speed with inertial sensors, and the error of the process of measuring one has no bearing on the error of the process of measuring the other), the matrix is often simplified to only keep each variable variance component (the error squared for each variable, position and speed, or the variance in the error for each variable), and:

$$P_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_{\dot{x}}^2 \end{bmatrix}$$

A has the same meaning in equation (1.24) as in equation (1.23). Thus P_{k_p} predicts the error in the process of position and speed estimation, based on the previous prediction and the noise in that prediction.

The position and speed is not only predicted, but also measured, with the updated measurement representation of the state matrix, Y_k, defined as:

$$Y_k = CX_{k_M} + Z_k \tag{1.25}$$

where X_{k_M} is the state measurement matrix (the state matrix as measured), modified by the adaptation matrix C (that nulls the elements of the state that are not measured, for example the system may only measure the position but not measure the speed, in which case C will apply a 1 coefficient to the position components of X_{k_M} and a 0 coefficient to its speed components), and Z_k the measurement noise (or uncertainty).

P_{k_p} and Y_k are used to compute the Kalman gain, at the heart of the process:

$$K = \frac{P_{k_p}H}{HP_{k_p}H^T + R} \tag{1.26}$$

where R is the sensor noise covariance matrix, *i.e.* the measurement error. In the case of FTM, a typical source is multipath, that causes the evaluation of the timestamps (and therefore the distance) to be biased when the retained signal is not the direct LoS but a reflected signal (we will come back extensively on this issue in Section 4.4.1). H is a transformation matrix required to maintain compatibility between the different objects of the equation (for example, for FTM and a 6x6 diagonal P_{k_p} matrix showing the covariance values of positions and speed in (x, y, z), H and H^T would each be the 6x6 identity matrix).

From the form of the equation, it is clear that K is a value between 0 and 1. The goal of the Kalman gain is to determine which part of the new state should be imparted to the new measurements, and which part should be imparted to the prediction of the state. Indeed, if R is small (the measurement error is small), K tends toward 1, with the consequence that the adjustment (below) will rely more strongly on the measurement update. If R is large, then K tends toward 0, with the consequence (below) that the adjustment will rely more strongly on the state prediction (and the adjustment caused by the Kalman gain, that incorporates how much the measured value differs from the predicted value, will be a small contributor). Therefore the new state X_k is determined as:

$$X_k = X_{k_p} + K[Y_k - HX_{k_p}] \tag{1.27}$$

Thus the new state is computed as the predicted state, modified by the Kalman gain multiplied by the difference between the observed state and the predicted state. A small Kalman gain will result in the next state giving more importance to the predicted state. A large (closer to 1) Kalman gain will result in the next state giving a larger importance to the measured state.

Once this determination is completed, and in preparation for the next iteration, the process covariance matrix is updated:

$$P_k = (I - KH)P_{k_p} \tag{1.28}$$

where I is the Identity matrix. P_k will be used at iteration $(k+1)$ to compute the process covariance matrix P_{k+1_p}.

1.5.2 Extended Kalman Filter

As FTM collects distance information (and not directly positions and velocity elements), the position in the X_k state matrix is computed through the standard Euclidean transformation from equation (1.7). The function is not linear, and linearity is one assumption of the standard Kalman filter process. In order to accommodate for scenarios where the equation is not linear (and those where the noise is not Gaussian around 0), an extension to the Kalman filter (the Extended Kalman filter, abbreviated EKF [49]) is rather used. The general principles are the same, with X_{k_p} expressing the function f leading to the new predicted state (which, again, is not linear, but needs to be derivable):

$$X_{k_p} = f(X_{k-1} + Bu_k) + w_k \tag{1.29}$$

Similarly, the updated measurement representation of the state matrix, Y_k shows the function

g leading to the new measurement, with:

$$Y_k = g(X_{k_M}) + Z_k \tag{1.30}$$

The EKF process uses derivation to linearize the function at each point of each iteration k, thus first looking for the predicted matrix of partial derivatives (the Jacobian), instead of looking directly for the predicted state:

$$F_k = \left.\frac{\delta f}{\delta X}\right|_{X=X_p} \tag{1.31}$$

Similarly, the linearized observation equation H_k is sought, instead of finding directly the measurement representation Y_k:

$$H_k = \left.\frac{\delta g}{\delta X}\right|_{X=X_p} \tag{1.32}$$

With these changes, the state covariance matrix for iteration k, P_{k_p}, becomes:

$$P_{k_p} = F_k P_{k-1} F_k^T + Q_k \tag{1.33}$$

With the same logic, the Kalman gain becomes:

$$K = \frac{P_{k_p} H_k^T}{H_k P_{k_p} H_k^T + R} \tag{1.34}$$

And thus the new state X_k is determined as:

$$X_k = X_{k_p} + K[Y_k - g(X_{k_p})] \tag{1.35}$$

In preparation for the next iteration, the process covariance matrix is updated as:

$$P_k = (I - KH_k)P_{k_p} \tag{1.36}$$

One great advantage of such transformation is of course the ability to locally linearize the equations, and thus compute location (and velocity) without a prior method. However, it is not an optimal estimator, especially at points where the measurement and state diverge. For example, while an ISTA is exchanging repeated bursts with a given RSTA, the Kalman gain decreases and the filter gets better and better and finding the right position with the prediction. As soon as the ISTA starts ranging with another RSTA, and if this RSTA displays dramatically different performances numbers than the previous one (*e.g.,* very different range precision figures [*e.g.,* because of a LoS vs nLoS scenario], or different LCI precision), then suddenly the predicted state and measurements diverge brutally and the Kalman gain brutally increases. To avoid this

issue, the implementer needs to be careful and start a n individual Kalman process for each RSTA. Otherwise, the effect of this change is an apparent sudden change of direction in the user trajectory, even if the user continues walking in a straight line.

The Kalman filter is only one of the many possibilities to compare a predicted state to a measured state. It is widely used for objects in movement (and thus for FTM, as will be seen in Section 7.4). However, other filters exist. For example, the α - β filter is simpler (it does not require a system model) and can also be used to establish the validity of new measurements. This filter will be used for example in Section 4.4.2 with FTM, to bound the timestamps received on both sides. Particle filters [50] substitute the linear projection of the next state found in the Kalman filter, with a sequential Monte Carlo method. All filters present the property of limiting the error when the next measurement is not available or very noisy. However, all filters, having to arbitrate between a measurement and a predicted value, may wrongly evaluate as noisy a measure that is valid but collected from a stochastic trajectory (*e.g.,* sudden turn). Thus filters limit the noise, but find limitations in environments where the prediction of the next value is made difficult by local circumstances.

1.6 Conclusion

In this chapter, we presented the primary techniques used to obtain a location output from the exchange of radio signals. Some techniques solely rely on the signal itself, leveraging the RSSI, and converting the signal strength into an approximation of the distance to the transmitter. The limitation of this method is the need to know the transmitter RF characteristics. Another method evaluates the time of flight of a signal between a transmitter and a receiver. No knowledge of the transmitter is needed, but an exchange of timestamp values supposes a synchronization of the clocks on both sides, or at least a bounding of the clock drifts. Once distances are estimated using either of these techniques, a three-sphere geometric method can be used to find the receiver location. If more than three anchors are available, the evaluated distances can be positioned in a matrix, and a minimization algorithm can be used to find the mobile location.

Instead of measuring the time of flight, the Time Difference of Arrival method measures the time of arrival of a given message on multiple receivers, then uses the time difference of arrival to compare the distance between the emitter and the different receivers. Each comparison draws an hyperbola of possible positions between two anchors, and multiple measurements are used to find the intersection of the hyperbolas formed that way. Because time is involved, the receivers

also need to report the arrival time within the same frame of reference, causing the drift issue to appear in this method as well.

Another method removes this limitation, by solely focusing on measuring the phase shifts at which a signal reaches a receiver with multiple antennas. The method is fundamentally geometric, but supposes a receiver with at least 2 or more antennas or radio chains.

It is thus clear that each method has advantages and downsides, trading simplicity for inaccuracy, or circuit complexity for cost. A mobile device still incorporates several of these systems, and should therefore be able to use them in turn or in combination to compute location. In the next chapter, we will examine how this localization is performed outside, and will attempt to bring these solutions to the indoor environment.

Chapter 2

Using Long Range Location Solutions

2.1 Introduction

In many ways, outdoor location seems to have been solved in many places where radio technologies are accessible. If you have a smartphone, and can connect it to the Internet to display a real time map of the area, you will also likely be able to display your location on that map, with an accuracy of a few meters. Thus it would be tempting to postulate that these technologies could simply be brought inside. This may mean using them directly for indoor navigation, or deploying inside technologies that have the same characteristic as these methods that have been successful outside. In this chapter, we study the outdoor localization technologies and evaluate how they can be ported to an inside world. We will start with GPS, and will examine the use case that this technology focuses on, and how it achieves an accuracy of a few meters. We will then look at cellular location, and will see how multiple techniques have been attempted to provide accurate location, not only for the client device, but also for the infrastructure. We will then see how these technologies perform when they are brought near or inside buildings, and will envision the scenarios where they can be used directly, or when they need to be complemented by indoor techniques.

2.2 Localization With GPS

Using GPS for location tracking has become a natural part of our outside connected life. However, the availability of the technology, and its ability to provide accurate location, rely on a combination of historic decisions and technical breakthroughs.

2.2.1 American GPS Availability

After The American National Air and Space Administration (NASA) started mastering the launch of rockets into space, in 1959, the US Navy requested the deployment of a series of satellites (the Transit System [51]) to track the position of nuclear submarines. The logic was

opposite to that of modern GPS. The transmitters were in the submarines, and the satellites were the receivers. The technology was also different. Transit used Doppler shifts in signals, the same technique the US Army had devised to detect if the first Russian satellite, Sputnik, was above the American soil.

The American Army soon determined that the opposite logic was needed for their ground forces: the ability for a vehicle to know its precise location on earth, and of course compare this location to the known position of enemy assets obtained through aerial photography. The first NAVSTAR test satellite was launched in 1974, and by 1978 the deployment of 11 Block I Global Positioning System (GPS) was initiated.

The system was first reserved for military usage. However, in 1983, the crew of Korean Air Flight 007, en route from Anchorage to Seoul, made a navigation mistake and the aircraft deviated from its intended trajectory. The Soviet estimated that the commercial identification was a decoy for a spy operation, and the plane was destroyed by a Soviet Su-15 interceptor, killing close to 300 people. Touched by the event, President Reagan ordered that GPS be made available to civilian use, to improve navigation and increase air traffic safety.

A few years later (1989), the first hand-held GPS devices appeared on the market, right at the time the US Air Force was launching improved GPS satellites (Block II). The US Department of Defense began decreasing the accuracy of GPS readings for non-military use, fearing to provide an advantage to the US adversaries. This lower precision signal was the one made available to civilian receivers. In 1998, the US army started launching GPS III satellites, with a military-grade (encrypted) high accuracy signal, and lower accuracy signals for aircraft and civilian use. The GPS system today includes 32 satellites, 24 of which are active at any time.

Because the American GPS solution was the first to be made available to civilian usage, it is the most widely used solution today, and its applications range from location to time synchronization. However, other equivalent systems were deployed since, such as Galileo (European Union), GLONASS (Global Navigation Satellite System, Russia), NavIC (Navigation with Indian Constellation, India), Michibiki (or QZSS, Quasi-Zenith Satellite System, Japan) and Beidou (China).

2.2.2 Position Determination with GPS

Although these different solutions present major differences (in particular for accuracy augmentation and security), they rely on the same principles for basic location determination [52]. These principles are useful for our study, because FTM will attempt to reuse similar ideas, but will also make critical simplifications along the way.

At the core of GPS localization, the end device includes a receiver that collects signals from several specialized satellites. Each signal includes a timestamp and the emitting satellite position. The satellites use an atomic clock so as to ensure the precision of the timestamp, solving at the

source the drift issue that littered the previous chapter. The receiver then proceeds to compute its own location on earth through the following steps (Mohan et al.[53] provides more details):

— For each satellite, correct the provided position based on time and date, accounting for the fact that the satellite trajectory is not a perfect ellipse.

— Determine the rough distance $\tilde{d}$ between the receiver and each satellite. This is done by multiplying the difference between the receiver time (T_R) and the timestamp of the GPS signal (T_S) by the speed of light c, with $\tilde{d} = c(T_R - T_S)$, in a very classical ToF fashion. This evaluation, commonly called pseudorange, is just a working approximation, as the client clock is of consumer grade. Although the receiver clock is realigned often, based on the computed location and the time sent by the satellites, the clock drift cannot be accounted for accurately, and the time of flight is always impaired by the imprecision of the receiver clock. Thus clock drift is only solved on one side.

— By comparing the pseudoranges to each detected GPS satellite, proceed through an iterative process to reduce the error and conclude on its own position.

This last step is the object of intense research to increase the precision in the presence of noise (*i.e.* cases when one or more readings provide inconsistent results with the others) and to allow for computation of an acceptable location value when few satellites are in range (*e.g.*, because of obstacles) or when the detected satellites are close to each other, causing dilution of precision [54] as will be seen later in this chapter. When enough satellites are in range, a classical solution uses the Least Square (LS) technique.

The logic of LS in this case is as follows. The Euclidean distance d_i between the receiver y with coordinates (u_y, v_y, w_y) and a satellite i (u_i, v_i, w_i) can be expressed in the standard form:

$$\sqrt{(u_i - u_y)^2 + (v_i - v_y)^2 + (w_i - w_y)^2} \tag{2.1}$$

When the distances to multiple satellites are compared, the time difference ΔT_i between the receiver and the satellite clocks causes a distance error that can be expressed as $c\Delta T_i$. This error, factored in each Euclidean distance equation to n satellites, results in the system:

$$\begin{cases} (u_1 - u_y)^2 + (v_1 - v_y)^2 + (w_1 - w_y)^2 = (d_1 - c\Delta T_1)^2 \\ (u_2 - u_y)^2 + (v_2 - v_y)^2 + (w_2 - w_y)^2 = (d_2 - c\Delta T_2)^2 \\ \qquad\qquad\qquad\qquad\qquad\qquad \cdots \\ (u_n - u_y)^2 + (v_n - v_y)^2 + (w_n - w_y)^2 = (d_n - c\Delta T_n)^2 \end{cases} \tag{2.2}$$

This system can be converted to a matrix, where each iteration of the observed distance $\tilde{d}_i$ can be expressed as the estimated change of position Δy of the receiver and the distance error

caused by its clock drift at epoch t_i, transformed by a scalar coefficient a_i^m of the pseudorange to each of the n satellites, in the form, for m observations:

$$\begin{bmatrix} a_{u_i}^1 & a_{v_i}^1 & a_{w_i}^1 & c \\ a_{u_i}^2 & a_{v_i}^2 & a_{w_i}^2 & c \\ & ... & & \\ a_{u_i}^n & a_{v_i}^n & a_{w_i}^n & c \end{bmatrix} \begin{bmatrix} \Delta u_y \\ \Delta v_y \\ \Delta w_y \\ \Delta t_i \end{bmatrix} = \begin{bmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ ... \\ \tilde{d}_n \end{bmatrix} \tag{2.3}$$

The details of this transformation can be found in [55]. This expression is in the form $Ax = B$ where the components of the unknown vector x can be found from the m observations $\tilde{D}_i$ of the pseudorange to each satellite i, by computing:

$$\Delta x_i = (A_i^T A_i)^{-1} A_i^T \tilde{D}_i \tag{2.4}$$

When 4 or more satellites are available for observation, this solution can be used to find the coordinates of the receiver in $\mathbb{R}^3$. It is therefore important to note that the receiver needs one more satellite than the number of dimensions it operates into. Each line of the matrix forms a three-dimensional object, and the intersection of 3 three-dimensional objects may provide two solutions. A fourth object is therefore needed to find the solution that is likely closest to the object correct location. When these conditions are met, the location accuracy can reach 1 meter [56].

We will see in Section 3.2 that the expectations for accuracy highly depend on the use case. However, 1 meter makes GPS in line with the expectations of most other localization technologies (many of which achieve much less). Another interesting property of GPS is that the receiver is passive. Therefore, at least conceptually, the privacy of the end-user device is respected (as it does not need to emit any signal). The fact that the traffic is only downstream also helps with scalability (a set of 24 GPS satellites can theoretically serve an infinity of clients). These properties are useful to keep in mind as we profile the characteristics of a "modern" localization technology for indoor use.

2.2.3 Time-to-first-fix

The architecture of the GPS technique is also critical to evaluate its reusability indoor. Each GPS satellite continuously broadcasts a 30 second-long navigation message. The first 6 seconds contains data describing the current time and the satellite status. The next 12 seconds provide the elements needed to evaluate the satellite exact location (this part is called the ephemeris). The last 12 seconds of the message contain the almanac, which is the coarse orbit and status information for the other satellites in the constellation, an ionospheric model providing the expected largest error for a single frequency GPS receiver, and information to convert the GPS-

derived time into Coordinated Universal Time (UTC).

One key element of the almanac is that each frame contains 1/25th of the entire constellation data [57]. Thus, 12.5 minutes are required to receive the entire almanac from a single satellite. Naturally, a receiver is likely to receive data from multiple satellites, and can thus reconstruct the entire almanac faster. Additionally, a receiver may not need the elements for all satellites in the constellation (for example those that are on the other side of the Earth from the receiver standpoint) to determine the receiver current position. However, this structure makes that it can take more than a minute (and possibly several minutes) for a receiver to collect enough information to compute a first location estimation (this is the time-to-first-fix). In many outdoor applications, this delay is too long, and there is a very active research stream dedicated to reducing this delay [58] [59]. The same concern can be expressed for indoor location for mobile devices. A user in need of displaying a phone location on a map is unlikely to accept several minutes delay before the position appears. This means that, if GPS is used indoor, an augmentation technique needs to be put in place to reduce the time-to-first-fix. An augmentation or alternate technology must be able to provide a first location reading within a few seconds. This limitation will have consequences when the FTM architecture model is designed.

2.3 Localization with Cellular multilateration

All these elements make that GPS receivers equipped with other radio technologies (*e.g.,* smartphone with cellular capabilities and internal sensors) tend to augment GPS with other techniques to refine their position determination. If the device is connected to a cellular network, using that radio technology for localization becomes a natural solution.

2.3.1 Cellular Localization Technologies

The landscape of localization technologies relying on cellular signal is complex. In effect, civilian GPS resolves around the resolution of a single use case: a receiver in need of determining its position. By contrast, over 30 years of technology development, cellular technologies have faced multiple needs around localization [60]. In some cases, the primary goal might be for the client device (User Equipment, UE) to determine its own location, solely from cellular radio technologies or to augment other techniques. But in many other cases, the infrastructure also has a good reason to also want to determine a UE location.

There may be technical requirements for such determination. For example, resources optimization can cause the infrastructure to need to determine the UE location, and then allocate the UE to a channel or a cellular tower radio based on that location. Such allocation allows to load balance UEs between cellular towers, while ensuring that the UE is always assigned to a radio likely to provide satisfactory services.

eNB and gNB

A cellular tower may include one or more cellular radios, each simply called Base Transceiver Stations in the first generation of cellular communication defined by 3GPP. When the third generation (3G) was deployed, some nodes started incorporating WDCMA (Wideband Code Division Multiple Access) for better scalability. These base transceiver stations were called "nodes of type B", or "Nodes B". With 3GPP 4th generation (4G) and Long Term Evolution (LTE) technologies, the radio came to be called LTE Node B, or eNode B (eNB). With 5G, the radio became gNB. For the purpose of this thesis and for simplicity, we will call eNB any infrastructure cellular radio providing control and data communication with a UE (unless a particular radio generation needs to be specified).

Cellular technology has also long been used for communication to emergency service calls. In an emergency context, the caller is expected to be in distress and possibly unable to communicate clearly about location. Automatically communicating the location of the UE to the emergency services system may prove invaluable.

Determining the UE location can also have commercial benefits (geo-marketing, asset tracking, augmented reality, *etc.*) The accuracy of the location depend very much on the targeted application. For emergency services, a rough estimation with a 50-meter imprecision may be sufficient in rural environments, but indoor localization may need higher precision and accuracy, and also need to estimate the correct floor [61] [62]. Augmented reality requires an accuracy down to a few meters, to properly overlay polygons over the visual scene. This rich landscape has brought multiple techniques for cellular-based localization, most of them defined by the 3GPP standard, and that can be organized in three groups:

1. UE-based location techniques: in this scenario, the UE itself determines its own location, by measuring signals received from the eNBs, possibly augmented with additional elements. As this technique is closest to the intent of FTM, it will be examined more in depth.
2. Network-based localization techniques: a network location server computes the UE location, based on signal measurements resulting from the UE exchanges with the infrastructure. One attractive aspect of this technique is that it is virtually passive. The UE undergoes the exchanges it needs for its cellular control and data communication, and the location determination is achieved directly from these signals. Therefore, the system scales without limitation other than the number of devices that the cellular system can communicate with. However, efficiency implies a multiplicity of receivers, which may be

difficult to achieve in rural environments (macro-cells) or in urban environments with multiple buildings. The techniques used are similar in concept with the infrastructure-based indoor localization techniques, and will be examined in chapter 3.

3. UE-assisted localization techniques: the UE and the infrastructure exchange on location. The messages can contain measurements, estimations of parameters or reports intended to increase the accuracy of the location determination.

Of these three, the UE-based techniques are those that most resemble the logic of FTM, and are the focus of our analysis.

2.3.2 UE-based techniques

The mobile-based techniques leverage measurement of the signal received by the UE. Among them, Observed Time Difference of Arrival (O-TDoA) is defined in 3GPP TS 23.271, and relies on the UE measuring the time of arrival of signals received from multiple eNBs. This architecture is similar in spirit to the TDoA technique described in the previous chapter but in reverse (the eNBs send a signal that the UE uses for location, instead of the other way round in the previous chapter). However, in its generic form, this technique left us with the problem of time synchronization. If all eNBs send signals, the question of transmission coordination also comes to the forefront.

O-TDoA

With O-TDoA, eNBs are time-synchronized. Each error of one nanosecond translates to a range error of about 30 cm (a foot), as signals are transmitted at the speed of light. Instead of relying on eNB-to-eNB synchronization signals, it is common for participating eNBs to embark a GPS receiver, and use the timestamp in the GPS signals to adjust their own clocks, with a resulting clock accuracy of 100 nsec or better. Such accuracy provides a range error of at most 30 meters [63].

The position of the eNB antenna is also known, typically with an accuracy of 3 to 5 meters maximum. The eNBs then each send a timestamped frame to the UE at regular intervals, and the UE measures the difference between the arrival times of these frames. This information is used to find the position of the UE in a 2D Cartesian space. Practically, the UE chooses a reference eNB (with coordinates (x_1, y_1)), and receives a signal (called Positioning Reference Signals, PRS) from both the reference eNB, at time T_1, and from another eNB with coordinates (x_i, y_i), at time T_i. Because the UE can use several channels, the design choice was to coordinate the eNB transmissions, and thus send the signals at the same time, thus $(T_i - T_1)$ should represent the time offset between the receive time of the signals from both eNBs. This difference is the Real Time Difference (RTD). In this case (contrary to GPS), the position of the antennas does not

change. However, the signal transmission is also affected by the environment (reflections *etc.*), and the eNB synchronization may be imperfect. The UE may also imperfectly detect the exact time of arrival of the beginning of the signal from each eNB. This issue also affects FTM and will be described more in details in Section 4.4.1. As such, each signal measurement is affected by a general UE ToA measurement error labelled n_1 and n_i and that covers all these imprecision parameters. The time difference $RSTD_{i,1}$ of arrival of the signals (transmitted at speed of light, c) at the UE (with coordinates (x_t, y_t)) is then represented as follows:

$$RSTD_{i,1} = \frac{\sqrt{(x_t - x_i)^2 + (y_t - y_i)^2}}{c} - \frac{\sqrt{(x_t - x_1)^2 + (y_t - y_1)^2}}{c} + (T_i - T_1) + (n_i - n_1) \quad (2.5)$$

We see that this method deviates from the generic TDoA equation (1.17) by incorporating the measurement error directly into the formula. With one reference eNB and two neighbors, equation (2.5) translates into two equations with (x_t, y_t) as unknown. The system of equations is then solved with least-squares or weighted least-squares techniques.

Although the UE can compute its location from these measurements, we should mention that in many cases, the O-TDoA technique is in fact inscribed in a UE-assisted architecture leveraging the LTE Positioning Protocol (LPP). This is because the airtime (and channel space) consumption caused by the PRS transmission is only valuable if there are UEs in need of such signal and able to interpret it. Additionally, the infrastructure also has a need to locate the UE, even just for properly choosing the eNBs that will send the PRS. Therefore, in the UE-assisted architecture, the location server first queries the UE for its LPP support capabilities. Upon a positive response, the location server triggers the PRS transmission on target eNBs (these messages are called, in the architecture, LPP Provide Assistance Data). The Location server then queries the UE for the collected information through the LPP Request Location Information message. The UE then returns the RSTD measurement data, including then timestamps, the identity of the reference cell and of the others, and an estimation of the quality of TOA estimation for each measurement (n_i).

O-TDoA is an interesting variation from GPS when the source position (the eNBs in this case) is known and not mobile, because there is no need anymore to estimate a scalar coefficient of the range. Instead, the range itself can directly be estimated, and this logic will be applied directly with FTM.

Cell-of-Origin

The fact that the reference points (eNBs) are static and their position known also opens the door to even simpler localization determination techniques, that were implemented before LPP was defined, and that are still in use today because of their simplicity. These techniques rely on

measuring the signal from the active eNB, and obtaining the identification and location of that eNB. This is not always an RSSI-based determination. An archaic version of this technique is called Cell-of-Origin (CoO). With CoO, the radius of the eNB is estimated (for example based on its known transmit power). Each cell communicates its identification (Cell Global Identity, CGI) at multiple stages of the session exchange with the UE. The CGI includes the Mobile Country Code (MCC) indicating the country where the cell is located, the Mobile Network Code (MNC) showing the operator of the network, the Location Area Code (LAC) that points to a group of cell towers, and the Cell ID, which is the unique identifier of the cell tower. It is common that the tower would use directional antennas (sectors), allowing for a more accurate representation of the zone where the UE is likely to be as it receives the tower signal. With that information, the device can find from a database (local to the device or remote) the exact location of the eNB, and position the device near the center of the projected coverage zone. Quite clearly, this technique is limited in efficiency, as it requires offline database knowledge of the cell location.

Angle of Arrival

The general CoO method is also not very accurate of course, as a cell can be several kilometers-wide. The UE is then represented in the center of the cell zone, with an uncertainty circle that spans the entire cell area. Several augmentation techniques are commonly used. One of them, using angles of arrival (AoA), collects the signal from several eNB, and uses the simple geometric method in Section 1.4.2 to determine the area where all towers can be detected.

AoA is possible because most cellular devices incorporate more than one antenna. If the positions of three eNBs are known, the triangle they form can be drawn, and the task is merely to position the UE in a position where the vertices formed from the evaluation of the angles θ_i are closest to each eNB. In cellular localization however, two main problems occur:

— In practice, the signal is noisy, and a difficulty is to determine the exact time, and the exact signal component, that should be retained as the main signal. This issue goes beyond using a clever algorithm like MUltiple SIgnal Classification algorithm (MUSIC [46]). With multipath, it is common that a reflected component is taken as the primary signal. The reliability and the stability of the measured angles can be limited. This issue will be examined further as it also affects FTM.

— In many cases, the number of cellular towers in range of the UE is small. When one or more angles show high variance, the UE does not always have alternative options.

These two issues limit the usability of AoA alone. Wang showed in [64] that, in LoS conditions, AoA could provide satisfactory measurements, especially if the goal is merely to display the UE in an uncertainty circle of variable size. However, as soon as nLoS occur, the performances degrade and other methods need to replace AoA. When computing resources are not a constraint, AoA

can be augmented with clever algorithms to limit the effect of nLoS. For example, Zhang [65] suggest to use a classical Bayesian method to resolve the noise and nLoS issues. By subsequent observations of the signal (in fact of individual OFDM components, which dramatically increases the same size, as Zhang works on a 5G case, where the signal can include up to 3300 subcarrier components), the system evaluates which angle is likely to be LoS (as LoS always arrive faster than any reflected component). This logic is similar to optimizations for FTM that we will examine in chapter 4. This optimization reduces the error. Quite naturally, a large sample improves the accuracy. Increasing the number of antennas (which also increase the sample size) is of course another way to improve the accuracy [66].

Distance from Signal Strength

Over the years, the validity of the AoA method has been widely debated, and other techniques have been sought that would not suffer from the large error introduced by the reflected components. Among them, the simple idea to deduce the distance to the eNB from the measured signal loss along the path has long been offered as a valid alternative.

The principles were examined in Section 1.3.1. The same issue appears for cellular technologies as in the general case: the UE has no information about the eNB RF characteristics, especially as most cells use directional antennas. Even if the UE knows the details of the eNB antenna (for example through a database query about the eNB CGI, that includes the antenna identifier in the cell), it would not know from which area of the lobe it is measuring the signal. There are therefore alternative ways that have been proposed to evaluate the expected power received at the UE by attempting to ignore the gain component. The Friis transmission equation is a classical variant [67]. By assuming an average eNB transmission power and antenna gain, the Friis model simplifies equation 1.4 by fixing the two eNB variables (power and transmit gain).

However, one major obstacle that this technique faces, besides its inherent approximation, is that the space between the eNB and the UE is nowhere close to free space. Antennas are close to the Earth, which absorption and radiation affects the signal. Even if there is direct line of sight between the UE and the eNB antenna, surrounding obstacles may affect the signal. This effect was discovered by Augustin-Jean Fresnel in the 1820s, and zones that bear his name are modeled around the direct LoS axis to evaluate this effect. The effect depends on the position of the obstacle, *i.e.,* its distance to the UE and the EnB antenna on one hand, and its distance from the LoS axis on the other as displayed in Figure 2.1. As usual for these matters, the effect also depends on the wavelength (λ) of the affected signal.

Thus Fresnel defined several zones around the LoS axis. Obstacles in some of them (odd numbers) are destructive to the signal, obstacles in others (even numbers) reinforce the signal quality (because reflected components through these zones are primarily out of phase with

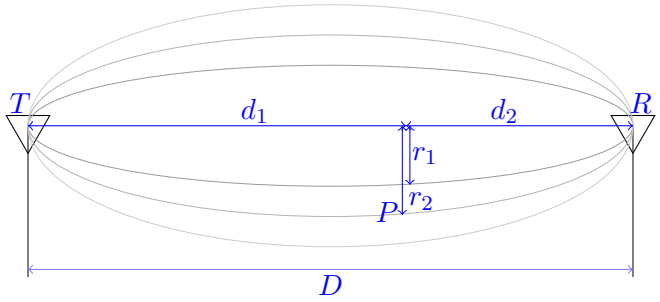


Figure 2.1 – Fresnel Zone

the main signal). This effect is easily understood by observing that the phases combine in a constructive or destructive manner depending on their relationship to the signal wavelength λ. If the reflected component matches λ exactly at a given point in space, then it reinforces the main component. If the reflected component trough matches the main component peak at a given point in space (and supposing same amplitude), they compensate each other exactly. Thus the limit between one zone to the next matches those points in space where the reflected component matches $\lambda/2$. Therefore the radius r_n of the n^{th} zone, at a point P between the transmitter T and the receiver R, positioned at distance D from each other, can be found by keeping in mind the relationship:

$$\vec{TP} + \vec{PR} - D = n\frac{\lambda}{2} \tag{2.6}$$

Supposing that the project of P on the LoS axis is at distance d_1 from T and d_2 from R, then 2.6 can be rewritten using P's coordinates in this system:

$$\begin{aligned} \sqrt{d_1^2 + r_n^2} + \sqrt{d_2^2 + r_n^2} - (d_1 + d_2) &= n\frac{\lambda}{2} \\ d_1(\sqrt{1 + r_n^2/d_1^2} - 1) + d_2(\sqrt{1 + r_n^2/d_2^2} - 1) &= n\frac{\lambda}{2} \end{aligned} \tag{2.7}$$

At this point, Fresnel makes an important assumption, that the distances d_1 and d_2 are usually much larger than the zone radius (and thus $r/d << 1$), and therefore that equation (2.7) can be simplified by using the binomial approximation for the square root, namely $\sqrt{1+x} \approx 1 + x/2$ (for $x << 1$), allowing (2.7) to be re-written as:

$$\frac{r_n^2}{2}\left(\frac{1}{d_1} + \frac{1}{d_2}\right) \approx n\frac{\lambda}{2} \tag{2.8}$$

This transformation offers a simple way to find the radius of the n^{th} zone as:

$$r_n = \sqrt{n\frac{d_1 d_2}{D}\lambda} \tag{2.9}$$

This model is very useful to evaluate the likely quality of a signal at any point away from the eNB. In particular, the first Fresnel zone is critical, and should be left free from obstacles as much as possible (general recommendations are around 20% obstruction max if possible, and never more than 40%).

If the first Fresnel zone is free from obstacles, then the signal quality should be acceptable. The obstacles in each Fresnel zone will still exert an influence. For location tracking, the core preoccupation is of course not only to receive a usable signal, but also model the influence of obstacles on the signal strength. the Fresnel model provides a good approximation of the degradation, but not a numbered evaluation of the resulting signal strength at the receiver level. It is also very manual. UEs do not have convenient mechanisms to draw a LoS axis to the eNB antenna and model the zones and their likely effect on the signal.

Modeling this influence in a systematic fashion, that can be automated based on the distance to the eNB and the environment, is a complex task. Semi-empirical models have been proposed. For example, Egli's model introduces a frequency-dependent empirical correction to equation (1.4) ($(40MHz/f)^2$)) for carrier frequencies in the 30 MHz to 1 GHz range. Free path loss model is better at short range, but Egli's model, relying on measurements he performed, offers a better estimation of the signal for 1 to 50 km ranges [68]. His proposed modification of the received power is as follows:

$$p_R = \left(\frac{40MHz}{f}\right)^2 \frac{p_T G_T}{4\pi d^2} A \tag{2.10}$$

Because the transmit power is known, a simple arbitration method (between Egli's path loss prediction and the free path loss prediction model) is to compute both values, then switch to Egli's model if the computed distance is larger than a given threshold (*e.g.*, 1 km), or adopt the free path loss model otherwise.

Position from Signal Strength and Scene Analysis

No theoretical or semi-empirical models (Egli, Friis or free path) can take into account the exact effect of the local terrain on the received signal strength. The estimated distance is therefore always an approximation. It may be sufficient in simple settings to evaluate the UE position, by performing lateration estimations from multiple eNBs.

Where angulation works with angles, lateration works from measured distances. Trilateration is a common term, although more than 3 distances are often used.

When using distances from three towers, trilateration is simply seeking the intersection of three circles as detailed in Section 1.3.3. In practice however, measurements are noisy and eNB antennas positions are not always known with perfect accuracy. The solution is therefore an approximation, and a common practice is to find the coordinates that minimize the errors of the system. Multiple techniques are available for this task. Some methods, like in [69], are entirely contained within the UE. However, many practitioners have concluded that a systematic and RF-based approach alone cannot reliably account for localities and leverage an augmentation technique called Scene analysis. A equivalent of this technique will come in the landscape of indoor location under the name fingerprint matching.

Scene analysis relies on collecting, at known locations, signal patterns (which eNB is detected at which signal level), and storing these profiles (or signatures) in a database (commonly called radiomap). When a UE tries to refine its location determination, it can then query the database and find the signature that best matches the signals it detects, and use the information to better pinpoint the UE location on a map. The UE then compares the signals it receives from all eNBs to the signals reported for the closest signature, and deduces its likely location (even if it is not exactly that of the signature).

The database does not exist for every location on earth where GSM/LTE (3G or higher) signal is available. However, with crowd-sourcing, forming this database from the mean locations of alike-RF reporting UEs has become easier. With the multiplication of uses cases where such database is useful (from commercial offering around hyper-accurate location [70], to ISP coverage validation [71]), such augmentation resource is becoming common. Here again, the technique is often not used alone, but augments the other techniques.

2.3.3 Design Challenges

For most users, outdoor location based on cellular techniques simply manifests as a blue dot on a map in regions where GPS is insufficient. However, it is important to keep in mind that location is often obtained by both sides (UE and infrastructure). This aspect presents fundamental design challenges. In most cases, the user is not queried (no "opt-in" mechanism), localization by the infrastructure just happens (often simply because it is needed to direct the UE to the best eNB). This is an issue for a "modern" localization protocol, in a world where privacy has become a central factor. It is also an issue because user location can be obtained also by bad actors, as illustrated in [72] and [73].

2.4 Challenges When Expanding to the Indoor Case

In the case of outdoor location, it is therefore common that multiple techniques are used together to provide the best possible location reading. These techniques can be carried to the indoor environment, but only to some subset areas of a building. GPS signals use different frequencies (1575.42 MHz for the first generation, or L1, satellites, 1227.60 MHz for the L2 generation, and 1176 MHz for the latest generation). These frequencies are optimal for the GPS outdoor use case (a downstream signal encoding up to 10 million chips per second, send through the atmosphere to an object in LoS). The atmosphere does affect the signal (in particular the ionosphere has a slowing-down and partial scattering effect called scintillation [74]), but the link budget allows the signal to reach an outdoor receiver in good conditions when LoS is achieved.

2.4.1 Signal Attenuation

However, near the ground, a clear line of sight is not always possible, and the GPS signal can suffer from different sources of degradation. Just like for cellular signals, nearby objects can cause reflection. When out of phase, the reflected component degrades the primary signal (multipath). Objects also absorb the signal (shadowing), causing the signal to be received, but weaker (shadowed signal) or to be lost (blocked signal) [75].

The effect of materials on the GPS and cellular signals (and RF signals in general) have been widely studied in the literature. Any material on the path absorbs some of the signal energy, causing a reduction of the signal amplitude. When the amplitude is below the receiver ability to distinguish the signal from the surrounding noise, the signal becomes unusable. The material of the obstacle and its thickness are the main elements that affect the signal transmission. Modeling this effect is a complex task, because the material is not always a constant. For example, the moisture content of a wall may dramatically affect its attenuation characteristics, as humidity affects the material dielectric constant ϵ_r (the amount of electric potential energy in the material) and its conductivity σ (its ability to forward electro-magnetic energy).

The outcome of such absorption is that the GPS signal does not penetrate well inside buildings. In some scenarios, using GPS indoor may be possible. However, the studies asserting the availability of the GPS signal indoor are often anecdotal in nature (*e.g.*, [76]). They observe that, although GPS is said to not penetrate buildings well, GPS signal can be used in tested locations. GPS can indeed be available indoor, either because the building uses thin material, or has large windows, and/or is not surrounded by other large buildings, thus allowing for LoS conditions for the building to multiple GPS satellites. However, as the building material becomes thicker, as the windows become smaller, as the building becomes larger, and as the density of surrounding buildings increases, the availability of GPS inside decreases to zero.

Another key factor is the position of the measured satellite in relation to the observer. If the

satellite is just above the horizon, with only a window in the line of sight, then the signal may still be perfectly usable indoor. However, if the satellite is at the zenith, and its signal has to go through multiple floors before reaching the user device, then the signal may be below the noise and unusable.

The same considerations apply to cellular signals, with an important variation: cellular towers are usually not in the sky, and the signal seldom falls from above. In most cases, cellular antennas are on pylons above the ground or on top of buildings, drawing a coverage that spans laterally from the antennas positions. As a result, the cellular signal penetrating buildings come commonly through the walls and windows, and less commonly from the top, and often does not have to go through multiple floors before reaching the UE. As a result, the attenuation of the signal is still large and dependent on the building material, but is often in the order of 10 to 20 dB [77] [78] [79], not because buildings are more permeable to cellular signals than to GPS signals, but because cellular signals encounter often less maternal than their GPS counterpart.

This advantage given to the cellular signal naturally comes with equivalent handicaps. If the primary cellular tower is on the other side of the building, then the signal has to cross multiple walls before reaching the user. If the primary tower is behind another building, the same difficulty occurs. In other words, the challenge may be displaced by trading an incident signal for a lateral signal, but the same problems caused by the presence of obstacles remain. The problems may be hidden by the common conclusion that an attenuation of 10 to 20 dB is often not sufficient to entirely block the signal, and that cellular communication is therefore often still possible indoor. But in the case of interest to this study, the goal is not to place or receive calls, but to locate the UE. For this purpose, the signal from multiple eNBs may need to be used. As least some of them will be on the "wrong side" of the building, and their signal may not be usable at the UE location for triangulation or trilateration. Additionally, some eNBs may be in LoS through a window, and the FPSL equation is a good model to estimate its signal and distance, but some others may be on the other side of a building, and neither Friis nor Egli provide a good model to convert signal to distance. The UE has no mechanism to factor this model asymmetry. Crowd-sourced database may correctly assess that the user is inside a particular building, but typically cannot determine where in the building with satisfactory accuracy. Thus, just like GPS, cellular localization is sometimes usable inside, but the usability is very dependent on locality (in which part, and on which floor of the building the UE is located). For our model in chapter 4, these difficulties mean that the trust put in the location obtained from cellular techniques reduce rapidly as the user enters and progresses deeper in the building.

2.4.2 Dilution of Precision

This does not mean that GPS or cellular signal cannot be used at all, but that usability reduces near and inside buildings. The practical effect of the obstacle is to block signal in one or

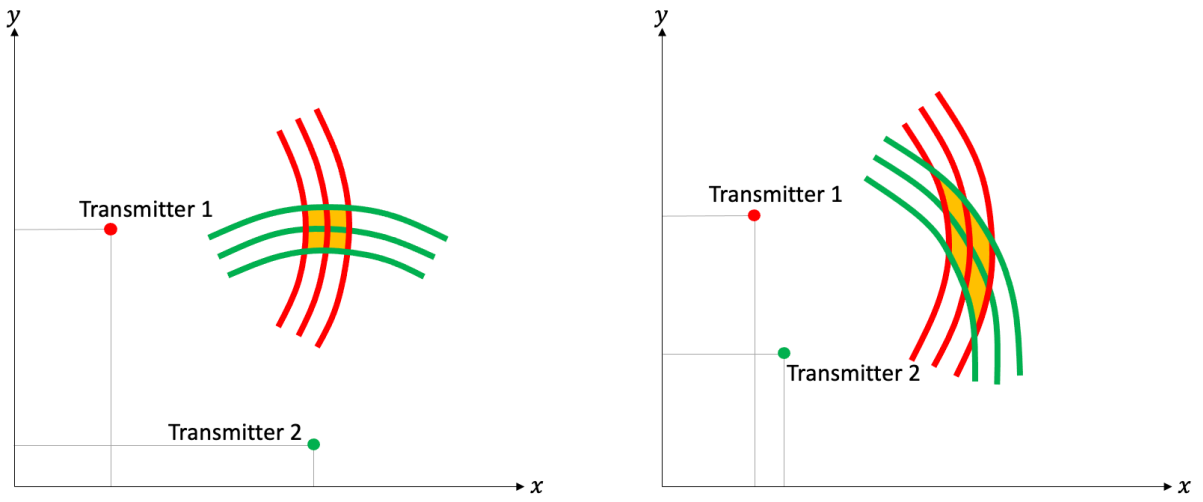


Figure 2.2 – Dilution of Precision

more directions, and cause shadowing in one or more directions [80]. The consequences on the localization equations depend very much on the technique in use, but in general this phenomenon causes what is called dilution of precision. This phenomenon can easily be understood with the geometric illustration in Figure 2.2. The position (x, y) of the UE is evaluated from its distance to each transmitter. As the measurement is noisy, each distance is a range, which spread depends on the noise figure. Suppose that the noise for the measurement to each transmitter is equivalent (which is the best case scenario). In the left part of the figure, the transmitters are far apart (in this case orthogonal to each other in relation to the UE). Therefore, the imprecision of the determination of the UE x component is the same as the imprecision of the determination of the y component. However, in the right half of the picture, the transmitters are closer to each other (particularly in the x axis). As a consequence, the uncertainty region is larger, especially in the y axis. The precision of the evaluation is diluted in that axis.

In a GPS measurement of course, the distances are pseudoranges, and the UE measures the difference of arrival between signals rather than the strict distance (thus drawing hyperbolas rather than circles). Such geometric figure also has dilution challenges on its own, but the principle of the dilution of precision remains the same. When one or more directions are blocked by obstacles, and when the only remaining known sources are grouped within a small angular region, the uncertainty of the computed location increases. This is true regardless of the number of available sources [81]. As the dilution is directly related to the standard deviation of the measurement error, it is possible, for a given position, to determine the dilution of precision in each direction [82], based on the number and directions of available sources. Different augmentation techniques were proposed to limit this issue. For example, Liu, Nath, and Govindan [83]

proposed using detection of the building and comparison with publicly available aerial imagery to revert to meter precision. Other authors propose to use the known degradation pattern of the signal [84] [85] to adjust the location calculation. Our measurements in chapter 5 show that accuracy can be maintained at a sub-meter level in many cases, even near buildings.

2.4.3 Limiting the dilution with dead reckoning and map magnets

Measuring the dilution of precision is useful, in particular for objects which trajectory leads to urban canyons [86]. In areas where precision may fall below a target threshold, the reliance on the computed location is limited, and additional parameters are used to determine the object trajectory. Two approaches are commonly used, either in isolation or in combination, to improve the location in such scenario:

— One approach is to compute series of location values. Each time, a system predicts what the next location should be, based on the current trajectory (*e.g.,* velocity) and the previous location values. Then, the next measured location is compared to the location prediction, and the final output is provided based on the weight assigned to each value (prediction vs. measurement). A classical technique to operate such evaluation is the use of a Kalman filter ([87], see also Section 1.5) or an alpha-beta filter. More recent techniques make use of machine learning to operate the prediction [88].

— Another approach is to rely on the object internal sensors to evaluate the displacement. This technique is more advanced than the mere trajectory estimation mentioned above and based, for example, on a speedometer readings. In the sensor case, multiple measurements are used to understand the movement (*e.g.,* gait or changes in the magnetic field [89]), and deduce the correlated change of location from one interval to the next. These measurements are the primary source of movement determination. Their output is then compared to the measured location. Here again, the final output is provided based on the weight assigned to each value (prediction vs. measurement). This augmentation is widely used for GPS, as will be detailed in Section 7.2.

These techniques are often complemented with the principle of magnets. Magnets are used to snap the display of the UE over specific areas. For example, when a device moves along a road, magnetism can be used to display the location on the road itself, even if the localization algorithm determines the location to be on the side of the road. The trajectory then appears to stick to the road object, even if the real computation would result in a zigzag shape. Quite clearly, this technique is limited to specific objects (like roads, train tracks, sidewalks, *etc.*)

In some cases, both velocity and other internal sensors are used to understand the movement. The technique by which a new position is estimated based on the evaluation of movement is in general called dead reckoning. The origins of the term are debated, but a common accepted

explanation is that the expression dates from Elizabethan times (1605-1615), where a log or other buoyant object (called the 'dead body'), attached to a rope, would be thrown in the water. After a while the displacement of the ship compared to the position of the object (expected to be fixed, at the scale of the measurement) would help assess the ship real trajectory.

In modern outdoor localization, it is common that fusion techniques would be used, combining GPS, cellular signal-based localization and dead reckoning to compute a location value with higher accuracy than each technique would offer individually [90]. The multiplicity of the sources is of course useful to remove the dependency to a single localization technique, thus allowing location to stay accurate even in places where the GPS signal is temporarily lost [91]. It is also useful to maintain the precision despite dilution factors. The effect of such fusion is that the location accuracy may start at high and satisfactory levels as the user stands in open space, with multiple GPS satellites in view. But the accuracy then stays satisfactory as the user approaches a building (even if only a few satellites in a narrow angled region of the sky are still in line of sight), then enters. It is only when the internal sensors become the sole source of location evaluation that precision starts to get lost, if there is not an additional reference point to compare against [92]. At that point, other technologies are then commonly used to deploy anchors with known positions and perform measurements (signal or distance) against them [93] [17]. These technologies either complement outdoor techniques like GPS [94], or replace them. This is the space where FTM becomes a technology of great interest.

2.5 Conclusion

This chapter brought several fundamental bricks to our study. On one hand, we saw how GPS performs localization, using both atomic and consumer-grade clocks, and also focusing solely on the end-device navigation case. By contrast, cellular technologies need the location of the end device, not only on the device itself for navigation, but also on the infrastructure for resource management and asset tracking.

This variety of use cases has brought forward multiple localization techniques, some using signals and distances, other using angles, some triggered by the end device, some triggered by the infrastructure. This learning will prove valuable for an evaluation of FTM. Most devices supporting both GPS and LTE tend to use each technique to complement the input of the other, often in combination with additional sources (like internal sensors). This is an important observation, because GPS focuses only on the end-device location case, and therefore fulfills the requirements for device privacy. LTE does not have this property, allowing unwanted location of the UE, without input from the user. On the end-device, the fusion of technologies achieves higher accuracy, and also limits the ability of an adversary to attack location through one of the techniques.

However, as the device enters buildings, zones of accuracy dilution appear. In the next chapter, we will see how indoor technologies can be added at this point, to extend accurate location based on technology fusion to places that GPS, or the cellular signal do not reach.

Chapter 3

Using Indoor Localization Solutions

3.1 Introduction

As the user moves deeper into the building and the internal sensors become the only source of information to perform localization, accuracy is lost and additional techniques need to be leveraged. Several approaches used outdoor can find an equivalent indoor. However, the environment itself, with strong obstacles (walls and others), and short distances to the client device, allow for many more techniques. This chapter will therefore first examine the different use cases for indoor location, as different techniques were developed with specific use cases in mind. We will then survey the various indoor localization techniques, and evaluate their likelihood of becoming prevalent to complement the outdoor techniques. We will focus more in depth on 802.11-based technologies, not only because of its long history, but also because of Wi-Fi omnipresence, that makes 802.11 an attractive vehicle for indoor location.

3.2 Location Use Cases

Indoor localization also diverges from outdoor localization in the use cases that are considered. Outside, the primary use case is a station needing to determine its own position. Therefore, a set of anchors (satellites, cellular towers, etc.) provide a structure from which the UE can make that determination. There may be a need for a central system to determine the location of a UE. One example is LTE/5G, with UE location determination made by the central system, to facilitate hand-off from one cell to another (or optimize communication parameters). Another example is asset tracking (fleet, etc.). However, in the vast majority of these use cases, the UE is still in charge of collecting localization element from the anchors, then communicate back information that allows the central system to display the UE location.

Indoor however, localization is a more varied landscape, with multiple use cases, some of which do not make sense outside.

3.2.1 Indoor Navigation

Indoor navigation is probably closest to the outdoor primary use case. A user visits a new venue, and attempts to find a particular point of interest (PoI). The indoor environment still presents two particularities that make this use case different from its outdoor equivalent:

1. Outdoor, there are landscape objects that can be used as magnets to the location display. For example, a UE moving along a road will cause most location display apps to snap the location to the road itself even if the user is in fact walking on the side of the road. Such magnets are useful in places where location accuracy is reduced. By contrast, indoor, such magnets are uncommon, except in particular settings examined later in this chapter.
2. Outdoor, the location accuracy may vary, but there is usually low pressure on the scale of the accuracy requirements for civilian use cases. When in a vehicle, the device needs to be positioned with an accuracy of a few meters. Even in magnet-free places (*e.g.,* parking lots), visual inspection is commonly sufficient to compensate for location inaccuracies. The user may notice that the reported location is wrong by, say, 5 meters, by visually comparing the location display and the real world. But at the scale of the outdoor environment, it is uncommon that such inaccuracy would have any critical effect on the navigation experience, if there is a user in control. When no users are in control (*e.g.,* self-driving cars) additional systems can take over to increase the accuracy (*e.g.,* cameras to detect lanes and lane separation marks). By contrast, the accuracy problem is a constant preoccupation indoor, because the scale changes with the use case. When a user is navigating a venue in search for a large PoI (*e.g.,* a store in a shopping mall), an inaccuracy of 5 meters is perfectly acceptable if it can be compensated by visual inspection. However, when the user is searching for a small PoI, then the accuracy needs change. For example, a user in a hardware store attempting to find the 4-cm nails in the "Screws and Nails" aisle will not be satisfied with a 5 meter inaccuracy. At that scale, 5 meters may mean the wrong aisle. Even in the correct aisle, 5 meter inaccuracy is not sufficient when the entire wall is covered with boxes of nails of various sizes. A 5-meter accuracy makes the navigation scheme useless in that case.

3.2.2 Proximity

In some scenarios, the requirement is to react on the proximity of the device to a PoI. One key characteristic is that the accurate location of the device is not necessarily a key factor to the success of the technique. For example, a system may need to detect if a sensor is near a particular object (*e.g,* a statue in a museum) to activate an audio-description. The exact distance and the orientation of the sensor to the statue do not matter, as long as it is within the expected range.

Here again, the scale is critical and variable. An accuracy to the scale of a room may be

sufficient for the term "proximity" to be valid in a museum where a single audio file is to be played to describe the object(s) of a particular area. An accuracy of one or two meters may be needed to determine if a nurse is servicing one or another patient in an ER dispatch area. An accuracy of a few centimeters may be required to determine if an object is on one side or the other of a door.

This variety of requirements under the same "proximity" umbrella leads to a multiplicity of solutions. Some of them are only loosely related to indoor location (*e.g.,* cameras, push-buttons) and will not be examined in details. Beyond the technological differences, the solutions relying on wireless signals can be classified based on the spectrum frequency they focus on:

— RF (Radio Frequency). The term is by itself vague, and includes multiple families of techniques that are studied below. Indoor, most of the solutions focus on the microwave range of the spectrum.

— Light (visible and infra-red). Although light is obviously also a radio signal (or, more correctly, all signals in the spectrum are electromagnetic in nature and use specific frequencies), the techniques used with light are fairly different from the ones used in the general RF family.

— Sound. This is of course also a wireless wave, but here again the detectors and properties are fairly specific. Sound includes audible and ultra-sound.

— Magnetic field. This family includes (in rare cases for proximity application) the earth magnetic field, and artificial source of magnetic fields.

3.2.3 Asset tracking

In this scenario, the location of an object is needed by an external system. The purpose may be to locate the object itself (*e.g.,* a laptop, and smartphone or tablet), or to use the object location as a proxy for another object location. This proxy function may be a property of the object, because it is attached (in a stable fashion) to the object to locate. For example, an RF tag is attached to a palette, and locating the tag is considered equivalent as locating the palette. The proxy function may also be a consequence of the way the object is used. For example, a barcode scanner may be located using RF signals at the time a worker presses a button to read a barcode. The location of the worker can be considered equivalent to the location of the scanner. Similarly, the location of the scanner can also be used as a proxy for the location of the object onto which the barcode is attached.

One key characteristic of asset tracking is that an individual object needs to be located. In other words, and continuing on the previous example, the goal is not to determine that there is "a" barcode scanner in location ABC, but that scanner 123 is in location ABC, and therefore worker DEF and scanned object GHI. This key component will have important consequences

for FTM.

Indoor asset tracking resembles its outdoor cousin. The object relies on signals from anchors (the eNBs in the outside case), and sends back to a backend system elements sufficient for the system to display the location of the object on the map. The location itself may be computed by the object itself. In other cases, the objects simply forwards the measurements it conducted, and a location engine computes the location. Because of the transmission (and possibly location computation) delays, the location is usually not displayable in real time (these systems commonly use the term nRTLS, for near-Real-Time Solution).

3.2.4 Traffic analytic

This scenario is close to the asset tracking case in concept. However, a major difference is that the near-Real Time location of individual and identified devices is usually not required (or wanted). The goal is to understand movement patterns, not to locate individual objects. For example, a store may want to understand how customers shop, to improve the store layout (*e.g.,* do customers buying fish also commonly go to the wine aisle? If so, is the path natural, for example because the wine aisle is between the fish aisle and the registers, or is-it complex, for example because wine and fish are in different corners of the store?). For example, an airport or a hotel may need to understand the length or waiting lines depending on the time of day, and the speed at which the lines move.

In all these cases, the identity of a particular tracked object (typically a smartphone used as a proxy for the location of the person) is not necessary to solve the use case. However, collecting useful statistics implies two key elements:

— a tracked object can be uniquely identified for a 'reasonable' amount of time (the value of reasonable' depends on the use case), so as to make useful conclusions on movement and position. This requirement means that the object needs to express an identifier, and that the identifier needs to be stable for some time.

— an individual object can be tracked for a 'reasonable' amount of time (the value of reasonable' depends on the use case), sufficient to determine a path and a trajectory. This requirement thus dictates that range measurement and data collection occur at a pace and for a duration compatible with the use case. For example, a smartphone sends location data at least once per 5-second interval, and for at least a minute with a given identity.

These two requirements have consequences on the collaboration structure between the tracked object and the localization infrastructure that will be examined later in this chapter. In effect, the object carried by the user is used as a proxy to the user, and the expectation is that the object is trackable. Many questions emerge on the benefit or the will of the user. These considerations directly weigh on the definition of a "modern" localization protocol. If it is solely intended for

the end device (blue dot), then it should operate with downstream-only traffic, making the end device silent (receive-only) and thus invisible for tracking actors. If it is intended for infrastructure tracking, then it should include an opt-in mechanism (no individual tracking by default, and no tracking before the user accepts the device to be tracked). If it is intended for analytic, then it should incorporate a mechanism to anonymize individual device data. These questions will be examined more in depth in chapter 4.

The information collected may in some cases have only long term value. In the above example, the store collects traffic information for a duration that may span over weeks or months. The data can then be divided into relevant time series (week-end vs. week-day traffic, peak hours, *etc.*), and conclusions are driven from the analysis of the time series. During data collection, a single time-slice has no value by itself, because it only provides anecdotal data. Multiple slices are needed until a pattern emerges. In other cases, the information collected has immediate value. In the above example, the hotel may deploy additional staff when the lines grow longer or move slower. These considerations also have important consequences in the design of a localization protocol, and will be examined more deeply in next chapter.

3.3 Localization Technologies

Most of the outdoor technologies described in the previous chapter rely on RF exchanges, and so does FTM. As such, we will only consider briefly the other indoor localization techniques, as they may be used as augmentation techniques, but are not the primary focus of this research. We will also ignore the technologies that do not attempt to locate a device (*e.g.,* radars using height to operate human counts, *etc.*)

3.3.1 Light-based technologies

Light is one form of electromagnetic signal. However, the technologies used with light-based systems have fairly specific properties that make them worthy of a special category. In particular, human buildings are built with materials that either let the light through or block it (curtains may be an hybrid category, but they are not of high interest for our purpose). Therefore, light-based technologies follow the building structure. They work in open spaces, but not from one room to the next through a wall. When going through a window or an open door, they only work in straight line of sight.

Infrared

Among the light-based technologies, Infra Red (IR) is probably the oldest. It was not engineered initially for location tracking, but for data communication. The emitter contains a diode that emits pulses of infrared light. The receiver incorporates a photo-diode, that receives the

pulses and converts them into a message (analog in the 1960s, digital today). A typical embodiment of this technique is the TV remote. The data rate for such transmission is low as the amount of data to transmit is expected to be limited in volume.

IR is sometimes used for indoor location. The emitter directivity and intensity can incorporated into the design. Narrow angle signals are only usable from within a small set of positions. Wide angle signals can be leveraged from multiple directions. Additionally, the receiver diode only reacts beyond a specific amount of received signals. These two properties can be used together to design a proximity system, where the receiver only reacts when within a certain range and angle of the emitter beam.

This apparatus supposes a receiver and a transmitter, but the human body also generates heat that radiates in the infrared spectrum. Thus other systems, like [95], use a passive infrared diode to detect changes in the infrared noise of a room, and deduce the proximity (and distance) of a human body.

The technology itself, with transmitters and receivers, can be used for indoor location in a more elaborate way. For example, Gorostiza [96] designed a system where a robot would transmit multi-directional IR beams. Multiple receivers were positioned around the robot, each reporting the pulses it received. By comparing the phase shifts between receivers, a distance difference was derived, and thus the position of the robot. This technique is similar in spirit to the Ultra-Wide Band (UWB) phase difference of arrival technique, and will be explained more in details in that section. Gorostiza observed an accuracy of 10 cm.

Such accuracy level is attractive. However, the apparatus is complex, requiring multiple sensors in each room, and also requiring reasonable LoS conditions from the robot to most sensors, which limits the span of the use case. For example, a human carrying such client device near the body would block about half of the sensors, making location determination almost impossible. For that reason, other systems use different approaches, like AoA with multiple IR receivers [97], at the cost of a lower accuracy (1 meter). These limitations, coupled with the observation that end-devices and venues commonly do not incorporate IR systems, makes that IR has seen limited adoption for indoor location.

Visible Light

Visible light communication (VLC) is an attractive technology indoor, because most rooms include artificial lighting. The early versions of the technologies relied on the same idea of pulses as for IR, where the light is switched on and off at fast pace (faster than the human retina can detect, thus providing a pulsing structure while the light seems to be constantly on for the consciousness of the human observer).

In a location context, a common use of VLC is illustrated in Figure 3.1 and relies on the idea that an open space would have multiple ceiling lights, at known locations. Each light emits its

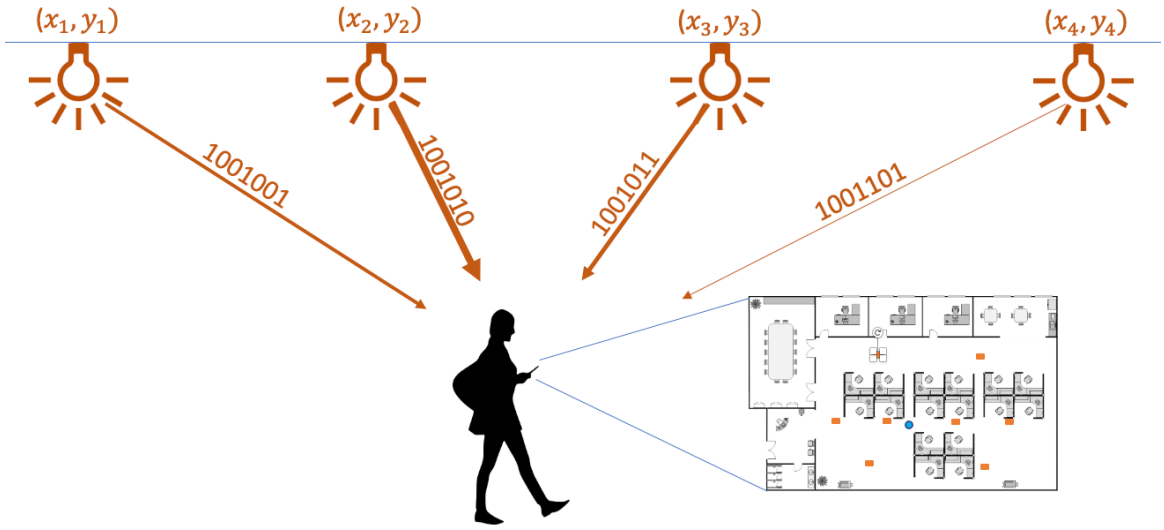


Figure 3.1 – Location with Visible Light Communication (VLC)

own data, including elements that uniquely identifies the emitting bulb. The receiver is mobile, and incorporates a photocell or a camera that registers the light pulses, decodes the identifiers, and also compares the intensity of the pulses received from each bulb.

The outcome of such comparison is a location system where the user is deemed closest to the location of the bulb which light was received at highest intensity. When historical values are kept, the variation of the intensity (light 1 intensity decreases, light 2 intensity increases) can be used to operate a geometric inference on the receiver position and thus increase the efficiency of the technique, providing close to 20 cm accuracy in some cases [98]. This method supposes that the transmitter intensity is known and can be controlled. Without this knowledge, the location determination is more complex, and becomes a search for an optimum solution within a space of possibilities. The computation cost is therefore higher, but an accuracy of 30 cm can still be achieved. [99] provides a good overview of the various possible techniques in that unknown power scenario.

VLC is also rapidly evolving. In 2018, the IEEE 802.11 working group created the 802.11bb task group, to design an amendment to 802.11 incorporating visible light technologies, bringing to VLC the notion of channel (based on the color of the signal) and modulations (like Orthogonal Frequency-Division Multiplexing, OFDM) that are well-known in 802.11. as such, it is very possible that common location techniques that are available for 802.11 (and detailed later in this chapter) will also apply one day to VLC. Until then, adoption is a key challenge. Certainly, there are lights in every room of every building. However, the cable leading to the lights are not data-bearing. Light circuits are also often serial (connecting multiple light bulbs with a single wire

pair), making communication individual lights challenging. Until PoE-to-light-bulb becomes a common practice, the VLC infrastructure is likely to stay uncommon. On the client side, communication is naturally only possible when the photocell or the camera are in LoS of the signal. This property may have some advantages, but it also has a major drawback: communication is impossible if the device is in a pocket, a bag, upside-down *etc.* Last, the downstream circuit is obvious, but the response back from the device is more difficult to design. Until these limitations have been overcome, VLC is likely to stay a technology that will complement another, instead of being the primary and sole option in a given venue.

3.3.2 Sound-based technologies

Sound-based technologies are attractive because of the speed of the wave. In dry air at 20° C, the sound wave travels at about 343 meters per second. By contrast, an electromagnetic signal (RF, light) travels at 299,702,547 meters per second in the same conditions. This difference can have dramatic effects on the accuracy of location, for any method that leverages time of flight and/or requires some synchronization between the anchor and the client clocks. With a signal traveling at the speed of sound, 1 microsecond drift between both clocks translates to an inserted inaccuracy of 0.343 millimeters (which is the distance traveled by the sound wave in 1 microsecond). By contrast, the same drift translates into an inserted inaccuracy 299.7 meters with RF signals traveling at light speed.

Ultrasound-based technologies

Ultrasounds are sound waves that are beyond the audible range (typically above 18 KHz). Their obvious advantage is that they are not detected by humans in the environment, thus limiting the associated problem of annoyance management. Ultrasounds can be used to carry data either by modulating the frequency or inserting a pulse structure. When the client device is the emitter, anchors can be deployed in the surrounding area, and receive a timestamped signal from each mobile unit. With the right density of sensors, and using a TDoA technique, Qi [100] obtained sub-millimeter accuracy.

One major difficulty is the requirement to deploy a multiplicity of sensors around the mobile objects. In Qi's experiments, each sensor had a Wi-Fi connection that was used to relay to a location server the measurements collected from each mobile station signals. It is clear that such setup does not scale easily. To overcome this problem, Park in [101] deployed the inverse logic. In a shipyard and an offshore oil platform, ultrasounds speakers were placed at strategic locations near metallic walls. These walls present the interesting property to propagate (at least partly) the ultrasounds (through resonance and reflection). The effect is not only a propagation effect, but also a distortion as the sound is captured along the wall farther away from the position of the transmitter. Each experimenter then carries a smartphone, which microphone receives the

signed signal from several ultrasound sources. By comparing the signal strength and frequency, the smartphone position could then be estimated, with an accuracy of 3.5 meters.

Park's experiment was facilitated by the structure of the walls in the particular environment where the apparatus was deployed. However, very few buildings on the planet have metallic walls, and the method is likely not easily transferable. Of course, Park's goal was not to experiment with ultrasound, but bring a localization system for health and safety purposes into a specific environment. Ultrasound happened to be a good solution there, but is likely not equivalently good in many buildings. Just like sound, ultrasound does not go very well through standard buildings plaster, wooden or brick walls, as most building walls are precisely designed to limit the transmission of sound from one room to the next. Therefore, with the need to equip each room with multiple anchors, scalability becomes an important difficulty, regardless of the direction of the transmission (anchors to device or device to anchors).

Ultrasound is still commonly used for proximity detection. For example, in enterprise buildings, many integrated teleconference units emit an ultrasound signal that is a device identifier. When a user sits in a meeting room near such a device, the teleconference application on the user mobile device detects the identifier and queries a database to find the IP address of the transmitting unit. Later, when the user decides to share the mobile device screen content onto the larger teleconference unit, the teleconference application on the user device merely establishes a screen mirroring session to the teleconference unit IP address.

This solution works because the teleconference unit is installed in the room (for teleconference purposes, *i.e.* with or without screen sharing) and connected to the local network. The location accuracy is reduced to "inside or outside the room". Thus, this system could not be simply extended to offer a full localization solution, until having multiple ultrasound units (within teleconference units or other connected objects) installed in each room will become common.

Audible sound-based technologies

Audible sounds could be used the same way as ultrasound, which the major difference that they are detectable by humans, and therefore present an annoyance factor that would be difficult to overcome if the sound bearing location data was transmitted alone. However, most systems leveraging this technology use a technique called watermarking [102] or steganography [103] (depending on the sources), where the location-bearing sound is mixed in a non-data bearing carrier (typically a musical piece). Thus the human hears the music, but does not detect the hidden signal. An application on the receiver parses the audio input in search for the hidden signals, which are commonly a source identifier and a timestamp. Then, TDoA techniques are used to compute the receiver position.

This approach is promising. However, it presents the limitation of relying on audible signals, with two obvious detrimental consequences:

— Audible noise is a factor. For example, in noisy environments (either because of the presence of objects, like machines, that also emit audible sounds, or because of the chatter of a crowd), the individual signature of each speaker becomes difficult to read. Increasing the speaker sound level to acquire a high signal to noise ratio (SNR) brings back the annoyance factor.

— The solution does not work in quiet environments. Conceptually, the hidden signal can be embedded into a low-volume white noise at the edge of the detection capability of a human ear. However, the introduction of a human in the scene also introduces additional noise that brings back the limitation above.

For these reasons, sound-based techniques, although attractive from a technical standpoint, have limited appeal from a use-case standpoint. They may be usable in some environments, even in combination with other technologies (for example with VLC, in [104]), but they are unlikely to become a universal localization technique (applicable to almost any indoor environment).

3.3.3 RF-Based Technologies

Just like radio technologies are used outside for location determination, they form an important set of options for the indoor case. One difficulty however is about emission control. In most countries, the spectrum is considered a resource shared by all, and individual band usage is therefore carefully regulated. The license to operate a system in a regulated frequency range comes at a cost, but also with strict requirements about power, coverage, signal structure, and coexistence. For this reason, most indoor systems, operated by networking professionals but not RF or regulation professionals, tend to rely on unlicensed bands. The signal structure sent in these bands (often labelled Industrial, Scientific and Medical, ISM) is still closely regulated. However, an operator, installing a system certified to be allowed in such band, does not have to pay a license and merely has to comply to simple deployment requirements (*e.g.*, spurious emissions toward the outside *etc.*)

Localization is just one of the use cases where radio technologies are useful indoor. Therefore, the success of an indoor radio-based localization technology also partially depends on its ability to fulfill other use cases.

Ultra Wide Band

Similar to the other technologies we will examine in this section, Ultra-Wide Band (UWB) has a complex history. The term itself merely designates a signal that is sent over a wide bandwidth (500 MHz or larger, or a signal that uses a fractional bandwidth greater than 20% of the arithmetic center frequency). Such wide signal can be highly disruptive to other systems, and consume a large segment of the spectrum. To avoid this issue, the signal must be sent at low

power (*e.g.,* -41.3 dBm in the FCC and ETSI domains), thus near what other radio technologies would consider as the noise level (allowing UWB to coexist with these technologies in the same band). Yet the short (ns scale) pulse structure makes that UWB receivers can detect a signal while minimizing the effect on other radio technologies. In addition to power limitations, several regulatory bodies limit the amount of energy that can be sent per unit of time. The spectrum allocation depends on the regulatory domain, but 3 to 10 GHz is a good representation of the range focus. There are multiple standards that describe "ultra wide band transmissions", but the IEEE 802.15.4a and 802.15.4z Standards are of particular interest in this thesis, because they focused heavily in the ranging, and therefore localization, use case.

802.15.4 describes two primary ranging modes that are relevant to localization. A first mode is an interesting variation of the generic ToF scheme described in Section 1.3.2. This mode exists in two flavors:

— One variation, called Two-Way Ranging (TWR) in 802.15.4a and renamed Single-Sided Two Way Ranging (SS-TWR) in 802.15.4z, a device (A) sends at time t_1 a range request to another device (B). The devices clocks are not synchronized, but have agreed on a reply delay (t_{replyB}). B receives the request at time t_2, waits for the interval t_{replyB}, then sends a reply at time t_3 that is received by A at time t_4. The mode expects that the travel time is the same in both directions, and that the imperfection of the measurement of the departure and arrival times are the same on both sides. Therefore, A can measure the round trip time ($rtt = t_4 - t_1$) and estimate that the time of flight is roughly $(rtt - t_{replyB})/2$. This mode is of course imperfect in many ways. One side may have better precision in estimating the arrival and departure times, thus causing the imperfect side's error to contaminate the precise size's evaluation. This limitation can be reduced by a performance certification program. More importantly, the clocks are not synchronized, causing a different estimation of t_{replyB} on both sides. This issue is problematic, because t_{replyB} is evaluated by B but used by A, while A has no information on the real duration of that interval in B.

— Another variation attempts to limit the clock drift issue, and is called Symmetric Double-Sided Two-Way Ranging (SDS-TWR) in 802.15.4a and Double-Sided Two-Way Ranging (DS-TWR) in 802.15.4z. This mode starts like SS-TWR. However, after receiving B's response at t_4, A wait for an agreed upon interval t_{replyA} (that can be of the same theoretical duration as t_{replyB}), then replies at t_5 with a frame received by B at t_6. At this point, both sides have an rtt value. B also knows t_{replyB} and t_{replyA}, thus the error of t_{replyA} can be partially compensated in B (roughly dividing the effect of the error by two. Additionally, if all timers are passed to an external system, the values of $(t_2 - t_1)$, $(t_3 - t_2)$, $(t_4 - t_3)$, $(t_5 - t_4)$ and $(t_6 - t_5)$ can be compared, the drifts evaluated, leaving the rtt value to an evaluation that largely compensates for the clock drifts. The standard also dictates a crystal precision of 80 PPM max, to bound the error further.

When using TWR, the location is found using at least the three-sphere method described in Section 1.3.3, or the matrix method when more anchors are available. A second ranging mode relies on TDoA calculations. 802.15.4a describes two modes. In *Mode 2*, the mobile node sends a single broadcast message (called a blink). All detecting anchors note the time of arrival, and TDoA is used to compute the device location as described in Section 1.4.1. Just like in the generic form, this mode supposes that the anchors synchronize their clocks. This is usually achieved by designing a primary anchor, and secondary anchors. The primary sends a synchronization frame (usually called sync) at intervals, with its timestamp, and the secondary anchors, knowing their distance to the primary anchor, adjust their clock accordingly. In real field deployments, the secondary anchors merely record the ToA of the sync frame, and forward the information to a location engine. The engine is then in charge of incorporating the time differences between anchors (and their respective distances) when computing the TDoA location. As will be detailed in next chapter, our experiments show that, with an 80 PPM clock, the sync messages need to be sent every 200 ms to maintain high accuracy. In *Mode 1*, the opposite structure is implemented (and this mode is often called Reverse-TDoA, or Downlink TDoA). The anchors still have their clock synchronized, and sent at the exact same time a blink (practically, we have observed implementations where the messages are not sent at the exact same time, but they all carry the primary anchor expected time). The mobile node receives these messages and computes its location based on the TDoA between the different messages.

The computation of the location result can theoretically be achieved using the method of simultaneous equations. However, in real life, the measurements are noisy and the solution is inconsistent. The evaluation then becomes difficult, as the equations are nonlinear. One common approach is to attempt an initial (possibly random) solution, and use then the first Taylor series at the initial location estimation, then solve the equations by iterations [105]. Multiple variations of this method exist. Some introduce auxiliary variables and thus transform the equations into a weighted least square (WLS) problem [106], others into a convex semidefinite programming structure [107]. In both cases, the system is then solved attractively with the Newton approach. Other methods use WLS, but augment it with additional techniques to increase accuracy and reduce the computation cost [108]. Others avoid the iterative methods, and use a closed form approach [109].

In all cases, and because of its large bandwidth, UWB offers high location accuracy (the relationship between these two variables is detailed in Section 4.4.1), achieving 10 cm in LoS conditions. However, because of its low power, the range is rather short. A single plaster wall may be sufficient to bring the signal amplitude below detectability level. Additionally, the standards leave important design challenges unsolved (that will be examined in depth in next chapter). UWB adoption also suffers from the chicken-and-the-egg problem. Adoption exists in verticals where a full UWB solution is deployed with location as the primary use case. However, for a

long time, adoption in general public devices (*e.g.,* smartphones) has been minimal, because there was no infrastructure to range from (and vice versa). With no usage outside of ranging, it was not very tempting to install a chip that would almost never be used. This trend is changing with large vendors adopting the technology, and UWB may follow the same path as BLE.

3.3.4 BLE

Bluetooth Low Energy (BLE), expanding from Nokia's Wibree specification, was introduced in the Bluetooth specification 4.0 in 2010. Under the influence of large end-device vendors, the technology adoption skyrocketed very fast, and is today present on most smartphones. Contrary to its parent Standard (Bluetooth, defined in IEEE 802.15.1), BLE is not based on the assumption of a primary system connecting and controlling secondary systems for data transfer. Instead, units can send broadcast messages (beacons) with a flexible data structure that include a unique identifier for the beacon source. The beacons are sent typically at low power. Although the Bluetooth specification (that includes BLE) v5.1 allows a power of up to 20 dBm, most beacons are sent between -20 dBm and -10 dBm.

The effect of such low power envelope is that BLE range is limited. A receiver should be able to demodulate successfully a BLE signal at -82 dBm, allowing thus a LoS range of about 50 meters. In practice, the RSSI signal curve is descending log in structure as mentioned in chapter 1. Therefore, the degradation of the signal is easy to map into a distance at close range, but the measure becomes uncertain at longer ranges.

As a consequence, many BLE systems rely on a quasi-proximity method, that maps RSSI values into regions (*e.g.,* immediate proximity, near proximity and far) that correspond to distance approximations (*e.g.,* less than 0.6 meter, 0.6 to 2.5 meters, more than 2.5 meters. The model then relies on the known location of the beacon sources to establish a proximity value. When more than one beacons are detected, classical trilateration methods can be used to deduce a likely location region. This technique offers only limited accuracy, because the RSSI difference between the beacons that are in the "far" region is small. Some authors attempt to increase this accuracy. For example, Rudic [110] propose an iterative search using a Markov model to reduce the location uncertainty. Urano [111] relies instead on a LSTM to keep track of the device movement and model the best likely trajectory. In most implementations however, BLE is either use as a proximity system (with the meaning expressed in Section 3.2.2), or uses the FPSL equation and lateration to estimate a distance, assuming LoS.

In recent iterations, the BLE specification also allows for AoA measurements, intended to complement the other techniques [112]. Although promising in theory, this technique is challenged in practice. In most of the systems we have tested, the angle can be determined when the beacon source is near, but the accuracy of the determination diminishes as the distance increases (because the signal is very low), making the system only imperfectly complementing

the signal-strength-based techniques.

Therefore, it seems that BLE may be an interesting technique for close range measurements, but may have limited appeal at larger scales.

3.3.5 RFIDs

Radio Frequency Identifiers (RFID) are worth mentioning, primarily because the term designates two different types of objects:

— Passive RFID tags include a coil and a small controller. The system then relies on a NFC technology, where the reader emits and EMF signal that activates the coil, and which back scatter radiation is detected by the reader. This system provides unique identification with a proximity logic. It is widely used in Retail for theft prevention. However, its utilization for indoor localization is limited.

— Active RFID tags are in fact usually 802.11 devices with a battery and a Tx-only radio. They can be configured to send a signal at regular intervals on specific channels, with a specific tag identifier. They are then located using the applicable standard 802.11 techniques described in next section.

There are also hybrid tags that carry a battery that is activated only in specific conditions. However, RFID itself is not a localization technique, although the literature tends to list it as its own category [20]. Instead, it is a type of method to identify an object, using standard radio technologies for the localization part.

3.3.6 The Special Case of 802.11

802.11 has a special place for indoor location, primarily because it has the longest history among the main indoor localization technologies. 802.11 was ratified in 1997, and there were location solutions based on 802.11 as early as 2001 (*e.g.,* Airespace).

Infrastructure-based Localization

The early location solutions were solely infrastructure-based, focusing on the asset-tracking use case described in Section 3.2.3, and used three basic principles:

1. Because of the unknown geometry of an individual 802.11 cell coverage (called Basic Service Set, BSS), client devices tended to often send broadcast discovery messages (probe requests). These messages could be captured by one or more access point (AP) on the target channel. As the client station (STA) would repeat this message on all available channels, all APs in range would likely receive these messages.

2. The probe requests were by design sent at the lowest possible data rate. The intent was to maximize the range at which an AP could be detected, but the effect was also to maximize the range at which the STA could be detected.

3. 802.11 STAs were following the IEEE 802-1990 Standard (clause 5.2) for Universal Local Area Network (LAN) MAC addresses [113], by which each STA would use as a source address a globally unique identifier, thus setting its second bit of the first 48-bit MAC address to 0, letting the IEEE assign to the 802.11 card vendor the first half of the address, and using the second half as a unique identifier for that card within the cards manufactured by that vendor. The intent was to avoid collision, and thus ensure that, if a STA would be moved from one network to another, anywhere on the planet, there would be no chance that another STA would use the same identifier (thus causing collisions). However, this principle produced exactly the intended effect, but with different intents. In particular, APs receiving the probe requests would uniquely identify the querying STA, allowing for tracking of individual STAs.

Multiple localization techniques for this use-case were proposed. A cell-of-Origin method, similar to that of Cellular location (see Section 2.3.2) offered a mere "presence" information about a STA, attaching the client to that AP to which the client associated. This mode provided of course very low accuracy (30 to 40 meters), as the radius of a cell indoor could be in the range of 15 to 20 meters. Additionally, the STA might not associate to the physically closest AP. Most clients would associate to the AP with the strongest signal, but walls reduce the relationship between signal strength and strict distance.

Other techniques attempted to pinpoint the STA location, and also relied primarily on RSSI measurement, as detailed in Section 1.3.1. However, the implementers realized early that the FPSL Figure (1.2) would fail indoor. As detailed in 2.3.2, the Fresnel zone effect can be determined only if the distance between points is large compared to the zone to compute (because of the binomial approximation for the square root), making it impractical indoor. Additionally, the effect of multipath was much more pronounced indoor than outdoor, overshadowing the Fresnel effect. The empirical models, like Egli's in Equation (2.10) were attempted, but proved impractical, because there are too many variations from one building to the next, as detailed in Section 1.3.2. Therefore, designers resorted to diverging from the FPLS equation and creating a specific equation for indoor propagation and RSSI attenuation. One approach was therefore to create a new path loss (PL) equation, in the form:

$$PL = PL_{1\_meter} + 10log(d^n) + s \tag{3.1}$$

where PL_{1_meter} is a reference path loss in dB for the desired channel, when the receiver is one meter away from the receiver, d is the distance between the transmitter and the receiver

antennas in meters, n represents a path loss exponent and s is the standard deviation associated with the degree of shadow fading in the particular environment (also in dB).

In this system, each AP product must be modeled, and a PL_{1_meter} value must be derived. However, given that the regulatory space limits the AP total output power (ERP detailed in Section 1.3.1, and represented for 802.11 by the Effective Isotropic Radiated Power, EIRP, as it establishes the comparison by using an isotropic antenna as the reference), many vendors used a fixed value for all APs, *e.g.,* -46 dB at 1 meter. In this equation, the main factor is thus n, that defines the rate at which the path loss increases with distance. The rate depends of course on the obstacles in a particular location. Therefore, the way to use the equation is to first determine the environment where the AP is deployed, and classify its type, for example Open Space, Office with Cubicles, Office with drywall, *etc.* For each type of environment, a template value for n would be provided (for example 3.2 for Open Space, 3.6 for cubicles, 4.0 for drywall *etc.*). The standard deviation for shadow fading s would then be used as a small correction factor.

One interesting aspect of this approach is that the equation does not attempt to be accurate for a given AP in a given location. Instead, it defines an average wall-density with distance. In other words, given that the AP is deployed in an environment with drywall, the equation postulates the average density of drywall that the signal might encounter when expanding away from the transmitter, and would thus output a likely average RSSI at that distance.

Quite obviously, the validity of the model depends on the default parameters, but also on the building structure (size of the rooms, consistency of the wall density and distance across the entire floor, wall composition, *etc.*) Because this model is highly prone to errors, the operator also has the possibility to perform a "calibration", where a user would first position the APs on a map in a location engine, then would position a STA at multiple locations on the floor, would click the location of the STA on the map, then let the system collect signals from the STA and re-compute better n and s values for that floor.

Because the method is inaccurate in nature, the author of this thesis suggested a simplified approach (Yagna project 2016), where, after APs were positioned on a map, each pair of APs detecting each other would exchange messages and derive for the floor a mean path loss:

$$PL = a.ln(d) + b \tag{3.2}$$

where a and b are coefficients derived from these multiple exchanges. Depending on the wall density between each pair, different coefficients are dynamically learned. Not surprisingly, this simplified equation offered better accuracy than Figure (3.1), in particular because the loss could be localized (path loss between APs A and B has different coefficient than between APs A and C, and coefficients are adapted stepwise as a STA moves from B to C), and because the -46 dB assumption was most of the time highly unreliable. The outcome of method (3.1) is a ranging accuracy of 10 to 15 meters, and 5 to 10 meters with method (3.2). Then, multipshere

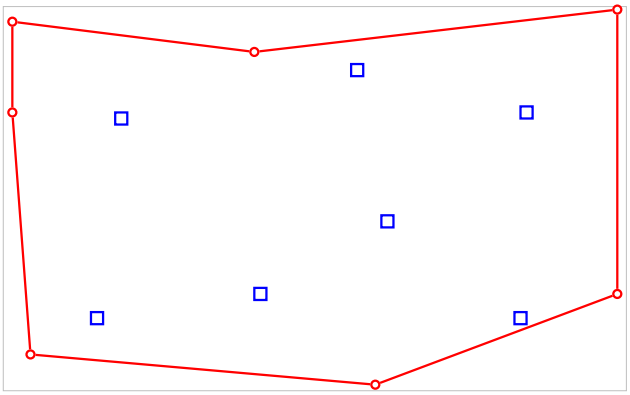


Figure 3.2 – Location is optimal within the convex hull formed by the APs (red circles) at the edge of the floor (gray rectangle). These APs may be in monitor mode, while other APs (blue square) provide data service.

techniques (Section 1.3.4) are used to compute the location, reaching 7 to 10 meter, and room-level, accuracy. This allowed an operator to know in which room an asset would be located, then visually find the asset when getting within a few meters of the object.

However, because the method relies on the sphere intersection method,one key consideration is that at least 3 APs should receive the probe requests for a solution to be found in 2D. Additionally, if all the APs are in the same direction (from the perspective of the STA), the issue of dilution of precision detailed in Section 2.4.2 cause the location error to increase dramatically. Therefore, most system would recommend to deploy APs that would be only in monitor mode (receiving the probes, but not transmitting any frames themselves), thus multiplying the number of APs on each channel (without increase the RF overhead on the channel), and also to ensure that APs would be deployed at the perimeter of the facility where location was expected, forming a convex hull around the location area, as represented in Figure 3.2.

One major limitation of this probe and RSSI-based approach is that it relies on the STA probe requests. In the early 2000s, STAs were primarily laptops, and were probing aggressively so as to maintain the best possible connection. As the technology entered the 2010 decade, the roaming algorithms evolved and the smartphone boom brought many STA vendors to reduce the probing rate. This evolution caused a major concern for the asset tracking technique, because a STA would probe upon startup, would join the best AP, then would stop probing until it would reach the edge of the cell. In between, the STA became invisible.

Thus augmentation techniques were found, where the active AP to which the STA would be associated would send ranging updates. In this mode, the AP would receive unicast frames from the STA (in a normal data communication structure), and would evaluate the RSSI for these frames, assuming a constant location if the RSSI was within the same range (*e.g.,* $\pm 3dB$) as the RSSI of the last probe. Quite naturally, the system could only assume a distance change if the

RSSI diverged, and had not real mechanism to assess the direction of the movement, making the augmentation imperfect. Additionally, the method also assumed that the power at which unicast frames are sent is the same as the power at which probe requests are sent. This assumption was true in most cases in the 2000s, but stopped being true when STAs vendors became more efficient at controlling transmission power, and when amendments to the IEEE 802.11 Standard (*e.g.,* 802.11h in 2003, 802.11ax in 2013) allowed the STA to dynamically reduce its power for multiple reasons.

These challenges limit the efficiency of asset tracking based on strict RSSI techniques. The system is however still widely in use. In an effort to increase the efficiency of the active AP range evaluation, some vendors have implemented AoA techniques. In this approach, a ring of up to 32 patch antennas is installed around each AP, and use the method detailed in Section 1.4.2 to evaluate the phase difference on each antenna, and therefore the direction of the STA that is sending the frame. This method improves the range evaluation, but also suffers from practical issues:

— The setup is prone to installation errors. The system relies on the angles to determine the direction of the STA. This requirement implies that the installer carefully documented the AP position, but also the ring orientation. Any error of a few degrees, in any direction, would output a correct direction in relation to the antennas, but incorrect in relation to the building or the other APs, causing the localization result to be unusable.

— Multipath makes the determination challenging, causing the primary component to appear at different directions, depending on the reflections or obstacles between the STA and the system. This issue is particularly present when the STA is moving (which is precisely the case where this augmentation method is needed).

Adding a ring around APs also comes at a cost. Therefore, the AoA augmentation technique is used, but has limited adoption.

To increase the difficulty, in the 2010s, STA vendors started realizing that highly mobile devices (smartphones, tablets, laptops) were also personal devices (*i.e.,* a single primary user), which meant that traffic that could be correlated with a personal device could also mechanically be associated with the primary user of that device. In this context, the location of the device, based on its MAC address, is equivalent to the location of the primary user, making individual asset tracking the equivalent to individual user tracking. To limit this exposure, many STA vendors reviewed the IEEE 802-1990 Standard guidance around the MAC address, and realized the the Globally Unique Identifier was nowhere a requirement, just one of two options, and the best choice to avoid MAC address collisions. But when a STA is sending probe requests, the risk associated to collisions is null (a probe request from a given address is in fact sourced from two different machines, but both are looking for the same kind of information: the details of the 802.11 networks available on this particular channel).

Therefore, STA vendors started using Locally Administered Addresses instead, where the second bit of the first octet is set to 1, and all other bits of lower weight are set randomly. The effect is that 2 subsequent probe requests from a given STA, on the same or different channels, are sourced from different MAC addresses. This change of behavior makes asset tracking very difficult. The STA would use a stable (locally administered or globally unique) MAC address when exchanging frames with the AP it is associated to, but these unicast frames are merely an augmentation technique to improve the accuracy of the main method, that relies on the broadcast probe requests. When the source of these requests cannot be traced back to a single STA anymore (and given the inaccuracy of the range derived from the RSSI), individual STA localization becomes challenged.

In many cases, this behavior is configurable. This is because asset tracking is not necessarily about unwanted privacy violation, but can be (in enterprise settings) about providing a better IT support service, as detailed in Section 3.2.3, and also tracking objects (palettes *etc.*) to which an 802.11 card is attached. Therefore, when the STA is an enterprise asset with a user interface, it is common that Randomized and Changing MAC addresses (RCM) can be enabled or disabled, either locally or through a Mobile Device Management (MDM) platform.

The RCM scheme also breaks the conditions for traffic analytic defined in Section 3.1.4, leaving the infrastructure-based location techniques with more practical challenges than solutions.

STA-based Localization

Infrastructure-based solutions also do not directly provide location information to the STA. When an infrastructure-based location solution is deployed, it is common that the solution vendor would make available a specific app that can be installed on the client. The app would use an authenticated connection to query the location server for the device location. The limitations of this system are obvious. The need to deploy a special app limits the adoption (the user would need a specific map for each location server, therefore each venue group), and also supposes that the device is associated to the 802.1 network, making the solution of limited interest for most public venues.

A STA can also use RSSI techniques to derive a distance from an AP message. This direction is facilitated by the fact that APs broadcast beacons at short intervals (102 ms by default). The distance is noisy, just like for the AP evaluation of the client distance. Additionally, the RSSI is just a number that translates the ability of the receiving system to convert a signal into meaningful binary values. Although the IEEE 802.11 Standard extensively uses RSSI, it is clear to all actors that the RSSI scale is only meaningful locally. For example, some systems may implement RSSI in a range from -127 to +127 in 254 increments of 1 dBm each, others may implement a scale from -111 to +111 with 74 increments of 3 dBm *etc.* Therefore, a report of "-53 dBm RSSI" from one STA may have a different meaning for another STA. However, because

the RSSI value has been exchanged between APs and STAs since amendments like 802.11k-2008 and 802.11v-2011, there has been a pressure on implementers to report values that were compatible with other actors. The scale is therefore only valid locally in theory, but practically most STAs tend to report an RSSI in the same range for a given signal level. Another issue remains, that the STA also has no idea on where the AP is, as the AP does not communicate its location in the beacon. This difficulty, coupled with the uncertainty of the RSSI scale, makes that determining the distance and location from the RSSI alone is of little use. However, the loss of accuracy of cellular or GPS-based techniques examined in Section 2.4.2 brought the need for an indoor augmentation technique. As we saw earlier in this chapter, UWB was not a great alternative because of its limited adoption. BLE emerged in the 2010 decade, but its limited range, and the need to learn the location of BLE beacon tags that could change their payload and identifier at any time made the solution a mild success. By contrast, 802.11 networks are pervasively present all around the planet. APs are often at fixed locations, making 802.11 an attractive augmentation solution.

Therefore, multiple organisations started establishing maps of popular indoor venues where 802.11 networks are deployed. Zhou in [114] provides a good summary of the various attempts. With these methods, an observer walks a venue, and clicks on a map of the venue the current location. In many cases, the exact location is not well-known, so these techniques use the concept of landmark, *i.e.,* objects that can be identified in the venue and on the map (*e.g.,* a bench, the entrance to a store, *etc.*) The observer the clicks the STA location when near these objects, then walks at steady pace and in a straight line toward the next landmark, repeating the procedure there. In between, the STA collects the signal from all APs in range. Each AP expresses its MAC address for a particular network in a field called the Basic Service Set identifier (BSSID). Contrary to the STAs, the APs typically do not use RCM for that identifier. It is therefore reliable to express a unique AP identity.

More elaborate systems use the dead reckoning technique described in Section 2.4.3 to evaluate the device movements between landmarks. However, the distance between landmarks has then to be reduced, as dead reckoning diverges in the absence of other references and when the movements are complex. The outcome of such mapping technique is a representation of the position of each AP in a venue. Figure 3.3 shows the output of such mapping taken by the author for a particular AP. The circles represent the author's detected path by the internal sensors in a smartphone, and the color code express the AP signal level, and thus the yellow square indicates the likely AP location. The right-bending grey rectangle represents the outline of the tested building.

The technique is also called crowd-sourced because multiple devices can contribute to enhancing the map. Once a base map with rough AP positions exists, when a user next enters the venue, the localization system identifies the likely location and collects data from the de-

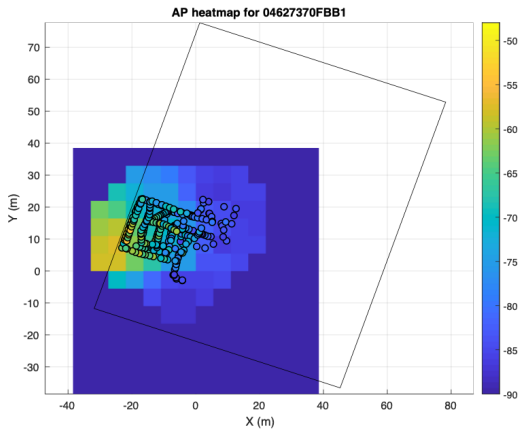


Figure 3.3 – An AP position and coverage area detected with crowd-sourcing technique. The building floor is a grey rectangle, the points where the phone location and RF data were collected are marked by circles.

vice (GPS, cellular, dead reckoning and other sources) and uses the information to refine the computed location of the detected APs.

The method is efficient, but also suffers from many limitations. Quite obviously, it only works in venues that have been mapped. With close to half a billion Wi-Fi hotspots on the planet, it is clear that many venues are left without crowd-sourced maps. Additionally, there needs to be a communication structure between the venue owners and the STA map source provider. The venue owner has the facility map, but the STA ma source provider has the source code of the STA navigation map. This communication is often manual, leaving many navigation apps with obsolete indoor maps. This problem also affects the APs themselves. The lifecycle of an indoor AP is typically inscribed between 5 and 7 years, after which the AP is updated to a newer protocol. The old AP is either decommissioned, or moved to another location. The result is that venues need to be re-mapped (with a new seed map and AP position) often. Without frequent updates, the navigation app fails to recognize that the user is on a floor where no APs are recognized. Worse, the system may also project the STA location near the position of the strongest detected BSSID, thus causing the wrong location issue mentioned in the introduction of this thesis.

3.4 Conclusion

At the end of this chapter, it is now clear that outdoor techniques cannot simply be brought alone to the indoor environment. Inside buildings, the need for location can overlap with the use cases that outdoor techniques address (and they can therefore be used directly when their signal reaches inside), but other use cases appear that require new solutions. Some techniques,

like light or sound, present specific attractive properties, but indoor deployment remains a challenge. Other radio technologies, like BLE or UWB, are also very interesting, but suffer from limited adoption. Support for these technologies may increase in the upcoming years. In between, 802.11 is present all over the planet, and offers multiple techniques for location to be provided to the end device or to the infrastructure. However, the various techniques were developed with only one side in mind (the client or the infrastructure), and the changes in RF behavior as the client population moved from early laptops to smartphones brought new challenges for these techniques efficiency. We are left in a state where 802.11 is still the best candidate for indoor location, but the legacy techniques have found their limitations In this context, FTM becomes a natural candidate to fill the gaps and help solve indoor location for good. In the next chapter, we will examine FTM more in details, and see if its initial design fulfill these promises.

Chapter 4

An Evaluation of FTM Performances and Design

4.1 Introduction

At the end of last chapter, we concluded that 802.11 was probably the best candidate to extend localization to the indoor environment. The technology is widely deployed all over the planet, and has implemented localization technologies for more than 20 years. However, we also saw that the techniques have aged, faced with the challenges of new behaviors in client chipsets, both for improved radio efficiency, but also because of privacy concerns.

FTM was developed in this context. In this chapter, we will therefore examine FTM more in details. We will first look at the circumstances where FTM was designed. We will then undergo a critical review of the design, to examine if FTM fulfills the requirements of a modern indoor localization technology. We will then attempt to implement FTM on an AP and a client chipset, to evaluate the challenges that need to be overcome, and validate the accuracy that can be achieved with this technology.

4.2 FTM Principles

Fine Timing Measurement (FTM) was designed in 2014 and 2015, and inserted in the 2016 revision of the IEEE 802.11 Standard. The idea was not entirely created from scratch.

4.2.1 Timing Measurement Procedure

In 2011, the IEEE 802.1 (802 Technologies architecture) working group published a series of standards for Time-Sensitive networking (TSN) applications, which allowed time-sensitive data flows to be forwarded reliably on 802 networks. Among the various provisions to allow this use case, the different nodes along the path would need to align their clock. In order to apply this possibility to 802.11 networks, the 802.11v-2011 amendment, integrated into the 2012 revision of the 802.11 Standard, introduced a method called Timing Measurement (TM). The spirit of TM is as follows:

1. An 802.11 client station (STA) would be associated to an access point.
2. At some point in time, the STA would identify some traffic that would be delay or jitter-sensitive, require precise packet scheduling and therefore time alignment with a server in the network (and as a way of consequence, with the intermediate networking nodes to that server).
3. The STA would expect that the nodes along the wired side of the network already implement time synchronization, for example through triggers from the server. However, the last hop (AP to STA) is under the control of the STA. Thus the STA would then send to the AP a request to initiate an ongoing Timing Measurement procedure (with a trigger value set to 1). The request takes the form of a specific Action frame, acknowledged by the AP.
4. The AP then starts sending series of action frames that follow the classical ToF procedure seen in section 1.3.2. The first frame is sent from the AP at time t_1 and received by the STA at time t_2. The STA sends an acknowledgement frame at time t_3, that is received by the AP at time t_4. In the subsequent frame, the AP adds the (t_1, t_4) values from the previous exchange. The AP can also include an additional field, that typically would include the time on the AP, synchronized with a reference time source.
5. the STA computes its time offset O from the AP with a simple modification of Equation (1.6), that removes the c component and searches for the time difference instead of the distance:
   $O = [(t_2 - t_1) - (t_4 - t_3)]/2$. By comparing the intervals, this equation also eliminates the synchronization issue detailed in section 1.3.2.
6. With this offset and the time reference sent by the AP, the STA can align its own clock with the reference clock in use by the networking devices and the server.
7. The exchange repeats until the STA sends a new Timing Measurement Request action frame, with a trigger set to 0.

The goal of the TM procedure was therefore for a STA to keep its clock aligned with the AP it was associated to, using the ToF value to compute the time offset. This logic has important consequences for the design analysis of FTM. Indeed, in 2014, the idea came to the mind of many 802.11 designers that integrating the speed of light into the equation would allow the STA to know its distance to the AP, and that therefore the process could also be used for localization purposes.

4.2.2 Fine Timing Measurement Procedure

One key limitation of TM is that the STA would only exchange with the AP it is associated to (which is a logical consequence of the TM goal). But for localization purposes, the STA would

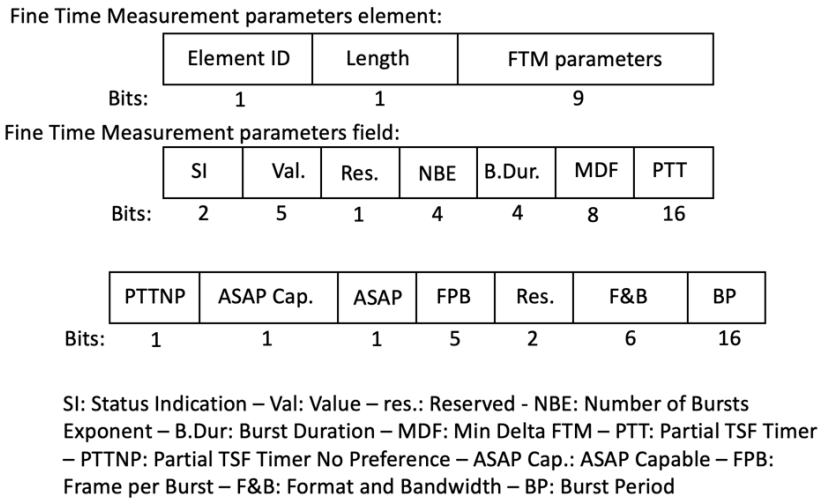


Figure 4.1 – The 802.11-2016 FTM Parameters Element that contains the FTM parameters negotiated by the ISTA and the RSTA

need to range with multiple sources. FTM was thus designed to allow for this multiplicity, which implied a more complex negotiation scheme. In effect, multiple constraints appear that need to be accounted for:

— The STA cannot range continuously, like with TM. It is more reasonable to exchange for some time, then stop (leaving the AP and STA perform other tasks, for example sending or receiving data, or ranging with another AP)

— The STA should not range only once, as the TM method has shown that multipath will cause small variations of the ToF. Therefore multiple exchanges are necessary.

— One set of exchanges might not provide the most accurate results. For example, some obstacle may be temporarily between the STA and the AP (*e.g.,* the user body, a piece of furniture, *etc.*) It is therefore better to operate several ranging exchanges, separated by some time interval (during which temporal obstacles may move).

— Using the Start/Stop TM method for each set of exchanges, adds overhead. It is more efficient for the STA to negotiate once multiple sets and consume the airtime for actual measurements, not for measurement planing.

— If the STA ranges against multiple APs, it may leave the channel between ranging sets. Depending on what happens on the other channels, the STA may then come back late. At the same time, the AP may also be delayed to start the next ranging exchange set (*e.g.,* because the AP is busy transmitting or receiving a frame at that time). Therefore, there must be some flexibility in the ranging set structure (its duration, the number of exchanges expected to occur, *etc.*)

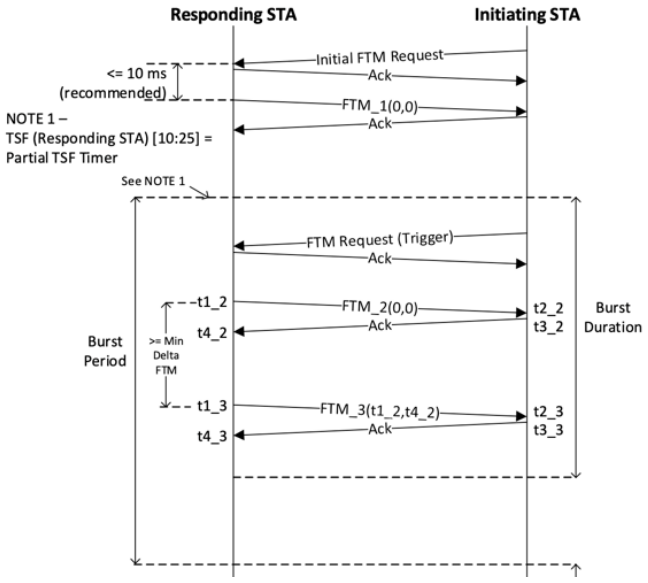


Figure 4.2 – The FTM choreography. An initial negotiation phase (top) is followed by one or more burst periods (bottom) during which FTM ranging frames are exchanged.

The roles also need to be clarified, as the STA could in fact range against any object with a fixed location. The STA is therefore renamed Initiating station (ISTA), and the AP, or any object that can be ranged against, a Responding station (RSTA).

Then, to account for all these challenges, a negotiation more elaborate than with TM occurs. The ISTA starts by sending an Initial FTM Request frame, that not only requests the AP for a ranging sequence, but also proposes multiple parameters represented in Figure 4.1. These parameters include how many ranging sets, or bursts, should be planned (using a Number of Bursts Exponent field), what should be the interval between two bursts (Burst Period), what should be the duration of each burst (Burst Duration), how many FTM exchanges should be expected within each burst (FTMs per Burst) and the expected interval between each FTM exchange within each burst (Min Delta FTM). These elements, and their flexibility, will be the object of a detailed analysis in section 6.2.

The ISTA proposes these parameters, and the RSTA responds with another initial FTM Request frame, in which the AP can either accept (and thus merely repeat) the parameters proposed by the ISTA, or override them by sending its own parameter values.

Then, at the time of the beginning of the first burst, both the ISTA and the RSTA are expected to be available on the channel. The RSTA sends an FTM frame, that is acknowledged by the ISTA, reproducing the TM structure and timers. Just like with TM, the subsequent FTM frame from the RSTA contains the (t_1, t_4) values (but the RSTA does not send a particular reference clock value, as the purpose is ranging, not clock synchronisation). The exchange continues

as negotiated for the planned duration and the number of bursts, as illustrated in Figure 4.2. The ISTA can then establish its distance d to the RSTA by using Equation (1.6). In a related exchange, the ISTA can request from the RSTA its Location Configuration Information (LCI), a set of geographical coordinates which logic is similar to that defined in RFC 6225 [115].

By repeating the process with multiple RSTAs, the ISTA should obtain enough ranges and LCIs to compute its own location (FTM does not describe how location is computed, and solely focuses on the ranging part).

4.3 A Critical Review of the FTM Design

It is important to keep in mind that FTM historically happened as an augmentation of TM, a protocol that was designed for very different purposes. FTM creation also did not happen in a vacuum, and the designers had the opportunity to look at other well-established localization protocols, such as GPS (section 2.2.2) or cellular techniques (section 2.3), and the indoor landscape, in particular UWB (section 3.3.3) that had been a common occurrence in specific verticals since 2007. Such comparison allows to evaluate what a modern localization protocol should include, with two specific angles: its ability to fulfill the intended use cases, and its technical efficiency.

4.3.1 Use Case Fulfillment Limitations

Recall from section 3.2 that indoor location is expected to meet use case requirements from both sides: the STA may need indoor navigation (section 3.2.1), or the ability to identify its proximity to specific PoIs. The infrastructure may need to locate individual assets (either with coordinates (section 3.2.3) or at proximity (section 3.2.2) of specific PoIs), and may need to collect statistics about mobile devices paths and presence (section 3.2.4).

Quite often, both sides' needs coexist in the same environment. Take the example of a shopping mall or alike public venue. Visitors that are new to the venue may need indoor navigation to find their way in an environment that they are not familiar with. They also need to know when they are near a particular PoI ("the 4 cm nails shelf") Quite clearly, staff (familiar with the venue) have no need for indoor navigation (with the possible exception of new staff in training). However, staff needs to be located for multiple reasons. When handling goods, the staff device is used as a proxy for the good put on shelves (see section 3.2.3). Thus locating staff helps build a map of the goods on display in the store. This aspect is critical to most retail businesses, as merchandise location tends to change based on seasons and sales events. In turn, this map can be used for customers to find their way to the product they need (with the obvious consequence that, if the retailer does not have a solution to locate staff devices and goods, no map can easily be built, and therefore indoor navigation for customer loses much of its appeal). In parallel, staff needs to be located for operations efficiency (*e.g.,* customer has a question on a particular

product and asks the first store crew member in range, who in turn needs to locate the person in charge of the particular product and accompany the customer to that person; in another example, sales or security personnel needs to be redeployed depending on crowd density in various areas of the venue, *etc.*)

However, FTM has no provision for infrastructure localization. The ISTA has all the timers required to compute the range, but the RSTA does not have any information about the ISTA timestamps. This gap was intentionally left in the design, because the primary intended use case was ISTA indoor navigation. However, this design intent, building on the GPS idea with an imitation of the UWB TWR model, ignored critical elements:

1. There is only limited incentive for a venue owner, or an infrastructure vendor, to implement FTM or any mode that resembles SS-TWR initiated from the STA side. As seen above, a venue owner may want to provide indoor navigation to customers, but also needs asset location to run the business. Without the second part, the requirement becomes to install a localization system for asset tracking, and an entirely different overlay system for FTM and navigation, as FTM can't do both. The benefits for business operations of asset tracking are clear, but the benefits of the added investment for FTM are not. Even if a venue owner was willing to make this double-investment, there is no feedback from the ISTA to the infrastructure, so the venue operator has no way to even know if FTM is being used, is accurate, efficient, provides a better user experience *etc.* By focusing solely on the ISTA navigation use case, the FTM design also removes the incentive for half of the participants (the infrastructure).
2. GPS and cellular localization did not suffer from these limitations. But the reason is historical. GPS was deployed by the US military because a particular category of users (military assets on the ground) had the need for their location in unknown places (see section 2.2.1). This requirement is not transposable to FTM, as FTM operates indoor in a location familiar to the venue staff. Cellular localization implements a mode where the UE can find its own location, but the model (see section 2.3.2) is integrated in a scheme where the infrastructure also typically obtains the UE location, even if this exchange is not directly visible to the user of the UE.
3. The discussions around FTM design show that one goal was to protect user privacy (by preventing unwanted infrastructure tracking). However, we claim that this view was naive. In the GPS system, the receiver is passive. As no signal is sent from the end-device, the end-device privacy is respected. But FTM implements a different mode, where the ISTA does send multiple frames. As will be seen in chapters 6 and 7, this does not provide enough elements for legitimate tracking by the infrastructure, but does provide ample elements for ISTA tracking by a bad actor. In the cellular system, there is an assumption that exchanges with the eNB are secured, simply because their observation requires specific equipment.

With unlicensed technologies like 802.11, the observation is trivial unless communications are encrypted (and FTM does not require any encryption). Therefore, comparing FTM logic to the cellular equivalent UE-based location misses many factors that are critical to the security and the privacy of the exchange.

4. The same discussions never examined the privacy of the AP location. In the FTM scheme, any ISTA can query for an AP LCI, and obtain the precise location of the AP (with an accuracy of up to 2 cm at the equator). There is no mechanism to protect that exchange. Here again, we claim that this view was naive. In many places, the precise location of networking assets is not made available to whomever asks. For example, hospitality often has an aggressive approach about Wi-Fi coverage (there are web sites that sort hotels by the quality of their Wi-Fi coverage, and Wi-Fi is commonly seen by business travelers as an essential commodity). A hotel may want guests to find their way to their room, but may not want anyone (*e.g.,* competitors) to know and publish where each AP is, how many rooms are covered by each AP, *etc.* In other venues, APs are hidden because they may be stolen, and a location accurate to 2 cm is not helping to alleviate the issue.

4.3.2 Technical Design

The choice of strict ToF, using Equation (1.6), also deserves to be weighted. By focusing on a distributed technique (see Section 1.2.2), the technical design assumes that the ranging need focuses on near-real-time location. It is then clear that the ISTA should evaluate its surrounding RF conditions, then operate a rapid series of exchanges to collect enough sample that a good ranging estimation can be made. By allowing the AP to override the parameters (see Section 4.2.2 above), the protocol allows the AP to insert parameters relevant to its own RF conditions, which is logical. However, by proceeding in this order (ISTA first, RSTA last), the procedure gives the final words to the RSTA. But the RSTA does not have the final timestamps, and is therefore unable to evaluate if the parameters were satisfactory or would need to be changed for the next exchange. Thus, by extending TM, FTM inserts inefficiency in its design.

On the other hand, the choice of interval-based ToF with Equation (1.6) is expected to remove the concerns related to clock drift. As detailed in Section 1.3.2, the drift issue still exists at the scale of the intervals. One interesting aspect of this issue is that, in 802.11, an ACK frame is expected to be sent as a response to a unicast frame after a known interval (the Short Inter-Frame Space, SIFS, which duration depends on the version of the standard, but is 16 μs in 802.11n, 802.11ac and 802.11ax). This SIFS incorporates the time required for the receiving circuit to process the incoming frame at PHY and MAC level. Therefore, it is expected to be fairly constant, and thus $(t_3 - t_2)$ should be $16\mu s - \delta$, where δ is the clock speed difference between the ISTA and the RSTA. But the unknown values that are communicated are instead (t_1, t_4), which is a much larger interval than $(t_3 - t_2)$, and depends both on the distance and

the RSTA clock speed. Thus, the difference in clock rates between the RSTA and the ISTA can still contaminate the ToF. This issue is not specific to FTM, and is inherent to Equation (1.6), unless a verification phase is introduced (as in DS-TWR in Section 3.3.3).

On the bright side, the use of direct distance measurements (and not hyperbolic structures as in Section 1.4.1) simplifies the deployment and the computation. Each range measurement is directly usable (no need to compare it to measurements against another anchor). Additionally, the hyperbolic structure is well known to display large errors when the intersections move away from the vertices and toward the asymptotes. Using direct measurements (radii) avoids this problem, that plagues all TDoA systems.

At the same time, multipath can cause large inaccuracies on the signal time of arrival (ToA), as detailed in Section 4.4.2. Although 802.11-2016 is mute on this point, this practical difficulty means that all implementations we have tested tend to measure the AP RSSI (through a probe request/response exchange, through which the ISTA discovers the AP and its support for FTM), and prefer the AP with highest RSSI. These APs are likely to be the closest, and have therefore a higher chance of being in LoS conditions (this choice also limits the drift). On the other hand, because their range is shorter, any mis-measurement caused by multipath will have a more dramatic effect (a given ToA error is proportionally heavier as the ToF is smaller).

Along the same lines, 802.11-2016 makes no recommendations on RSTA placements. Yet, the Standard implicitly assumes (and so does 802.11az in next section), that the primary case for an RSTA is an AP (as many functions expected of an RSTA suppose beacons or probe responses, which are only sent by APs). But we saw in Sections 2.4.2 and 3.3.6 that a convex hull was needed to operate proper location indoor. This supposes that some devices, operating as RSTAs (and therefore APs) should be installed at the edge of the building, along the walls, where they would have limited benefit for data coverage (because about half their RF cell would be outside of the building). Alternatively, standard AP placement could be used, but at the cost of lower accuracy (or complete loss of usability) at the edge of the building. The same assumption also limits the use of FTM for proximity (see Section 3.2.2), unless each PoI of interest suddenly incorporates an AP function.

4.3.3 802.11az Privacy Partial Remediation

This analysis is by no mean intended to be a critique of FTM, nor a complete list of issues (and the next chapters will examine other issues). After the 802-11 update of 2016 with the integration of FTM, another task group was created in the IEEE 802.11 Working Group, under the code 802.11az, to work on Enhancements for Positioning, and thus augment FTM with new features, and address some of the initial designs limitations.

Several of the problems above were raised, by the author of this thesis and others. In an effort to ensure better privacy, a passive mode was proposed. In that mode, some fixed points

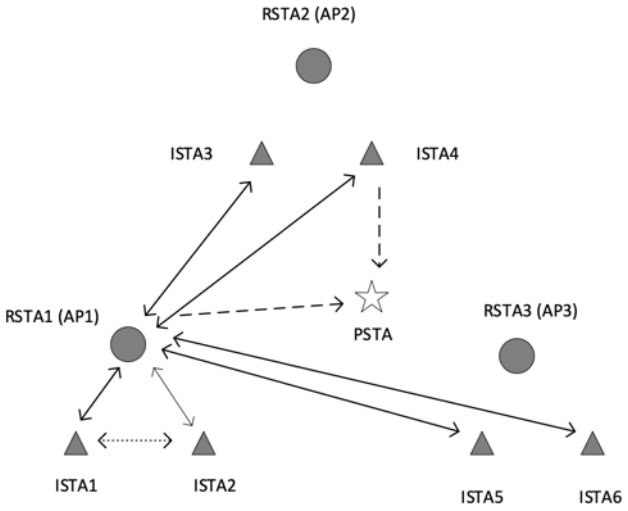


Figure 4.3 – 802.11az Passive mode, where the PSTA listens to the FTM exchanges between on RSTA and multiple fixed points acting as ISTAs, and computes a form of TDOA to deduce its location.

become ISTAs, and range against another fixed point operating as the RSTA as shown in Figure 4.3. Another station, the Passive STA (PSTA), listens to these exchanges and deduces its own location.

The location determination is based on the same logic as the reverse TDoA technique used for UWB and detailed in Section 3.3.3. The timestamp terminology is reverse to that of FTM (in this case the ISTA triggers the exchange at time t_1) and is represented in Figure 4.4. The ISTA and the RSTA display, in the subsequent exchanges, the time of departure of the respective frames they sent.

The PSTA detects all transmissions between each ISTA and the RSTA and receives all the times of departure and arrival. From these numbers, the PSTA can compute its location relative to each RSTA/ISTA pair. The PSTA attempts to determine the difference in time of flight between frames coming from the ISTA and frames coming from the RSTA, by comparing the time at which the PSTA receives a frame to the time at which the fixed stations (RSTA or ISTA) declare receiving that same frame. More formally, if we denote with a prime sign any timestamp declared by an ISTA or RSTA and translated into the PSTA time reference, the difference of time of flight $D_{ToF(PRI)}$ can be determined as follows:

$$D_{ToF(PRI)} = t_6 - t_5 - 0.5t'_3 + 0.5t'_2 - 0.5t'_4 + 0.5t'_1 \tag{4.1}$$

This equation is fascinating to the author of this thesis in multiple ways. First, it is simple and elegant. It is also functional regardless of the relative position of the PSTA to the ISTA and RSTA. Last, by bringing all times into the PSTA reference, it is strictly linear. However, it is worth comparing to other techniques that incorporate similar TDoA concepts into a proce-

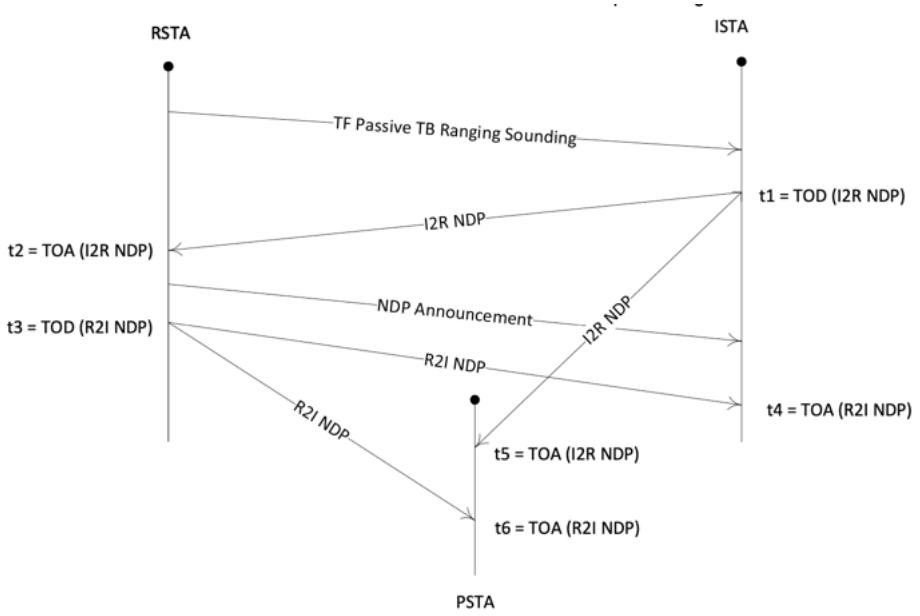


Figure 4.4 – In 802.11az Passive mode, all ISTAs and RSTAs announce their timestamps, the PSTA translates them into its own time reference, and computes its relative position.

dure. Take Equation (2.2) as an example. Although GPS satellites incorporate highly accurate clocks, $c\Delta T_i$ incorporates the distance error due to the measurement noise and the receiver clock inaccuracy. Similarly, Equation (2.5) used for cellular O-TDoA incorporates n_i, the errors due to measurement noise and clocks drifts. Unfortunately, the PSTA equation makes no such provision. Clearly, it could be claimed that Equation (4.1) only focuses on $D_{ToF(PRI)}$, and that the measurement noise can be incorporated into the system of equations that converts these differences into a position. However, the procedure provides no guidance as to what this system should look like. The procedure also provides no mechanism to convert the RSTA and ISTA announced times into the PSTA clock of reference. Therefore, the addition of this passive mode helps address the privacy concern, but it is very likely that the procedure will need to be modified to become practically usable.

Additionally, the adoption challenge remains that a venue has very little incentive in implementing a method that provides no benefit to the venue operations. One additional difficulty of the passive mode (over the standard FTM mode) is that, the STA being entirely passive, the venue would not even know if devices are using the ranging exchanges that are in operation and consuming airtime between fixed point.

4.3.4 802.11az Feedback Partial Remediation

In an effort to increase the attractability of the technology for infrastructure owners, a feedback mechanism was proposed by a group (with the author of this thesis [4]). At the end of

FTM exchange, this new mode allows the ISTA to return to the RSTA its (t_2, t_3) values, thus allowing the RSTA to know its range to the ISTA. By combining the reported ranges to multiple RSTAs, the infrastructure can then compute the ISTA position with the standard location techniques examined in Section 1.3.2. This technique has the benefit that it is designed with a user input. In other words, the user allows or blocks such feedback, thus allowing infrastructure tracking or not. This mode is still imperfect, because a proper design should account for three types of zones within any venue:

— Zones where individual tracking is never required (*e.g.,* common public zone in a shopping mall, a train station, *etc.*) In these zones, a mechanism for traffic analytic should exist (unfortunately, such mechanism does not exist for FTM)

— Zones where individual tracking is possible upon user opt-in (*e.g.,* individual store in a shopping mall)

— Zones where individual tracking is always mandated (*e.g.,* hazardous warehouse areas at the back of stores, restricted areas in hospitals *etc.*)

FTM does not distinguish between these different cases. Additional issues exist, that we consider as solved, and that will be examined in the next three chapters.

4.4 Testing FTM Technical Efficiency

In order to evaluate FTM, and as the implementation of the Standard was still inactive development for most actors in the Wi-Fi field, we implemented FTM support on an AP, and tested extensively with a client device (whose driver team was developing support as well at the same time). This section summarizes our findings.

4.4.1 Bandwidth and Hardware Delay

A first task when implementing FTM support is to design a mechanism to reflect the time of departures (ToD) and time of Arrival (ToA) for the frames, so as to be able to timestamp the various frames. The Standard describes that those times are intended as those at which the beginning of the frame preamble leaves or arrives at the antenna. The physical layer of each 802.11 frame starts by various training fields (the details depends on the frame type, *e.g.,* 802.11a, 802.11n, 802.11ac, 802.11ax). The first training field is a simple sequence of energy changes that allow the receiver to (i) detect the arrival of a frame (ii) calibrate its clock on the transmitter clock and (iii) identify the exact position of the signal in the frequency domain.

The MUSIC algorithm detailed in Section 2.3.2 can be used to identify the time of arrival. However, the efficiency of this detection is highly related to the bandwidth. The receiving circuit samples the active channel at regular intervals. These intervals depend on the expectation of

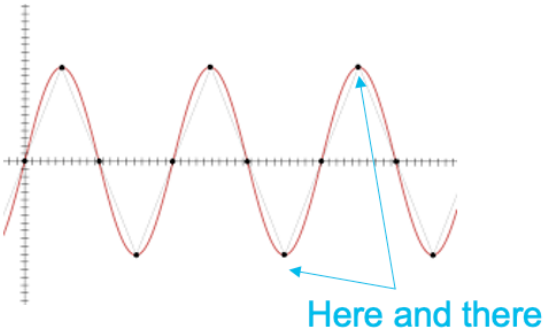


Figure 4.5 – To correctly interpret the signal, sampling the channel has to be done at least at twice the channel bandwidth.

the signal bandwidth. This is easily understood with an example (the example assumes a single signal *sin* cycle in the time domain to make the explanation more intuitive). The received signal is a wave of energy, where the potential of the electric field oscillates between two values (that we'll call positive and negative), and that is represented by a sin wave, as shown in Figure 4.5.

In order to detect this wave and the associated variation of energy cycle, the receiver needs to collect one sample of the channel as the energy increases (or gets to the max positive value) and another one when the energy decreases (or gets tot the max negative value), in other words twice per cycle.

Therefore, if the signal is sent over a bandwidth of 1 Hertz (thus expressing that the *sin* wave expresses one cycle per second), the receiver needs to collects two samples per second over that 1 Hz bandwidth. If the bandwidth is larger, for example 20 MHz (thus expressing 20 million cycles per second), the collection rate has to be 40 million samples per second. At that rate, the channel is sampled every 25 nanoseconds. But light travels about 30 cm per nanosecond. Therefore, when the receiver samples the channel and detects no energy, then detects energy when sampling the channel at the next interval, it can know that the signal started arriving between "right after the previous sampling" (25 ns ago) or "just now" (0 ns ago). As the receiver has no mechanism to know the exact arrival time, we can establish an average estimation uncertainty toward the mean of that interval, *i.e.* 12.5 ns. Translating this uncertainty delay into distance (traveled by the signal) gives an uncertainty of 3.75 meters. In reality, the uncertainty is higher, because the detection is also affected by other factors. For example, it takes time for the energy to accumulate to a level where detection occurs (this is the signal rise time), therefore the detection is always after the signal has started to arrive. A common uncertainty lower bound at 20 MHz is therefore of about 5 meters.

If the monitored channel is instead 40 MHz-wide, then the sampling rate is 80 million samples per second, thus 12.5 ns inaccuracy interval represented by its mean at 6.25 ns, and thus an inaccuracy lower bound of about 2.5 meters, then 1.5 meters at 80 MHz. This is the reason why UWB provides high accuracy (the channel a bandwidth is 500 MHz). 802.11 allows channels of

up to 160 MHz with 802.11ax, but the availability of the spectrum and the density of APs limit the channel overlap and reuse, thus limiting most deployments to 40 or 80 MHz. Therefore, a larger bandwidth will provide more accurate ranging results. This section shows results for 80 MHz experiments.

Naturally, the antenna is a piece of copper and does not have a clock. The received signal travels to the circuitry to be processed, and the component documenting the time of the preamble arrival is deeper in the system. Similarly, when a frame is generated, a circuit modulates the signal, and the signal then travels to the antenna. There is therefore a hardware delay between the "time at the antenna" and the "timestamp record". This hardware delay needs to be compensated for, by ranging at different known positions between similar devices and inserting a delay factor to the recorded timestamps until the reported range is closest to the actual range.

4.4.2 Calibration challenges

However, this approach is often insufficient. The above hardware delay was tuned in an isolation chamber, to avoid the possible local effect of multipath with 5 and 10 meter distances, until the obtained range was within 50 cm of the real distance. In each burst the shortest ToF was retained (because the shortest ToF is likely to be closest to the LoS transmission - this is a case where the mean value is not meaningful). In the next phase of the implementation, we bring the AP into a real office environment, and range against a laptop ISTA (instead of ranging AP to AP), that was also calibrated in an isolation chamber (against another laptop of the same model). The results are shown in Figure 4.6.

The red points represent the ground truth distance (values displayed on the x axis). The blue points are the ranges obtained at the conclusion of individual bursts (the range is computed from the shortest ToF in the burst). The blue line goes through the mean of the computed ranges. Despite the calibration, two elements of inaccuracy appear:

— The blue dots are distributed over a large set of values. At 5 meters, although the mean calculated range is close to the ground truth, the individual values of each bursts are spread between 1.47 m and 10.25 m. The consequences of such large distribution is that many bursts are needed to obtain a valid measurement. However, the tested systems are not moving, nor are the operators. Despite multipath, obtaining a smaller set of values would be desirable.

— The calibration on each side is not symmetrical. The computed distance mean is close to the real distance at 5 meters, but the evaluations of both sides diverge as the distance increases. This means that the internal logic used by each side to establish the ToF or ToA is different.

Overcoming these two issues takes coordination between the client and the AP algorithms. In

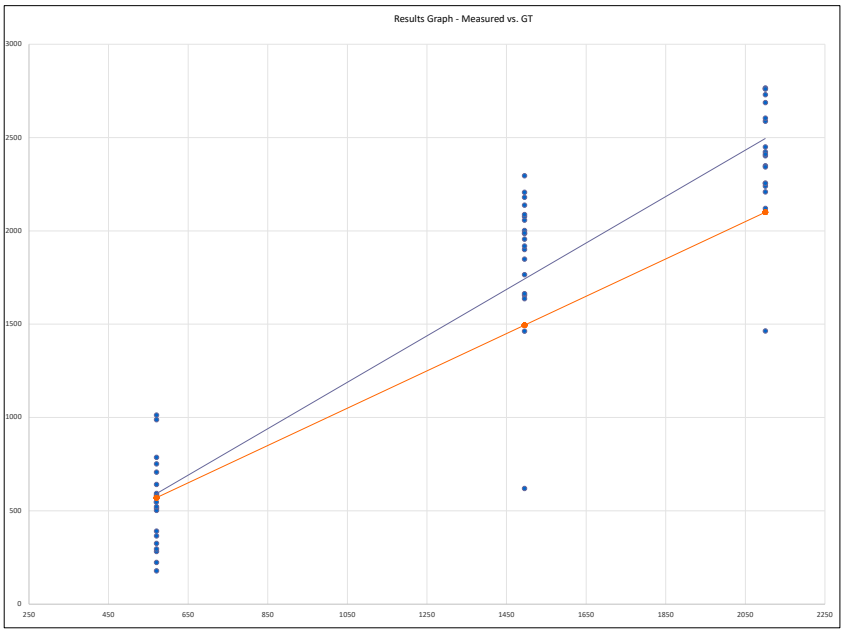


Figure 4.6 – Initial real field FTM measurements after calibration in isolation chamber.

this early implementation, a Kalman filter (see Section 1.5) is applied on the AP measurements, to constrain subsequent t_1 values, and the (t_1, t_4) intervals within a reasonable range. This is because we expect that, in a burst, the AP would send frames at regular pace, and the ISTA would acknowledge each frame after a standard $SIFS$ (short inter-frame space) period, as per the 802.11 Standard. Any outlier is therefore likely a faulty estimation and is discarded. However, no particular filter is on the client side. As a result, any processing delay on the client suddenly increases the (t_1, t_4) interval and is discarded on the AP, but the client knows that it added delay (because it knows (t_2, t_3)). However, any processing delay on the AP is taken as is by the client, introducing more outliers. This difference of treatment on both sides causes an increasing divergence as the distance, and therefore multipath, increases. After multiple exchanges between the AP and the client teams, we decide to implement a filter on both sides, using a system simpler than Kalman (an alpha-beta filter, that uses simpler variables and is less processor-intensive than the Kalman version).

To improve the system further, the channel state information (CSI) is also recorded. The CSI reflects the conditions of the channel. This element is useful to adapt the transmission matrices and group clients together for the AP data transmissions, but it is also useful in our case to bound the outliers. An unstable channel should allow for a larger range of timer intervals than a stable channel (where multipath should be less disruptive). This method also allows the AP to evaluate the ISTA initial FTM Request frame, and decide whether to accept or override the proposed parameters (see Section 4.2). This mode will also prove invaluable to solve an other

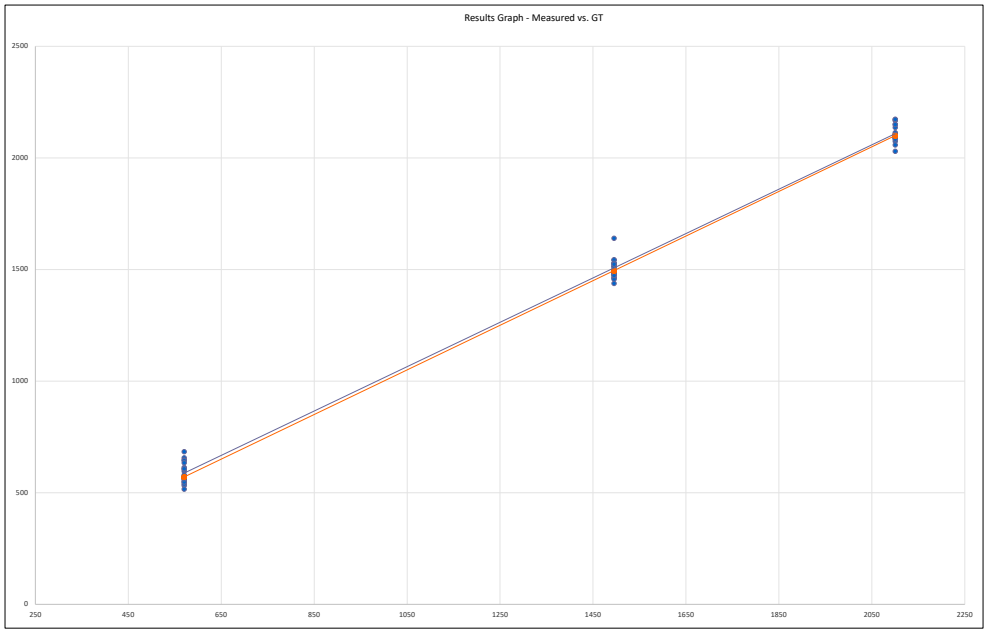


Figure 4.7 – Real field FTM measurements after AP/STA algorithms unification strategy

issue detailed in chapter 6.

Another concern relates to the implementation of the MUSIC algorithm on both sides. In a strong multipath environment, the LoS component does not always reach a detection level sufficient to be accounted for, leaving the first reflected component to be detected as the correct LoS signal. This issue particularly affects the AP, because it is set to send and receive all management frames from its first radio chain (although the AP is a 4x4:4 AP, thus can use up to 4 radio-chains). This is because management frames are always sent at the lowest mandatory rate, which supposes a single transmission chain (no beamforming or Multiple Input, Multiple Output [MIMO], structure). The client is a 2x2:2, but uses both radio chains on the receive path. Therefore, if one radio chain is affected by multipath (causing the LoS signal to be below detection threshold), there is a good chance that the other radio chain would not be affected the same way, thus increasing the chances of detection of the LoS signal ToA. However, with a single radio chain, this chance does not exist on the AP, increasing the risk of mis-detection or outlier on the AP. Following this structure, the AP is re-programmed to forward all 4 radio chains signal on the receive side, letting the system select as LoS the first reported signal.

After these adjustments, the system is tested again as displayed in Figure 4.7. All reported ranges are within 50 cm of the ground truth position, and the set is linear over distance increase.

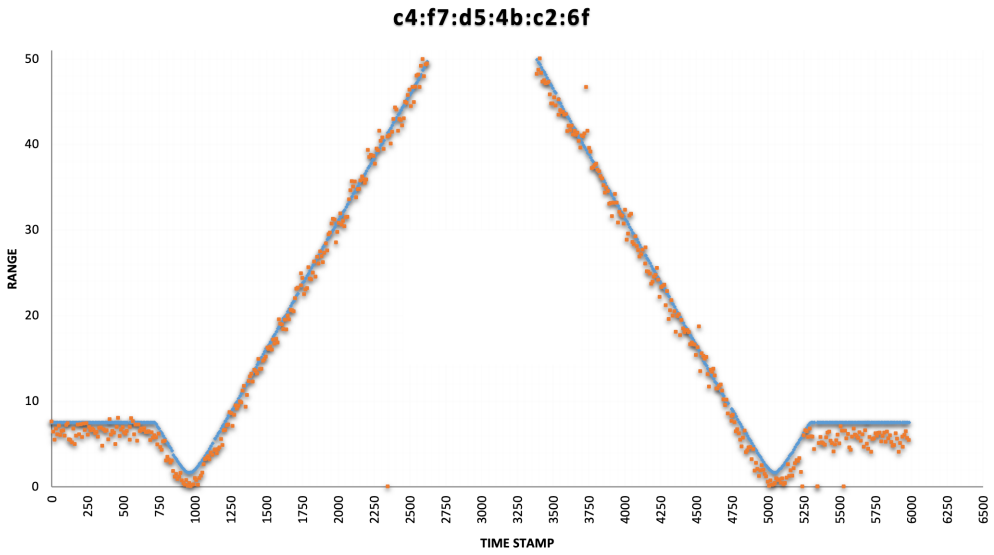


Figure 4.8 – FTM Mobility experiment. True ISTA distance to the RSTA is in blue, FTM computed distance is in orange.

4.4.3 Mobile Tests

We tested this implementation in real building conditions. The development of the solution detailed above included tests, but these were realised with the client at a static position (and of course, the AP also at a static position). But in a real setting, a client would likely monitor the device position on a map while moving toward a target destination, therefore mobility is a key component.

To represent this scenario, our final test is set in an automation lab. The lab is an office building with cubicles and closed offices. The building is 70 meter-long. The AP is installed at one end of the building floor. The client is installed on an automation robot. Then, as the robot moves toward the other side of the building and back, the client ranges against the AP. The system clocks are synchronized. In this validation lab, the robot distance to the AP is continuously measured with lasers.

Figure 4.8 represents the outcome of this validation experiment. The blue line in blue represents the ground truth (true distance between the AP and the client), and the orange points are the distance evaluations realised with FTM. Despite a few outliers, most measurements are very close to the ground truth.

Figure 4.9 represents the measurement errors (difference between the ground truth and the range measured with FTM). 90% of the measurements are within 1.55 meters of the ground truth. As each range burst typically includes 8 measurements or more, these results indicate that, on an 80 MHz channel, a range accuracy of 1.5 meters is a realistic target.

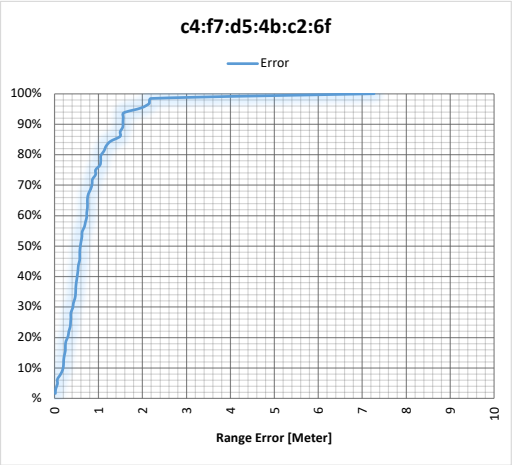


Figure 4.9 – FTM Mobility experiment. Cumulative Distribution Function of the error (ground truth distance vs. computed distance).

Naturally, this implementation validates one AP with one client. In the real world, any AP supporting the RSTA function may need to range with any ISTA. Thus, the system is only valid if all RSTAs, all ISTAs, implement similar mechanisms. This is where the 802.11 Standard stops, and where other organisations, like the Wi-Fi Alliance, play a major role. As the community of AP and STA vendors were looking at implementing FTM, multiple exchanges and joint test events occurred between vendors, to ensure that the above schemes were agreed upon, and implemented by all. The addition of a light machine learning structure on the client side [56] then allowed the ISTA to even improve on these results and achieve sub-meter accuracy in LoS conditions.

4.5 Conclusion

At the outset of this chapter, it is clear that FTM suffers from design challenges. Having been designed as an augmentation of another protocol that was developed for entirely different purposes, the original FTM missed several critical use cases while presenting severe privacy and technical gaps. Some of these gaps are addressed in 802.11az, making FTM a workable solution for indoor localization. Yet challenges remain, that will likely mandate multiple iterations of 802.11 Standard revisions, before the improved FTM becomes a solution that solves all use cases and fulfills all requirements of a modern indoor localization technology.

Yet FTM is very promising. Wi-Fi is present everywhere, and this chapter shows that a ranging accuracy of near 1.5 meter can easily be achieved. In the next chapters, we will examine FTM further, and identify critical weaknesses (for which we will propose solutions) that, left unanswered, would prevent the wide adoption of the technology.

Chapter 5

Resolving the RSTA Placement

At the conclusion of the previous chapter, it is clear that FTM has design challenges, but can also achieve good ranging accuracy. 802.11 is also widely adopted worldwide, and a ranging protocol that relies on 802.11 exchanges also presents high potential for adoption, especially in indoor areas where GPS or cellular lateration are not available or insufficient. In these scenarios, the station could complement the outdoor techniques by ranging against indoor RSTAs, which location is known and shared, and thus realise against indoor anchors the location that outdoor anchors could not provide.

However, FTM, as initially designed, presents unsolved challenges. This chapter will examine and propose a solution for the first one, which is the assumption that an ISTA can range against an RSTA, which location is known.

5.1 The Challenges of RSTA Placement for Indoor Localization

However, this idea introduces the very problem that it aims to solve. FTM supposes that RSTAs, positioned indoor, can be configured with their correct geo-position. Yet configuring their geo-position is difficult. Objects intended to act as RSTAs (APs, digital signage and other static objects - we will call them anchors for simplicity) typically do not include a GPS sensor. Even if they would, GPS is challenged indoor. Most venue owners using network management tools are used to uploading a floor plan and positioning AP icons to approximate positions (based on visual landmarks in the real building), but informing the geo-coordinates of each AP is a much more difficult task (especially as, in most cases, venue owners do not even need geo-coordinates details for the building, a mere "Lyon Building 4" denomination is sufficient). Many implementers are left with manual and time consuming techniques, where the geo-positions of points at the edge of the building are fist established (using mobile devices GPS functions outside, or high-resolution reference aerial pictures), then high precision ranging techniques (laser, Ultra Wide Band, *etc.*) are used to finally come to an evaluation of each sensor position and location. The process has to be reiterated each time an RSTA is moved or added. This is a major obstacle for the adoption of FTM. Without RSTA accurate LCI, the entire FTM process becomes unusable.

Therefore, our first research task is to find a way to automate the RSTA placement and coordinates resolution process. In an ideal version, the RSTAs should find a way to self-locate

themselves and find their own geo-coordinates without the need for IT admin input. The method we propose in this chapter achieve these goals. We found that, as soon as the geo-position of one or more device is known, the other sensors can use FTM to self-locate one another, considerably simplifying the deployment of an FTM-ready infrastructure. Solving this problem implies seeding the system with one or more initial positions, which is trivial and can be achieved with a mobile device ranging to one or more RSTAs from outside the building. The solution also implies solving the issue of self-positioning, where the various anchors can reliably establish their respective position based on noisy environments. In short, the proposed method operates in two phases. Noting that pairwise FTM distance noise is not symmetrical, we first propose to reduce the noise by identifying its contribution to geometric distortion of the triangles formed by AP sets, in a phase called Wall Remover. By identifying the effect of inner walls of FTM exchanges between AP pairs, this phase allows for the reduction of the effect of walls on FTM measurements. In a second phase, a recursive examination of all possible Euclidean distance matrices is conducted, to identify the anchors least affected by noise. By combining standard EDM techniques with geometric projections, this phase allows the selection of the best sub-matrices (least effected by remaining measurement noises) to compute the positions of anchors which are most likely to be accurate. From their position, the location of all other anchors is computed, leading to a better position accuracy than EDM resolution techniques alone.

5.2 Multidimensional Scaling (MDS) background

This chapter focuses on FTM exchanges between fixed points. In this context, Wi-Fi access points and other static devices (*e.g.* digital signage) can be configured to play the role of ISTAs and RSTAs, alternating between one and the other and ranging to one another over time. As the points are fixed, and as each ranging burst takes a few hundreds of milliseconds (*e.g.* a burst of 30 exchanges would consume about 250 ms in good RF conditions), a large number of samples can be taken (*e.g.* a burst per minute represents 43200 samples per 24-hour period). This flexibility allows for obtaining ranges between pairs far apart, and for which only a few exchanges succeed. From all exchanges, only the best (typically smallest) distance is retained, as will be seen later. The output of these FTM measurements is a network of $n > 0$ nodes, among which $0 \leq p \leq n$ nodes have a known position.

The measured distance between these nodes can be organized in a matrix that we will note $\tilde{D}$. Conceptually, the set is similar to any other noisy Euclidean Distance Matrix (EDM). The set contains an exhaustive table of distances $\tilde{d}_{ij}, \forall i, j \in 1, \ldots, m$ between points taken by pairs from a list of $0 < m \leq n$ points x in N dimensions (x_i and $x_j \in \mathbb{R}^N$ for any point x_i). Each point is labelled ordinally, hence each row or column of an EDM, *i.e.* $\forall (i, j) \in [\![0; m]\!]^2$, individually addresses all the points in the list. The main task of the experimenter is then to

find the dimension N and construct a matrix of distances that on one hand best resolves the noise (which causes inconsistencies between the various measured pairs), and on the the other hand is closest to the real physical distances between points, called ground truth, and which matrix is noted D. Such task is one main object of Multidimensional Scaling (MDS). Resolving such matrix is a complex problem, which challenges are summarized in [116]. [117] provides an overview of the main resolution techniques. In the particular case of sensor location, [118] shows that the problem can be convex if the dimension space is known, which is often the case for RSTAs in FTM (but not when using FTM alone, as we will show). An approach close to ours, to reduce the problem with sub-matrix spaces, is detailed in [83], although the proposed method is strictly algebraic (while we propose a geometric component). As the distances are organized in a matrix, algebraic approaches are natural. Some authors, like Doherty and al. [119], explore geometric resolutions in scenarios where link directionality matters. We will show that the addition of learning machine can considerably enhance the resulting accuracy.

MDS draws its origin from psychometrics and psychophysics. MDS considers a set of n elements and attempts to evaluate similarities or dissimilarities between them. These properties are measured by organizing the set elements in a multi-dimensional geometric object where the properties under evaluation are represented by distances between the various elements. MDS thus surfaces geometrical proximity between elements displaying similar properties in some dimension of an $\mathbb{R}^N$ space. The properties can be qualitative (non-metric MDS, nMDS), where proximity may be a similarity ranking value, or quantitative (metric MDS, mMDS), where distances are expressed. This chapter will focus on mMDS.

Two main principles lie at the heart of mMDS: the idea that distances can be converted to coordinates, and the idea that during such process dimensions can be evaluated. The coordinate matrix X is an $n \times N$ object, where each row i expresses the coordinate of the point i in N dimensions. Applying the squared Euclidean distance equation to all points i and j in X (Euclidean distance $d_{ij}(X)$, which we write d_{ij} for simplicity) allows for an interesting observation:

$$d_{ij}^2(X) = d_{ij}^2 = \sum_{a=1}^{N}(x_{ia} - x_{ja})^2 = \sum_{a=1}^{N}(x_{ia}^2 + x_{ja}^2 - 2x_{ia}x_{ja}) \tag{5.1}$$

Applying this equation to the distance matrix D, noting $D^{(2)}$ the squared distance matrix for all d_{ij}, c an $n \times 1$ column vector of all ones, x_a the column a of matrix X, noting e a vector that has elements $\sum_{a=1}^{N} x_{ia}^2$, and writing A' the transpose of any matrix A, the equation becomes:

$$D^{(2)} = ec' + ce' - 2\sum_{a=1}^{N} x_a x_a' = ec' + ce' - 2XX'. \tag{5.2}$$

This observation is interesting, in particular because it can be verified that the diagonal elements of XX' are $\sum_{a=1}^{N} x_{ia}^2$, *i.e.* the elements of e. Thus, from X, it is quite simple to compute

the distance matrix D. However, mMDS usually starts from the distance matrix D, and attempts to find the position matrix X, or its closest expression U. This is also FTM approach, that starts from distances and attempt to deduce location. As such, mMDS is directly applicable to FTM.

Such reverse process is possible, because one can observe that D, being the sum of scalar products of consistent dimensions, is square and symmetric (and with only positive values). This observation is intuitively obvious, as D expresses the distance between all point pairs, and thus each point represents one row and one column in D. These properties are very useful, because a matrix with such structure can be transformed in useful ways, in particular through the process of eigendecomposition, by which the square, positive and symmetric matrix B of size $n \times n$ can be decomposed as follows: $B = Q\Lambda Q'$, where matrix Q is orthonormal (*i.e.* Q is invertible and we have $Q^{-1} = Q'$) and Λ is a diagonal matrix such that $\forall (i,j) \in [\![1;n]\!], \lambda_{i,j} = 0 \forall i \neq j$ and $\lambda_{ii} \neq 0$. Values $\lambda_{ii}, \forall i \in [\![1;n]\!]$ of Λ are the eigenvalues of B. Eigenvalues are useful to find eigenvectors, which are sets of individual non-null vectors u_i that satisfy the property: $Bu_i = \lambda_i u_i$, *i.e.* the direction of u_i does not change when transformed by B.

In the context of MDS, this decomposition is in fact an extension of general properties of Hermitian matrices, for which real and symmetric matrices are a special case. A square matrix B is Hermitian if it is equal to its complex conjugate transpose B^*, *i.e.* $B = B^*$.

In terms of the matrix elements, such equality means that $\forall (i,j) \in [\![1;n]\!], b_{ij} = b_{ji}$. A symmetric MDS matrix in $\mathbb{R}^N$ obviously respects this property, and has the associated property that the matrix can be expressed as the product of two matrices, formed from one matrix U and its transpose U', thus $B = UU'$. Because this expression can be found, B is said to be positive semi-definite (definite because it can be classified as positive or negative, positive because the determinant of every principal submatrix is positive, and positive semi-definite if 0 is also a possible solution). A positive semi-definite matrix has non-negative eigenvalues.

This last property is important for mMDS and for the FTM case addressed in this chapter. As $B_{ij} = B_{ji}$ can be rewritten as $(B - \lambda_i I)u_i = 0$, it follows that non-null eigenvectors are orthogonal. As such, the number of non-null eigenvalues is equal to the rank of the matrix, *i.e.* its dimension. This dimension is understood as the dimension of the object studied by MDS. With FTM, this outcome determines if the graph formed by APs ranging once another is in 2 dimensions (*e.g.*, all APs on the same floor, at ceiling level) or in 3 dimensions (*e.g.*, APs on different floors). In a real world experiment, B is derived from $\tilde{D}$ and is therefore noisy. But one interpretation of the eigendecomposition of B is that it approximates B by a matrix of lower rank k, where k is the number of non-null eigenvalues, which can then represent the real dimensions of the space where B was produced. This is because, if q_i is the i-th column vector

of Q (and therefore q_i' the i-th row vector of Q'), then $B = Q\Lambda Q'$ can be written as:

$$B = [\lambda_1 q_1 \ \lambda_2 q_2 \ \dots \ \lambda_n q_n] \begin{bmatrix} q_1' \\ q_2' \\ \vdots \\ q_n' \end{bmatrix} \tag{5.3}$$

And thus

$$B = \lambda_1 q_1 q_1' + \lambda_2 q_2 q_2' + \cdots + \lambda_n q_n q_n' \tag{5.4}$$

Therefore, if $n - k$ eigenvalues are 0, so is their individual $\lambda_j q_j q_j'$ product, and an image of B can be written as:

$$C = \lambda_1 q_1 q_1' + \lambda_2 q_2 q_2' + \cdots + \lambda_k q_k q_k' \tag{5.5}$$

C and B have the same non-null eigenvectors, *i.e.* C is a submatrix of B restricted to the dimension of B's non-null eigenvalues. In noisy matrices, where distances are approximated, it is common that all eigenvalues will be non-null. However, the decomposition should expose large eigenvalues (*i.e.* values that have a large effect on found eigenvectors) and comparatively small eigenvalues (*i.e.* values that tend to reduce eigenvectors close to the null vector). Small eigenvalues are therefore ignored and considered to be null values that appear as non-zero because of the matrix noise.

An additional property listed above is that positive semi-definite matrices only have non-negative eigenvalues. In the context of MDS, this is all the more logical, as each eigenvector expresses one dimension of the $\mathbb{R}^N$ space. However, noisy matrices may practically also surface some negative eigenvalues. The common practice is to ignore them for rank estimation, and consider them as undesirable but unavoidable result of noise. We will apply the same principles for measurements obtained with FTM. However, it is clear that the presence of many and/or large negative eigenvalues is either a sign that the geometric object studied under the MDS process is not Euclidian, or that noise is large, thus limiting the possibilities of using the matrix directly, without further manipulation.

Therefore, if from equation 5.3 above, one defines $B = XX'$, then an eigendecomposition of B can be performed as $B = Q\Lambda Q'$. As scalar product matrices are symmetric and have non-negative eigenvalues, we can define $\Lambda^{1/2}$ as a diagonal matrix with diagonal elements $\lambda_i^{1/2}$. From this, one can write $B = (Q\Lambda^{1/2})(Q\Lambda^{1/2})' = UU'$. The coordinates in U differ from those in X, which means that they are expressed relative to different coordinate systems. However, they represent the geometric object studied by MDS with the same dimensions and scale. Thus, a rotation can be found to transform one into the other.

With these transformations, mMDS can convert a distance matrix into a coordinate matrix, while estimating the space dimension in the process. Thus, classical mMDS starts by computing, from all measured distances, the squared distance matrix $\tilde{D}^{(2)} = \left[\tilde{d}_{ij^2}\right]$. The measured distance matrix $\tilde{D}$ is also commonly called, in MDS in general, the proximity matrix, or the similarity matrix. Next, a matrix called the *centering matrix* J is computed, that is in the form $J = I_n - \frac{1}{n}cc'$ where c is an $n \times 1$ column vector of all ones. Such matrix is a set of weights which column or row-wise sum is twice the mean of the number of entries n. This matrix has useful properties described in [120]. In particular, applied to $\tilde{D}$, it allows the determination of the centered matrix $B = -\frac{1}{2}J\tilde{D}^{(2)}J$, which is a transformation of $\tilde{D}$ around the mean positions of $\tilde{d}_{ij} \in \mathbb{R}^N$. This can be seen as follows:

$$\begin{aligned} -\frac{1}{2}J\tilde{D}^{(2)}J &= -\frac{1}{2}J(ec' + ce' - 2XX')J \\ &= -\frac{1}{2}Jec'J - \frac{1}{2}Jce'J + \frac{1}{2}J(2B)J \end{aligned} \tag{5.6}$$

By transposing J into the expression, it can easily be seen that, as $e'J = 0$ and $Je = 0$:

$$-\frac{1}{2}J\tilde{D}^{(2)}J = B \tag{5.7}$$

At this point, the distance matrix is centered. This phase has a particular importance for the method proposed in this chapter, because we will see below that its direct effect is to dilute the noise of one pair into the measurements reported by other pairs, thus causing mMDS to fail in highly asymmetric measurements like FTM. mMDS then computes the eigendecomposition of $B = Q\Lambda Q'$. Next, the experimenter has the possibility to decide of the dimensions of the projection space ($\mathbb{R}^2$ or $\mathbb{R}^3$ in our case, but the dimension can be any $m \in N$ in mMDS). This can be done by arbitrarily choosing the m largest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ and their corresponding eigenvectors of B. This choice is useful when the experimenter decides to project the distance matrix into m dimensions, and has decided of what value m should be. Alternatively, the experimenter can observe all eigenvalues in B and decide that the dimension space $\mathbb{R}^m$ matches all m large positive eigenvalues in B, ignoring the (comparatively) small positive eigenvalues, along with the null and negative eigenvalues as detailed in the previous section.

Then, if we write Λ_m the matrix of these m largest positive eigenvalues, and Q_m the first m columns in Q (thus the matrix of eigenvalues matching the dimensions decided by the experimenter), the coordinate matrix is determined to be $U = Q_m\Lambda_m^{1/2}$.

5.3 Problem Space Framework

5.3.1 mMDS Limitations in FTM Measurements

We note that mMDS is a family of techniques, with multiple possible variations, but all of them abide by the general core principles expressed in the previous section. These principles present properties of great value for multiple distance applications, but also three unsolvable limitations for indoor measurements like FTM: pairwise asymmetry is ignored, dimension space cannot be determined, and as the errors are centered, location accuracy is always typically poor. This section examines these issues in turn.

Pairwise Asymmetry

Although FTM does build a square distance matrix $\tilde{D}$, the FTM distance matrix displays several specific properties. One such property is asymmetry. In most cases, measurements are bidirectional, one side acting as the initiator, the other as the responder. At the conclusion of one exchange the roles are inverted. The process repeats for each sensor pair. These exchanges are affected by localities. As the measurements are collected in a noisy environment, they suffer from the side effects of multipath. Although most sensors attempt to determine the first signal to establish the line of sight (and therefore shortest distance), it is common that strong multipath may drown that signal to a point where the receiver identifies the first strong reflection as the first signal. In an environment where the position of reflective surfaces cannot be predicted, such effect may affect differently each receiver. As a consequence, the initial distance matrix set is asymmetric, where, for many pairs, $\tilde{d}_{ij} \neq \tilde{d}_{ji}$.

If obstacles or reflection sources are static (*e.g.*, walls), such asymmetry cannot be resolved by continuous measurements or by averaging over a long period (asymmetry persists). In an implementation where FTM sensors are calibrated (*i.e.* where the computed distance in line of sight is close to, but never less than, the ground truth), a simple corrective measure consists to only consider the smaller value for each reported pair distance, thus allowing $\tilde{d}_{ij} = \tilde{d}_{ji} = \min_{(i,j)}\{\tilde{d}_{ij}, \tilde{d}_{ji}\}$.

Dimension Determination

This approximation solves single pair distance inequalities, but indoor measurements surface another type of asymmetry. As illustrated in Figure 5.2, indoor walls are local objects. As the signal passes through a wall, it can be slowed down enough to induce an increased travel time between the ISTA and the RSTA, and thus a measured distance $\tilde{d}_{ij}$ dilated by a factor k_{ij} (compared to the ground truth d_{ij}) different than the dilation for another pair, between which

a wall would not be found, and thus, for 3 APs i, j and l:

$$\frac{\tilde{d}_{ij}}{d_{ij}} \neq \frac{\tilde{d}_{jl}}{d_{jl}} \neq \frac{\tilde{d}_{il}}{d_{il}} \tag{5.8}$$

Algebraically, a similar dilation factor k for all measured distances would be ideal and would make the dilation easy to resolve. Each measured distance would then be a simple (dilated) single scalar representation of the ground value matrix. As:

$$\tilde{D} = \begin{bmatrix} \tilde{d}_{11} & \dots & \tilde{d}_{1n} \\ \dots & \dots & \dots \\ \tilde{d}_{n1} & \dots & \tilde{d}_{nn} \end{bmatrix} = \begin{bmatrix} kd_{11} & \dots & kd_{1n} \\ \dots & \dots & \dots \\ kd_{n1} & \dots & kd_{nn} \end{bmatrix} = k \begin{bmatrix} d_{11} & \dots & kd_{1n} \\ \dots & \dots & \dots \\ d_{n1} & \dots & d_{nn} \end{bmatrix} = kD \tag{5.9}$$

Then $\tilde{D}$ would be a simple homothetic transformation of D, and our only task would be to find the scalar k, the dilation factor. But in real measurements, noise is not linear and each $\tilde{d}_{ij}$ is affected by a different dilation. Finding individual k_{ij} values becomes challenging, even if we reduce the scope by half by making the hypothesis that $d_{ij} = d_{ji} = \min\{d_{ij}, d_{ji}\}$ as above (thus considering that the smallest distance is likely closest to the true LoS distance), and thus that $k_{ij} = k_{ji}$. This geometry of the indoor space results in two additional major difficulties when attempting to use mMDS.

The first difficulty relates to dimension definition. FTM measurements are collected either in 2 dimensions (all APs or sensors at the same level, *e.g.*, at ground or ceiling level on the same floor) or 3 dimensions (multi-floor scenario). One task is therefore to determine the dimensions of the collection space, expecting 2 or 3 as a result. However, the interference locality expressed before results in different inaccuracies between measurements. This problem is difficult to resolve. A common MDS approach in this case is to let the experimenter decide of the dimensions, then use the least square error technique to project the matrix in the assigned $\mathbb{R}^n$ space. But this approach is only valid if the experimenter knows *a priori* the dimension n. In an indoor setting where RF travels in all directions and where distances are only measured by time of flight, such an arbitrary decision puts an unacceptable burden of knowledge on the FTM experimenter. Without knowing *a priori* k_{ij}, the determination of the vertical component of a set of sensors is only possible if k_{ij} happens to be small relative to d_{ij}.

This constraint can easily be understood with the geometric representation of an example. Suppose 4 sensors. A plane is defined by three non-collinear points. As such, any set of 3 sensors x_1, x_2 and x_3 will define a plane u, v and their distance matrix will surface 2 positive eigenvalues. Adding a fourth sensor x_4 introduces two possibilities: the additional sensor will either be on the plane (or close to it), or away from the plane (thus indicating $n = 2$ or $n = 3$, and thus a three dimensional space u, v, w) as shown in Figure 5.1.

Thus, if we note d_{ij} the distance between points i and j, u_i the component of point x_i along

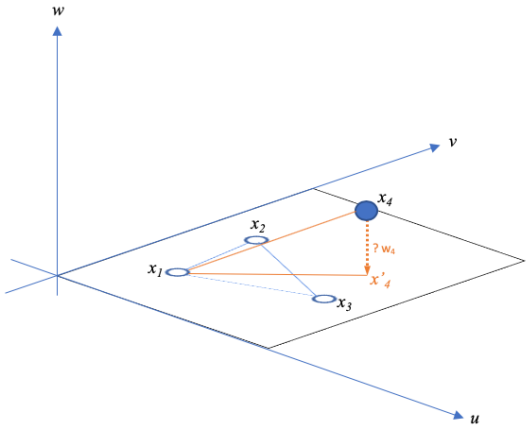


Figure 5.1 – Plane determination from AP sub-sets.

the u axis, and v_i and w_i its components along the v and w axis respectively, it is possible that $w_4 = w_1 = w_2 = w_3 = 0$, bringing x_4 to the plane defined by x_1, x_2 and x_3. Quite naturally, in a multi-storey building, it could also happen that x_1, x_2 and x_3 are on different floors, forming a plane that is not parallel to the ground (and x_4 could be yet on a third floor). However, as will be seen below, this scenario does not apply in the case of FTM, as the signal from one sensor would not be detected across two floors.

However, FTM does not provide $d_{12}, d_{13}, d_{14}, d_{23}, d_{24}$ and d_{34}, but $\tilde{d}_{12}, \tilde{d}_{13}, \tilde{d}_{14}, \tilde{d}_{23}, \tilde{d}_{24}$ and $\tilde{d}_{34}$. Determining the dimension of the linear span of $\{x_1, x_2, x_3, x_4\}$ is congruent to computing w_4. As $\tilde{d}_{14} = k_{14}d_{14}$, and therefore, considering a evaluated distance matrix reduced to the pair (x_1, x_4), then the matrix B_{14} obtained after double centering becomes:

$$B_{14} = -\frac{1}{2}J\tilde{D}_{14}^2 = -\frac{1}{2}k_{14}JD_{14}^2 \tag{5.10}$$

And the simple resolution to U shows that:

$$\begin{aligned} \tilde{d}_{14} &= \|k_{14}d_{14}\|_2 \\ &= k_{14}\|d_{14}\|_2 \\ &= \sqrt{(k_{14}u_4 - k_{14}u_1)^2 + (k_{14}v_4 - k_{14}v_1)^2 + (k_{14}w_4 - k_{14}w_1)^2} \end{aligned} \tag{5.11}$$

If $w_1 = 0$ to inscribe x_1 onto the plane defined by x_1, x_2, x_3, then:

$$k_{14}^2 w_4^2 = k_{14}^2((d_{14})^2 - (u_4 - u_1)^2 - (v_4 - v_1)^2) \tag{5.12}$$

And naturally:

$$w_4^2 = k_{14}^2 d_{14}^2 - d_{14}'^2 = d_{14}'^2(k-1) \tag{5.13}$$

Even if $x_1, x_2, x_3, x_4 \in \mathbb{R}^2$, any imprecision in the measurement of $\tilde{d}_{14}$ generates a dilation k_{14} that is directly translated into a non-null w_4 proportional to $k_{14} \geq 1$. In an environment where vertical separation is often considerably less than horizontal separation, an unknown pairwise k_{ij} soon prevents the resolution of $N \in \mathbb{R}^N$. For example, in a deployment where APs or sensors are positioned every 20 to 25 meters and floors (including slabs and isolation) are 4 meters apart, a measurement of $\tilde{d}_{14} = 25$ prevents the experimenter from determining if $d_{14} = 25$ and $w_4 = 0$ or if $20 \leq d_{14} \leq 24.68$ and $4 \leq w_4 \leq 15$, if k_{14} cannot deterministically be determined to be 1.2 or lower.

In the case of Wi-Fi ranging, this issue is easily solved with Received Signal Strength Indicator (RSSI) evaluations and LoS path loss equations. But we postulate that time of flight techniques alone cannot solve this issue unless k is consistent across all d_{ij} and estimated precisely. As will be demonstrated below, such quantization and uniformization of k is possible, but the experimenter cannot know if the dilation is due to a wall or the material separating two floors.

mMDS does introduce the idea of different dilation values with several possible techniques, commonly centered around the idea of defining a cost function, often called the strain or stress function. Such function is defined for example in [121] as

$$\sigma(D) = \sum_{i \neq j \neq k=1}^{n} w_{ijk}(\epsilon_{ijk} - \tilde{d}_{ij})^2$$

where ϵ_{ijk} is an ideal distance set between the points i, j and k, given the observed measurement $\tilde{d}_{ij}$, and w_{ijk} is a weight used to express the relative importance or precision of a given measurement $\tilde{d}_{ij}$. This relative importance does not aim at resolving dilation asymmetries. In fact, the resolution to U can take several iterative forms around mechanisms all aiming at minimizing the cost function $\sigma(D)$. In all cases, the choice of w_{ijk} is critical, as it can be different for each $\tilde{d}_{ij}$. However, in most cases, w_{ijk} is modulated based either on confidence indices (w_{ijk} is high for $\tilde{d}_{ij}$ known to be close to the ground truth) or on point criticality (w_{ijk} is high for i and j being anchors important to the problem to solve, *e.g.*, connection points to other networks *etc.*), not on dilation asymmetries. Other MDS techniques proceed with similar logic, with the general idea that distances may be noisy, but the noise being unknown, it can be considered as Gaussian, *i.e.* symmetric in most directions. Regardless of the method chosen, both the minimization and the double centering techniques tend, as noted in [122], to center the points along the mean of the error, and therefore to center the error. However, in the case of FTM (and probably multiple other time-of-flight-based distance estimation techniques), the noise is not Gaussian throughout the matrix.

Error Averaging

This last point causes the third additional difficulty. mMDS only considers the positive eigenvalues. This is a necessary requirement of the distance-to-coordinate resolution process, which needs the coordinate U to be derived from a centered matrix B eigendecomposition ($B = Q\Lambda Q'$ described in the previous section). Through this process, the coordinate matrix U is found as $U = Q\Lambda^{1/2}$. To maintain Λ in $\mathbb{R}$ (*i.e.* ensure that such matrix does not have a complex part), and working on the principle that the number of eigenvalues reflects the object dimension, $(\Lambda^{1/2})^2$ needs to have an expression solely in $\mathbb{R}$ (*i.e.* no complex part). This requirement means that negative eigenvalues are ignored.

However, as displayed in Figure 5.1, real world FTM measurements typically produce through the mMDS process multiple large positive eigenvalues, but also, sometimes large, negative eigenvalues. As demonstrated in [122], B is an approximation of XX', because it is built from $\tilde{D}$, not from D. As a consequence, writing $R = [r_{ij}] = B - XX'$ and $||R||_F^2$ the square of R Frobenius norm, and $tr(R)$ its trace, [122] shows that the resulting coordinates estimation error can be found to be:

$$L(X) = 2n\sum_{i=1}^{n} r_{ii}^2 + 2(tr(R))^2 + 4||R||_F^2 \tag{5.14}$$

This result shows that the error increases as the difference between d_{ij} and $\tilde{d}_{ij}$ increases. It is also easily seen that the mMDS process centers the error. This issue can easily be seen through a simple example. Suppose a simple measured distance matrix with 3 points, each distance being affected by a specific k_{ij} dilation factor:

$$\tilde{d}_3 = \begin{bmatrix} 0 & k_{12}d_{12} & k_{13}d_{13} \\ k_{12}d_{12} & 0 & k_{23}d_{23} \\ k_{13}d_{13} & k_{23}d_{23} & 0 \end{bmatrix} \tag{5.15}$$

In this simple configuration:

$$J = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{3}{4} & -\frac{1}{4} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{3}{4} \end{bmatrix} \tag{5.16}$$

And thus:

$$B = -\frac{1}{16}\begin{bmatrix} -3k_{12}^2d_{12}^2 - 3k_{13}^2d_{13}^2 + k_{23}^2d_{23}^2 & 5k_{12}^2d_{12}^2 - k_{13}^2d_{13}^2 - k_{23}^2d_{23}^2 & -k_{12}^2d_{12}^2 + 5k_{13}^2d_{13}^2 - k_{23}^2d_{23}^2 \\ 5k_{12}^2d_{12}^2 - k_{13}^2d_{13}^2 - k_{23}^2d_{23}^2 & -3k_{12}^2d_{12}^2 + k_{13}^2d_{13}^2 - 3k_{23}^2d_{23}^2 & -k_{12}^2d_{12}^2 - k_{13}^2d_{13}^2 + 5k_{23}^2d_{23}^2 \\ -k_{12}^2d_{12}^2 + 5k_{13}^2d_{13}^2 - k_{23}^2d_{23}^2 & -k_{12}^2d_{12}^2 - k_{13}^2d_{13}^2 + 5k_{23}^2d_{23}^2 & k_{12}^2d_{12}^2 - 3k_{13}^2d_{13}^2 - 3k_{23}^2d_{23}^2 \end{bmatrix} \tag{5.17}$$

From B, it can clearly be seen that dilation factors that affect a single distance pair in $\tilde{D}$ are projected as weights for the computation of the centered matrix B, affecting all other entries of B in the process. The weights, because of the structure of J, affect differently the various pairs. The sum of the contribution of each dilation factor k_{ij} in B is now 1 for each row or column where the factor was present in $\tilde{D}$ and -1 for each row or column where the factor was not present. Overall, the contribution in B is reflective of the overall contribution in $\tilde{D}$. However, centering the matrix also has the effect of distributing the dilation factors, and thus distributing individual dilation factors.

The outcome of such distribution is that a single large dilation factor will then affect the computed position of all points in U, thus distributing the error to all positions. In a distance setting where the noise is Gaussian across all sensors, this distribution has only minor effects. However, in settings, like indoor localization with FTM, where noise asymmetry is present, and where accuracy depends on identifying the RF interference localities and the associated dilation differences between pairs, mMDS, used alone, can provide an acceptable result, but that will always be worse than quality LoS measurements. As a consequence, highly dilated distance pairs get hidden in the transformation, and precise measurements get degraded by the contribution of dilated pairs.

Therefore, there is a need for a method that can identify and compensate for the highly dilated segments, to attempt to reduce their dilation before they are injected in a method, such as mMDS, where all segment contributions are treated equally.

5.4 Materials and Methods

5.4.1 First Component: Wall Remover - Minimization of Asymmetric Errors

One first contribution of this chapter is a method to reduce dilation asymmetry. Space dimension is resolved using other techniques (*e.g.*, RSSI-based), and Section 5.5 will provide an example. Once the dimension space has been reduced to $\mathbb{R}^2$ and only sensors on the same floor, we want to reduce the dilation asymmetry. Figure 5.2 illustrates a typical asymmetry scenario. In this simplified representation, a strong obstacle appears between the sensor x_1 and sensors x_6, x_7 and x_8, causing the distances $d(x_6, x_1)$, $d(x_7, x_1)$ and $d(x_8, x_1)$ to be appear as $d(x_6, x_1')$, $d(x_7, x_1'')$ and $d(x_8, x_1''')$ respectively. At the same time, supposing that the system is calibrated properly and LoS conditions exist elsewhere, the obstacle does not affect distance measurements between x_1 and x_2, x_3, x_4, x_5 and x_9, that are all approximated along the same linear dilation factor k.

In such case, using a centering technique to average the error may make sense algebraically, but not geometrically. Finding a method to reduce the error while detecting its directionality is therefore highly desirable. Luckily, geometry provides great methods to this mean, that only

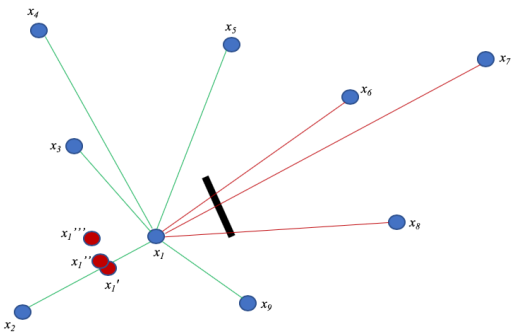


Figure 5.2 – Directionality of measurement errors.

have the inconvenience of requiring multiple comparisons. Such process may be difficult when performing individual and mobile station ranging, but becomes accessible when all sensors are static. When implemented in a learning machine, this method can be implemented at scale and reduce the distance error only when it is directional, thus outputting a matrix $\tilde{D}$ which k factor is closer to uniformity. It should be noted that the purpose of such method should not be to fully solve the MDS problem, as some pair-distances are usually not known, and the method has a limited scope (*i.e.* it cannot assess some of the pairs, as will be seen below). However, in many cases, sensors can be found that display interesting properties displayed on the left part of Figure 5.3. In this scenario, 3 sensors x_1, x_2 and x_3 are selected that form a triangle. The triangle represented in the left part of Figure 5.3 is scalene, but the same principle applies to any triangle. A fourth sensor x_4 is found which distance to x_1, x_2 and x_3 is less than d_{12}, d_{13} or d_{23}, thus placing x_4 within the triangle formed by (x_1, x_2, x_3).

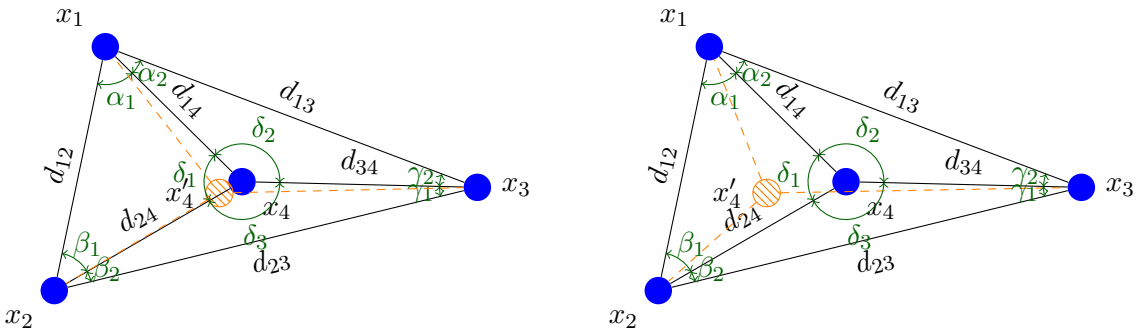
A natural property of such configuration is that $\delta_1 + \delta_2 + \delta_3 = 2\pi$ and, in any of the triangles $(x_1, x_2, x_4), (x_1, x_3, x_4)$ or (x_2, x_3, x_4), one side can be expressed as a combination of the other two and of its opposite angle. For example, d_{24} in (x_1, x_2, x_4) can be expressed as:

$$d_{24}^2 = d_{12}^2 + d_{14}^2 - 2d_{12}d_{14}cos(\alpha_1) \tag{5.18}$$

The above easily allows us to find d_{24} and can also be used to determine angles from known distances, for example δ_1, knowing that:

$$cos(\delta_1) = \frac{d_{14}^2 + d_{24}^2 - d_{12}^2}{2d_{14}d_{24}} \tag{5.19}$$

Therefore, angles and missing distances can be found from known distances. In an ideal world, properties (5.18) and (5.19) are verified for each observed triangle ((x_1, x_2, x_3), (x_1, x_2, x_4), (x_2, x_3, x_4) and (x_1, x_3, x_4)). In a noisy measurement scenario, inconsistencies are found. For example, an evaluation of the triangle (x_1, x_2, x_4) may be consistent with the left side of Figure 5.3,



(a) Small computed range vs. ground truth error (b) Large ranging errors due to obstacles and multipath

Figure 5.3 – Sensor location accuracy evaluation through geometric relationship.

but an evaluation of triangle (x_1, x_3, x_4) may position x_4 in the hashed representation x'_4 of the right part of Figure 5.3. It would be algebraically tempting to resolve x_4 as the middle point between both possibilities (mean error). However, this is not the best solution. In Figure 5.3 simplified example, the most probable reason for the inconsistency is the presence of an obstacle or reflection source between x_3 and x_4. If all distances are approximations, some of them being close to the ground truth and some of them displaying a larger k factor, a congruent representation of such asymmetry is that k is larger for x_4x_3 than it is for the other segments. Quite obviously, other possibilities can be found. For example, in the individual triangle (x_1, x_2, x_4), it is possible that k is larger for the segments d_{12} than it is for segments d_{24} and d_{14}. However, as x_4x_3 is compared to other segments and the same anomaly repeats, the probability that the cause is an excessive stretch on x_4x_3 increases.

As the same measurements are performed for more points, the same type of inconsistency appears for other segments. Thus an efficient resolution method is to identify these inconsistencies, determine that the distance surfaced for the affected segment is larger than a LoS measurement would estimate, then attempt to individually and progressively reduce the distance (by a local contraction factor that we call $0 < \zeta_{ij} \leq 1$), until inconsistencies are reduced to an acceptable level (that we call γ).

Thus, formally, a learning machine that we call a geometric Wall Remover engine, is fed with all possible individual distances in the matrix, and compares all possible iterations of sensors forming a triangle and also containing another sensor. Distance matrices do not have orientation properties. When considering (x_1, x_4, x_3) for example, x_4 can be found on either side of segment x_1x_3, and the solution can also be a triangle in any orientation. However, adding a fourth point (x_2), which distance to x_4 is evaluated, can constrain x_4 within the (x_1, x_2, x_3) triangle. As the evaluation proceeds iteratively throughout all possible triangles that can be formed from the distance matrix, the system first learns to place the sensors relative to each other. The resulting sensor constellation can have globally any orientation, but the contributing points partially

appear in their correct relative position.

Algorithm 1: Wall Remover Algorithm

Input : ϵ: learning rate
γ: acceptable error range
Output: optimized d_{ij}

// convert $\tilde{D} \in \mathbb{R}^2$ to pairwise distance list d_{ij}
$d_{ij}[i, j, dist_{ij}] \leftarrow dist2list(\tilde{D})$;
// set a list of same length ζij of all ones
// set a list of same length w_{ij} of all ones
$d_{ij}[i, j, dist_{ij}, \zeta ij, w_{ij}] := \{d_{ij}, \zeta ij, w_{ij}\}$;
while *true* **do**
  reverse_sort d_{ij};
  for $d_{ij} \geqslant d_{ik} > d_{jk} > 0$ **do**
    find $l : d_{il}, djl, d_{kl} \neq 0$;
    compute $\widehat{ijk}, \widehat{jki}, \widehat{kij}, \widehat{ijl}, \widehat{jkl}, \widehat{kil}$;
    if $\widehat{ijl} > \widehat{ijk} \vee \widehat{jkl} > \widehat{jki} \vee \widehat{kil} > \widehat{kij}$ **then**
      find next l (l is outside (i,j,k));
    if $d_{il}/\sqrt{d_{ij}^2 + d_{jl}^2 - 2d_{ij}d_{jl}cos(\widehat{ijk} - \widehat{kjl})} > 1$ **then**
      $w_{il} := w_{il} + (d_{il}/\sqrt{d_{ij}^2 + d_{jl}^2 - 2d_{ij}d_{jl}\cos(\widehat{ijk} - \widehat{kjl})} - 1)$;
    repeat along d_{jl} and d_{kl};
    next l;
  foreach d_{ij} **do**
    $p_{ij} := 1/(1 + e^{w_{ij}})$; // compute stretch probability:
  reverse_sort p_{ij};
  $tp_{ij} \leftarrow$ **pick_top** p_{ij};
  if $tp_{ij} < \gamma$ **then**
    break;
  $d_{ij} := d_{ij}\epsilon$;
  $\zeta_{ij} := \zeta_{ij} - \epsilon$;

Then, each time a scenario matching the right side of Figure 5.3 is found, the algorithm learns the asymmetry and increases the weight w of the probability p that the matching segment (x_3x_4 in this example) has a dilated k_{ij} factor (*cf.* Algorithm 1). As a segment may be evaluated against many others, its w may accordingly increase several times at each run. The algorithm starts by evaluating the largest found triangles first (sorted distances from large to small), because they are the most likely to be edge sensors. This order is important, because in Figure 5.3 example, a stretched x_2x_4 value may cause x_4 to be graphed on the right side of segment x_1x_3, and thus outside of (x_1, x_2, x_3). This risk is mitigated if multiple other triangles within (x_1, x_2, x_3) can also be evaluated. An example is depicted by Figure 5.4. In this simplified representation (not all segments are marked), the position of x_4 is constrained by first evaluating $(x_5, x_4, x_7), (x_5, x_4, x_{10}), (x_{10}, x_4, x_{12})$ and (x_7, x_4, x_{12}) against

$(x_5, x_{10}, x_7), (x_5, x_{10}, x_{12}), (x_5, x_7, x_{12})$ and (x_{10}, x_7, x_{12}). This evaluation allows the system to surface the high probability of the stretch of segment x_4x_7, suggesting that x_4 should be closer to x_7 than the measured distance $\tilde{d}_{x_4x_7}$ suggests, but not to the point of being on the right side of x_1x_3. The system can similarly detect a stretch between points x_3 and x_8 (but not between x_7 and x_8 or x_{10} and x_3).

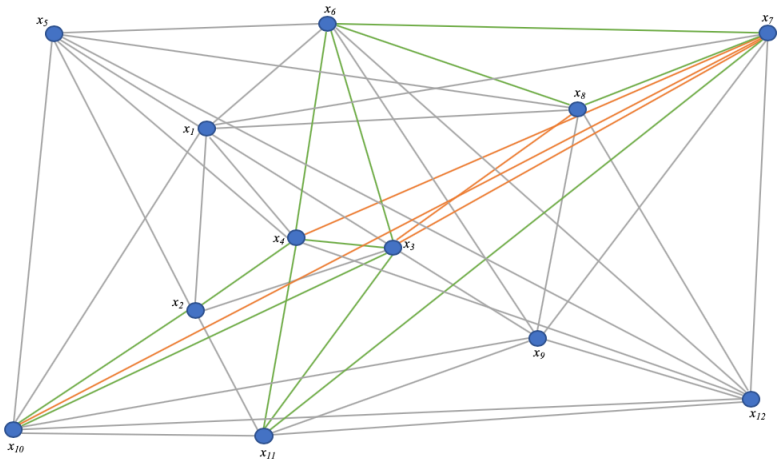


Figure 5.4 – Stretch detection between x_3 and x_8.

At the end of the first training iteration, the system outputs a sorted list of segments with the largest stretch probabilities (largest w and therefore largest p). The system then picks that segment with that largest stretch, and attempts to reduce its stretch by proposing a contraction of the distance by an individual ζ_{ij} factor. The ζij factor can be a step increment (similar to other machine learning algorithms *learning rate* logic) or can be proportional to the stretch probability. After applying the contraction, the system runs the next iteration, evaluating progressively from outside in, each possible triangle combination. The system then recomputes the stretch probabilities and proposes the next contraction cycle. In other words, by examining all possible triangle combinations, the system learns stretched segments and attempts to reduce the stretch by progressively applying contractions until inner neighboring angles become as coherent as possible, *i.e.* until the largest stretch is within a (configurable) acceptable range γ from the others.

This method has the merit of surfacing points internal to the constellation that display large k factors, but is also limited in scope and intent. In particular, it cannot determine large k factors for outer segments, as the matching points cannot be inserted within triangles formed by other points. However, its purpose is to limit the effect on measurements of asymmetric obstacles or sources of reflection.

5.4.2 Iterative algebro-geometric EDM Resolution

The output of the wall remover method is a matrix with lower variability to the dilation factor k, but the method does not provide a solution for an incomplete and noisy EDM. Such reduction still limits the asymmetry of the noise, which will therefore also limit the error and its locality when resolving the EDM, as will be shown below. Noise reduction can be used on its own as a preparatory step to classical EDM resolution techniques. It can also be used in combination with the iterative method we propose in this section, although the iterative method has the advantage of also surfacing dilation asymmetries, and thus could be used directly (without prior dilation reduction). Combined together, these two techniques provide better result than standard EDM techniques.

EDM resolution can borrow from many techniques, which address two contiguous but discrete problems: matrix completion and matrix resolution. In most cases, the measured distance matrix $\tilde{D}$ has missing entries, indicating unobserved distances. In the case of FTM, these missing distances represent out-of-range pairs (*e.g.*, sensors at opposite positions on a floor or separated by a strong obstacle, and that cannot detect each other). A first task is to complete the matrix, by estimating these missing distances. Once the matrix contains non-zero numerical values (except for the diagonal, which is always the distance d_{ii} and therefore always 0), the next task is to reconcile the inconsistencies and find the best possible distance combination.

Several methods solve both problems with the same algorithm and [123] provides a description of the most popular implementations. We propose a geometric method, which uses partial matrix resolution as a way to project sensor positions geometrically onto a $\mathbb{R}^2$ plane, then a mean cluster error method to identify individual points in individual sets that display large asymmetric distortions (and should therefore be voted out from the matrix reconstruction). By iteratively attempting to determine and graph the position of all possible matrices for which point distances are available, then by discarding the poor (point pairs, matrices) performers and recomputing positions without them, then by finding the position of the resulting position clusters, the system reduces asymmetries and computes the most likely position for each sensor.

Formally, the measured distance matrix $\tilde{D}$ of n sensor distances is separated in sub-matrices. Each sub-matrix $\tilde{d}_m$ contains $2 \leq m \leq n$ points for which inter-distances were measured, so that:

$$\forall 2 \leq m \leq n \text{ and } \forall i, j \in \tilde{d}_m, \tilde{d}_{ij} > 0 \tag{5.20}$$

We want to graph the position of each point in the sub-matrix. A pivot x_1 is chosen iteratively in $\tilde{d}_m$. In the first iteration, $x_1 = i_1$, then $x_1 = i_2$ in the second iteration, and $x_1 = i_m$ in the last iteration. For each iteration, x_1 is set as the origin, and $x_1 = (0, 0)$. An example is displayed in Figure 5.5. The next point x_2 in the matrix (i_2) is iteratively set along the x-axis, and $x_2 = (\tilde{d}_{x_1 x_2}, 0)$. If $m > 2$, then the position of each other point x_i of the set $\{x_1, x_2, x_i\}$ is found

using standard triangular formula illustrated by the points in Figure 5.5 and where:

$$u_{x_i} = \frac{x_i x_2^2 - x_1 x_i^2 - x_1 x_2^2}{-2x_1 x_2}$$
$$\text{and} \tag{5.21}$$
$$v_{x_i} = \sqrt{x_1 x_i^2 - u_{x_i}^2}$$

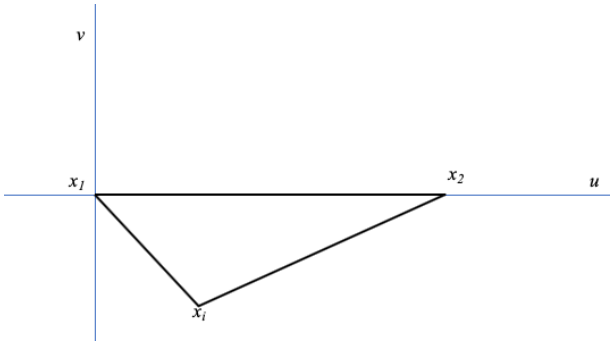


Figure 5.5 – Geometric representation of a set of 3 points in a submatrix.

Such formula fixes the position of x_i above the u-axis (because v_{x_i} is always positive). This may be x_i's correct position in the sensor constellation in some cases, but can also result in an incorrect representation as soon as the next point x_j is introduced in the graph. A simple determination of the respective positions of x_i and x_j can be made by evaluating their distances, as in the wall remover method. In short, if $d_{x_i x_j} > v_{x_i}$, then x_i and x_j are on opposite sides of the u-axis and v_{c_j} becomes $-v_{c_j}$.

As the process completes within the first matrix, the position of m points are determined using the first pair of points x_1 and x_2 as the reference. In the next iteration, x_1 is kept but x_2 changes from i_2 to i_3. Algebraically, the first iteration determines the positions based on the pairwise distances expressed in the first 2 columns of the distance matrix, while the second iteration determines the positions based on the pairwise distances expressed in the first and third columns. Both results overlap for the first 3 points x_1, x_2 and x_3, but not for the subsequent points. This can be easily understood with an example. Recall that for any pair of points i and j, $d_{ij} = d_{ji}$. Using d_{ij} as a representation of both d_{ij} and d_{ji}, and the following small matrix of 5 points as an example:

$$\tilde{D}_5 = \begin{bmatrix} 0 & \tilde{d}_{12} & \tilde{d}_{13} & \tilde{d}_{14} & \tilde{d}_{15} \\ \tilde{d}_{12} & 0 & \tilde{d}_{23} & \tilde{d}_{24} & \tilde{d}_{25} \\ \tilde{d}_{13} & \tilde{d}_{23} & 0 & \tilde{d}_{34} & \tilde{d}_{35} \\ \tilde{d}_{14} & \tilde{d}_{24} & \tilde{d}_{34} & 0 & \tilde{d}_{45} \\ \tilde{d}_{15} & \tilde{d}_{25} & \tilde{d}_{35} & \tilde{d}_{45} & 0 \end{bmatrix} \tag{5.22}$$

The first iteration ignores $\tilde{d}_{34}$ and $\tilde{d}_{35}$ that are represented in the second iteration (but the second iteration does not represent $\tilde{d}_{24}$ or $\tilde{d}_{25}$).

As in the second iteration i_3 is used as a reference point for the x axis, the geometrical representations of the first and the second iterations are misaligned. However, x_1 is at the origin in both cases, and x_2 is represented in both graphs (we note them x_2 and x_2'). Using Equation (5.21), finding the angle (α) formed by the points $x_2x_1x_2'$ is straightforward, and projecting the second matrix into the same coordinate reference as the first matrix is a simple rotation of the second matrix, defined by T as:

$$T = \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{5.23}$$

In the subsequent iterations, x_1 ceases to be at the origin. Depending on the sub-matrix and the iteration, x_1 may or may not be in the new matrix. However, 2 points $i = (u_i, v_i)$ and $j = (u_j, v_j)$ can always be found that are common to both the previous matrix $\tilde{d}_p$ and the next matrix $\tilde{d}_n$. Projecting $\tilde{d}_n$ into the same coordinate reference as $\tilde{d}_p$ is here again trivial, by first moving the coordinates of each point found from $\tilde{d}_n$ by $proj(u_i) = u_{ip_n} - u_{ip_p}$ and $proj(v_i) = v_{ip_n} - v_{ip_p}$ then perform a rotation using Equation 5.23. These operations are conducted iteratively. As measured distances are noisy, for each point represented in $\tilde{D}$, different coordinates appear at the end of each iteration, thus surfacing for each point a cluster of computed coordinates. Figure 5.6 represents this outcome after 4 iterations over a 5-point matrix and $x_1(0, 0)$ fixed about the first point.

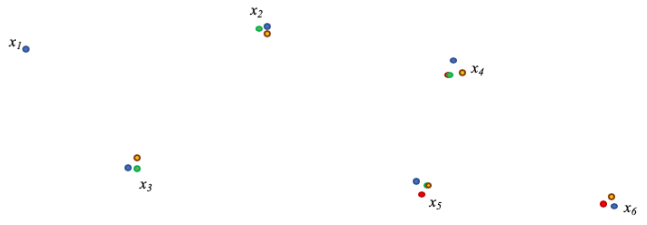


Figure 5.6 – Graphing points in a 5x5 matrix using the geometric approach.

The effect of projection and rotation makes that the red and green points are overlapping in x_4, and the green and orange point are overlapping in x_5. The choice to determine which points should be used to project $\tilde{d}_n$ onto $\tilde{d}_p$ is sequentially obvious, but arbitrary otherwise. In the example displayed in Figure 5.6, x_6 instead of x_2 could have been chosen to project iteration green onto iteration red *etc.* A reasonable approach could be to compute iteratively, another one to compute all possible combinations, a third approach is to compute the mean of the positions determined for a point as the best representation of the likely position of a given point for a given

iteration. The second method is obviously more computationally intensive, but provides a higher precision in the final outcome. As a cluster of positions appears for each point, representing the computed position of each sensor, asymmetries and anomalies can here as well be surfaced. All points associated with a sensor form a cluster, which center can be determined by a simple mean calculation, where for each cluster center $\mu_i = (x_{\mu_i}, y_{\mu_i})$ and m associated points:

$$u_{\mu_i} = \frac{\sum_{j=1}^{m} u_j}{m} \quad \text{and} \quad v_{\mu_i} = \frac{\sum_{j=1}^{m} v_j}{m} \tag{5.24}$$

Projections that are congruent will display points that are close to one another for a given cluster. However, the graph will also display some representations that display a large deviation. This deviation can be asymmetric. It is then caused by a dilation factor k different for a sensor pair than for the others. In most cases, strong obstacles or reflection sources may increase the dilation factor. The geometric wall remover method exposed in the previous section of this work is intended to reduce such effect. As the algorithm acts on the angles of adjacent triangles, it is more precise than this section of our proposed method. However, it may happen that the dilation occurs among pairs than the geometric wall remover cannot identify (for example because the pair is formed with sensors at the edge of the constellation).

The deviation also surfaces matrices coherence. A coherent matrix contains a set of distances displaying a similar dilation factor k. An incoherent matrix contains one or more distance displaying a k factor largely above or below the others. For example, several sensors may be separated from each other by walls, but be in LoS of a common sensor, which will display a k factor smaller than the others. Such sensor is an efficient anchor, *i.e.* an interesting point $x_1(0, 0)$ or $x_2(\tilde{d}_{x_1x_2}, 0)$ for the next iteration. On the other hand, some sensors may be positioned in a challenging location and display large inconsistencies when ranging against multiple other sensors. The position of these sensors needs to be estimated, but they are poor anchors for any iteration.

An additional step is therefore to identify good, medium and poor anchors and discard all distances that were computed using poor anchors. The same step can identify and remove outliers pairwise computed positions that deviate too widely from the other computed positions for the same points), and thus accelerate convergence. For each cluster i of m points for a given sensor, each at an Euclidian distance $x_i\mu_i$ from the cluster center μ_i, a mean distance to the cluster center can be expressed as:

$$r_i = \frac{\sum_{j=1}^{m} |x_j - \mu_i|}{m} \tag{5.25}$$

where r_i thus expresses the mean radius of the cluster. By comparing radii between clusters, points displaying large r_i values are surfaced. Different comparison techniques can be used for such comparison. We found efficient the straightforward method of using the 2σ rule, where a cluster which radius is more than 2 standard deviations larger than all clusters mean radii is highlighted as an outlier. The associated sensor therefore displays unusually large noise in its distance measurement to the other sensors, and is therefore a poor anchor. Matrices using this sensor as an anchor are removed from the batch and clusters are recomputed without these sensors' contribution as anchors. As the computation completes, each cluster center is used as the best estimate of the associated sensor position. By reducing the variance of the dilation factor k, by removing sub-matrices and sensor pairs that bring poor accuracy contribution, this method outperforms standard EDM completion methods when $\mathbb{R}^n$ is known, because it reduces asymmetries before computing positions, but also because the geometric method tends to rotate the asymmetries as multiple small 3×3 matrices are evaluated, thus centering the asymmetries around the sensor most likely position.

Algorithm 2: Iterative algebro-geometric EDM resolution algorithm

Input: $\tilde{D}_{m*m} \in \mathbb{R}^2$
for *i from 1 to* $m-1$*:* **do**
  generate all $\tilde{D}_{i*(i+1)}$ to $\tilde{D}_{i*(m-i)}$ matrices;
remove each $d_{ij} = 0$ from each matrix;
remove any $\tilde{D}_{1*1}$ matrix;
for *generated* $\tilde{D}_{n*n}$ **do**
  p=q=1;
  for $o = 1$ *to* n **do**
    plot $x_o = (0, 0)$;
    plot $x_{o+p} = (d_{op}, 0)$;
    $x_{o+q} = (u_{o+q}, v_{o+q})$;
    $u_{o+q} = (\frac{x_{o+q}x_{o+p}^2 - \ x_o x_{o+q}^2 - x_o x_{o+p}^2}{-2x_o x_{o+p}},)$;
    $v_{o+q} = (\sqrt{x_o x_{o+q}^2 - u_{x_{o+q}}^2})$;
    if $d_{x_{o+q-1}x_{o+q}} > v_{o+q}$ **then**
      $v_{o+q} := -v_{o+q}$;
    next q;
    next p, compute $\widehat{p_{-1}op}$ and rotate new plot onto previous plot coordinates;
    next o, compute $d_{o_{-1}o}$ and translate new coordinates into previous reference;
for *each point* **do**
  compute cluster center position;
for *each cluster* **do**
  compute σ;
for *all clusters* **do**
  compute average σ;
record this initial σ;
for *each cluster* **do**
  record any point 2σ or more from cluster center;
select farthest point of all;
if *point was an anchor (o or p)* **then**
  mark matching matrix as invalid;
else
  select next farthest point;
go back to plotting all remaining matrices;
when no anchor is outlier anymore, remove all non-anchor outliers;
compute cluster means for remaining points;

5.5 Results

5.5.1 Experiment Methodology

We tested this method in five different buildings already equipped with Wi-Fi APs providing active coverage. As the active APs do not support FTM, we position FTM devices near each active AP (typically 15 cm, of half a foot, from each active AP position). Building 1 is a three-storey office building with cubicle areas alternating with blocks of small offices. Our testbed is installed on the second floor. Building 1 is interesting, because the wall structure is irregular, causing different reflection and absorption patterns for each AP pair. The test floor is already equipped with 13 access points positioned at ceiling level. We therefore positioned 13 FTM stations at ceiling level, one near each AP, as represented in Figure 5.7.

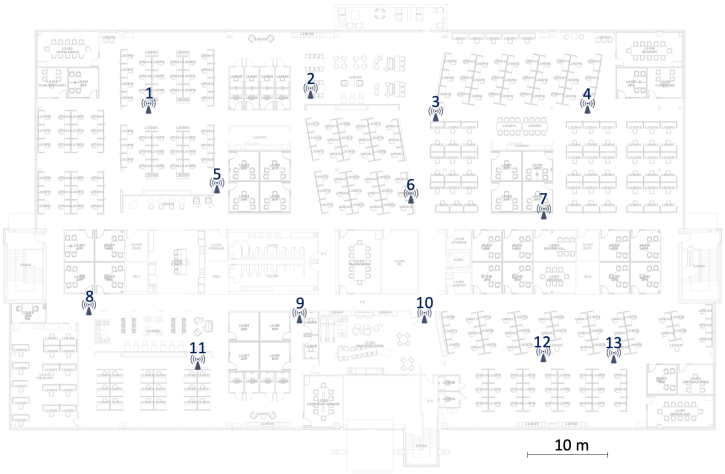


Figure 5.7 – Building 1 experimental setup.

Building 2 is also an office building with 30 APs, with denser sitting density than building 1 and where zones of open space alternate with strong obstacles. Similarly to building 1, one FTM device is positioned 15 cm (1/2 foot) away from each existing AP, as represented in Figure 5.8.

Building 3 is a large open space building with partial high ceilings, used as an enterprise restaurant. It presents high peaks of user and Wi-Fi traffic alternating with no activity. This traffic pattern allows us to evaluate the effect of traffic (or lack thereof) on ranging accuracy. Additionally, the building structure makes that AP height is not consistent throughout the floor, thus presenting an interesting geometry. Building 3 is represented in Figure 5.9. High-ceiling APs are represented by upside down icons. In this setting, all APs are the same model and use omnidirectional antennas. Thus, the only difference between APs is their height, 3.2 meters for standard ceiling heights, and 7.3 meters for high ceiling APs. Our FTM devices are positioned near each AP as in the previous buildings.

Building 4 is a warehouse, with metallic racks, high ceilings and some APs mounted with

Figure 5.8 – Building 2 experimental setup.

directional antennas. An office space is present in the upper right corner of the building. High ceiling APs use omnidirectional antennas and are positioned on poles hanging from the ceiling, between 6.4 and 8.1 meters above the ground. On the side of the aisles, some APs are mounted at 3.4 meters height and equipped with directional patch antennas, as represented in Figure 5.10. For our experiments, we use these antennas to evaluate the effect of directionality.

Building 5 is a training center with small to medium-sized classrooms. It presents an interesting denser wall structure (as walls separate the classrooms). As represented in Figure 5.11, each classroom has its own AP in the center. The dense (hashed) non-accessible technical area at the center makes that APs on one side do not detect APs on the other side.

For the main presentation of this section, we will focus on building 1 as the main example, and will highlight the other buildings when they present interesting variations. In all cases, our FTM devices are Linux systems equipped with Intel 8260 cards enabled for FTM. In the rest of this section, we will refer to AP-to-AP distances and measurements, although it should be clear that the FTM device associated to each AP was used to operate the measurement. Ground truth distances are known and confirmed with floorplan blueprint and onsite laser ranging. The FTM devices can be configured to act as ISTAs or RSTAs. The system is left active for two days (with the exception of the warehouse, where the system is left active for 6 hours, so as to minimize the disruptions to the warehouse operations). Every hour, the system wakes up, and each station is randomly affected the role of ISTA or RSTA. Each ISTA ranges against the detected RSTAs on various channels for 10 minutes and logs the result. At the end of the collection phase, the logs of all stations are collected and injected into the learning machine.

This duration was chosen to ensure that collection would happen both at quiet times (at night, with no user is in the building) as well as during busy day times. Although daytime mea-

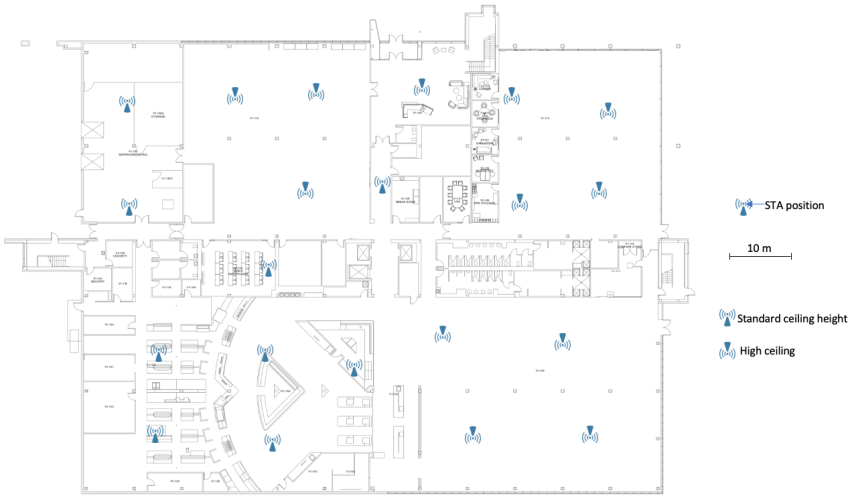


Figure 5.9 – Building 3 experimental setup.

surements do show larger collision and retry counts, no significant difference could be observed in ranging accuracy. It is therefore likely that the system could only be left active for a shorter duration and yet obtain convergence. Mobile stations are also walked around the building, to determine locations where ranging to indoor APs would be possible. From left entrance and stairs, APs 01, 08 and 11 are reachable. From the main entrance (bottom), APs 11, 09, 10 12 and 13 are reachable. From the right entrance and stairs, APs 13, 07 and 04 are reachable as can be seen in Figure 5.12.

The purpose of the outdoor mobile stations is to serve as a seed for an initial GPS position communicated to the APs in range of the station. One key determinator is the accuracy of the GPS value seeded by the mobile station. GPS accuracy utilities are installed on different phones. As GPS position is computed, the operator clicks the real location, based on high resolution aerial maps, and the system outputs the error of the computed GPS estimation. As displayed in Figure 5.13, the error is commonly within half a meter. Inside the building, accuracy collapses (not represented here).

An examination of the distance matrix for all APs (measured through their associated FTM devices) shows high noise. Figure 5.1 displays the ground truth (real distances) and Figure 5.2 the distances obtained via FTM measurements, after applying the $d_{ij} = d_{ji} = min\{d_{ij}, d_{ji}\}$ simplification. The ground truth 13×13 matrix surfaces 9 positive eigenvalues, because measurements suffered from natural scale and rounding inaccuracies. However, only 2 of these eigenvalues are large, indicating a 2-dimensional geometrical object. The measured FTM matrix, on the other hand, surfaces 'only' 7 positive eigenvalues, but all of them are large. The first two eigenvalues are larger by a factor of 10 compared to the others, but this factor is not sufficient to discount the other 5 large positive eigenvalues.

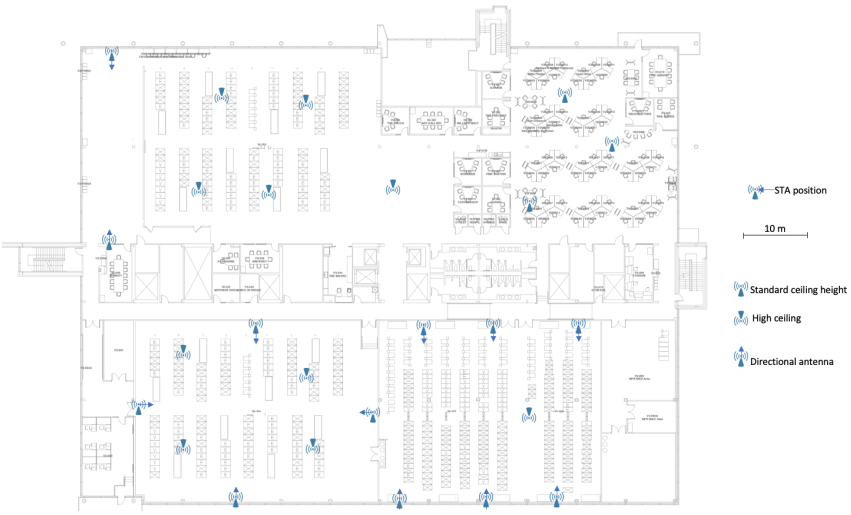


Figure 5.10 – Building 4 experimental setup.

Device	1	2	3	4	5	6	7	8	9	10	11	12	13	eigenvalues
1	0	2086	4183	5632	1430	3655	5262	2452	3275	4387	3295	5564	6702	$5.6797e^{+7}$
2	2086	0	2135	3581	1704	1999	3414	3734	2895	3187	3722	4242	5150	$2.1831e^{+7}$
3	4183	2135	0	1470	3362	1312	1564	5278	3405	2582	4672	3137	3579	42674
4	5632	3581	1470	0	4789	2475	1442	6650	4543	3310	5875	3352	3221	34379
5	1430	1704	3362	4789	0	2507	4135	2045	1875	3007	2200	4182	5375	28945
6	3655	1999	1312	2475	2507	0	1623	4218	2118	1407	3422	2275	3156	27480
7	5262	3414	1564	1442	4135	1623	0	5749	3404	1978	4750	1920	2033	21641
8	2452	3734	5278	6650	2045	4218	5749	0	2504	4118	1530	5172	6490	5206.5
9	3275	2895	3405	4543	1875	2118	3404	2504	0	1605	1345	2670	3987	5.8021^{-9}
10	4387	3187	2582	3310	3007	1407	1978	4118	1605	0	2892	1175	2415	-10435
11	3295	3722	4672	5875	2200	3422	4750	1530	1345	2892	0	3825	5153	-11658
12	5564	4242	3137	3352	4182	2275	1920	5172	2670	1175	3825	0	1318	-46289
13	6702	5150	3579	3221	5375	3156	2033	6490	3987	2415	5153	1318	0	$-1.1762e^{+5}$

Table 5.1 – Building 1 - Matrix of ground truth distances and related eigenvalues

The measured FTM matrix generated from the other buildings present the same type of difficulty. Building 2 (also office building, but with 30 APs), the matrix surfaces 9 large eigenvalues. In building 3 (restaurant with partial high ceiling), 6 large positive eigenvalues are seen, obfuscating the fact that the deployment is three-dimensional. In building 4 (partial high ceiling warehouse), 11 large engeinvalues are seen. In building 5, 3 large eigenvalues are seen (while all the APs are on the same plane). This last observation may be caused by the strong obstacle in the center of the building, which reduces the number of measurable distances (although 14 APs are deployed, no AP/ FTM device has measurable distance to more than 8 other APs (FTM devices); *i.e.* throughout the matrix each AP has no measured distance to 8 to 11 APs). As such, it is clear that the distance matrix alone is not sufficient to assert the dimensionality of the space. However, complementing with RSSI evaluation easily solves the issue. In building 1, another AP (AP15) is positioned on the upper floor, above AP05, and another (AP16) above AP06. As the ISTAs and RSTAs exchange at constant power (20 dBm), and although the RSSI

Figure 5.11 – Building 5 experimental setup.

is an inaccurate representation of the received power, a simple linearization and comparison between the distance computed by FTM and the expcted signal (in free path) at that distance shows that AP15 signal to any AP systematically appears 12 to 28 dB below the pairwise signal between these APs at equivalent distance. By contrast, AP16 to AP15 signal appears 10 to 25 dB above the signal from AP16 to AP01 or AP07. This result immediately indicates that AP15 and AP16 are on different floor from the others (and that both AP15 and AP16 are on the same floor). The same logic is applicable to the other buildings.

5.5.2 Geometric Wall Remover Phase

Table 5.3 displays the ratio between the mean of the measured distance and the ground truth in building 1. Highlighted in yellow, orange and red the distances that the geometric wall remover engine identifies as deviating from the others. Some devices are out of range from one another (*e.g.*, device 1 against sensor 13) and are not addressed in this phase.

As can be seen, the geometric wall remover engine correctly identifies most incorrect distances, except those affecting devices positioned at the edge of the floor. A geometric illustration of this detection is represented in Figure 5.14, where stretches on AP6 position are detected through measuring its distance to APs 1, 4 and 10.

At the scale of the entire floor, the method initially identified pairs highlighted in red and orange in Table 5.3. A contraction factor ζ was applied to each of these outliers (the step value

Figure 5.12 – Detectability of RSTAs from outside the building.

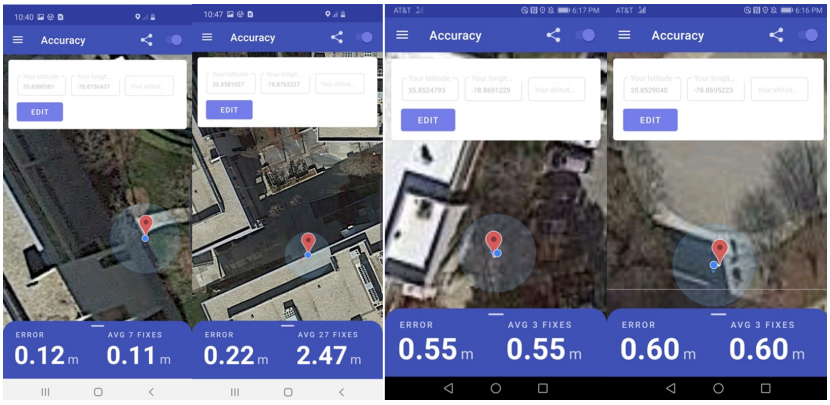


Figure 5.13 – GPS measurement accuracy near the experimental building.

of ζ was chosen to be static and small, set to 0.98). As more iterations were run, each with an additional ζ applied to identified distance dilations, the yellow pairs surfaced as abnormal. On the last iteration, the engine did not surface any more outlier, and the largest deviation from the ground truth (for the flagged pairs) was brought down to 1.13. This validation indicates that the geometric wall remover can correctly identify inner outliers and reduce the associated error, without distorting the graph (over or under reduction).

Comparable display is visible in the other buildings. FTM devices near APs on high ceiling display large distance stretches when ranging against devices near APs on lower ceilings, but devices at the edge of both domains display LoS distances and link both domains. The Geometric Wall Remover correctly identifies the high ceiling / low ceiling distant pairs as stretched, which a subset is illustrated in Figure 5.15. High ceiling APs are 1 to 4, APs in low ceiling are 11 to 15, in italic.

Device	1	2	3	4	5	6	7	8	9	10	11	12	13	eigenvalues
1	0	2409	4363	6108	1454	4022	5501	2847	3936	4870	3471	6411	7610	$8.4768e^{+7}$
2	2409	0	2266	3713	1894	2387	3959	4800	2971	3809	4732	4848	5734	$4.1676e^{+7}$
3	4363	2266	0	1896	3624	1535	1969	7619	3520	2739	6702	3376	4514	$4.3061e^{+6}$
4	6308	3713	1896	0	5273	3114	1582	9680	4880	4101	8321	3672	3557	$3.969e^{+6}$
5	1454	1894	3624	5273	0	2712	4211	2813	2786	3467	2596	5166	6476	$2.8132e^{+6}$
6	4022	2387	1535	3114	2712	0	1775	4796	2582	1620	4636	2734	3587	$5.3881e^{+5}$
7	5501	3959	1969	1582	4211	1775	0	7472	4030	2717	6293	2241	2390	$-2.037e^{-8}$
8	2847	4800	7619	9680	2813	4796	7472	0	2787	4218	1745	5481	6283	-28396
9	3936	2971	3520	4880	2786	2582	4030	2787	0	2007	1880	2940	4246	$-1.642e^{+5}$
10	4870	3809	2739	4101	3467	1620	2717	4218	2007	0	3345	1277	2649	-1.1989^{+6}
11	3471	4732	6702	8321	2596	4636	6293	1745	1880	3345	0	3842	5296	$2.3985e^{+6}$
12	6411	4848	3376	3672	5166	2734	2241	5481	2940	1277	3842	0	1464	$-3.4372e^{+6}$
13	7610	5734	4514	3557	6476	3587	2390	6283	4246	2649	5296	1464	0	$-2.1794e^{+7}$

Table 5.2 – Building 1 - Distance matrix of FTM distances and related eigenvalues

Node	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.00	1.15	1.04	1.08	1.02	1.10	1.05	1.16	1.20	1.11	1.05	*NaN*	*NaN*
2	1.15	0.00	1.06	1.04	1.11	1.19	1.16	1.29	1.03	1.20	1.27	1.14	1.11
3	1.04	1.06	0.00	1.29	1.08	1.17	1.26	*NaN*	1.03	1.06	*NaN*	1.08	1.26
4	1.08	1.04	1.29	0.00	1.10	1.26	1.10	*Nan*	1.07	1.24	*NaN*	1.10	1.10
5	1.02	1.11	1.08	1.10	0.00	1.08	1.02	1.38	1.49	1.15	1.18	*NaN*	*NaN*
6	1.10	1.19	1.17	1.26	1.08	0.00	1.09	1.19	1.22	1.15	1.35	1.20	1.14
7	1.05	1.16	1.26	1.10	1.02	1.09	0.00	*NaN*	1.18	1.37	*NaN*	1.17	1.18
8	1.16	1.29	*NaN*	*NaN*	1.38	1.19	*NaN*	0.00	1.11	1.02	1.14	1.06	1.00
9	1.20	1.03	1.03	1.07	*1.49*	1.22	1.18	1.11	0.00	1.25	1.40	1.10	1.06
10	1.11	1.20	1.06	1.24	1.15	1.15	1.37	1.02	1.25	0.00	1.16	1.09	1.10
11	1.05	1.27	*NaN*	*NaN*	1.18	1.35	*NaN*	1.14	1.40	1.16	0.00	1.00	1.03
12	*NaN*	1.14	1.08	1.10	*NaN*	1.20	1.17	1.06	1.10	1.09	1.00	0.00	1.11
13	*NaN*	1.11	1.26	1.10	*NaN*	1.14	1.18	1.00	1.06	1.10	1.03	1.11	0.00

Table 5.3 – Measured distances to ground truth ratios, and pairs identified by the geometric wall remover method as 'abnormal' in building 1

5.5.3 Iterative algebro-geometric phase

In this phase, we start with the largest possible matrix. Building 1 has 13 APs and FTM devices, and with missing distance pairs (out-of-range APs), the system iteratively finds that a matrix of size 9 starts allowing for multiple usable solutions to appear with different individual pivots, as displayed in Figure 5.16 (left).

Some points are used as pivots for all combinations in this iteration (and thus display a single position). As the matrix size decreases, more combinations appear. As the pivots change, noise also appears, as is visible in Figure 5.16 right, displaying the output of a matrix of size 8.

The process repeats iteratively. With smaller matrices and more combinations, clusters start to appear for each sensor, with various densities. In buildings 3 and 4, APs and devices in individual zones (low ceiling or high ceiling) are in LoS condition to each other. This is visible in Figure 5.17 (left), where a subset of building 3 is displayed (APs in the upper left part of the building). Green circles represent LoS (same height) matrices, while red crosses represent matrices with cross-height distances. The same density difference can be seen in building 5 (Figure 5.17 right), where APs/devices on the same side of the central block display close

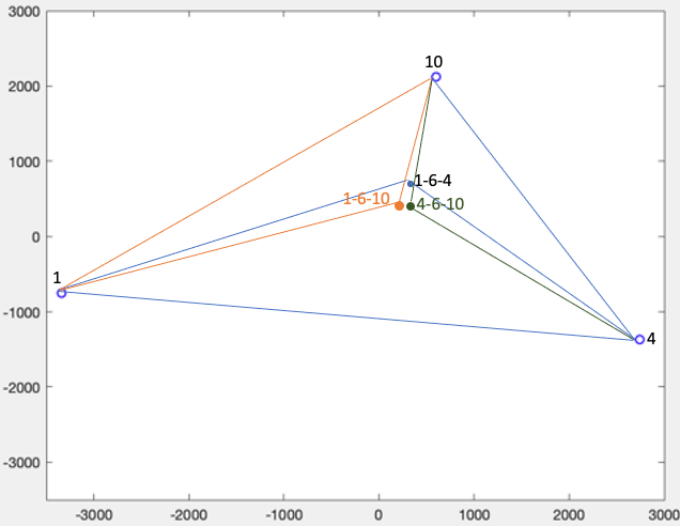


Figure 5.14 – Stretches detected for AP6 position.

Node	1	2	3	4	11	12	13	14	15
1	0.00	1.03	1.02	1.01	1.08	1.12	1.29	1.43	1.47
2	1.03	0.00	1.04	1.03	1.31	1.42	1.44	1.51	1.49
3	1.02	1.04	0.00	1.02	1.11	1.17	1.27	1.39	1.45
4	1.01	1.03	1.02	0.00	1.34	1.36	1.47	1.49	1.54
11	1.08	1.31	1.11	1.34	0.00	1.06	1.07	1.15	1.21
12	1.12	1.42	1.17	1.36	1.06	0.00	1.09	1.10	1.11
13	1.29	1.44	1.27	1.47	1.07	1.09	0.00	1.06	1.13
14	1.43	1.51	1.39	1.49	1.15	1.10	1.06	0.00	1.09
15	1.47	1.49	1.45	1.54	1.21	1.11	1.13	1.09	0.00

Figure 5.15 – Measured distances to ground truth ratios, and pairs identified by the geometric wall remover method as 'abnormal' in building 3, lower half.

measurements (narrow clusters) while measurements through the block, when they succeed, display large variations (larger clusters).

Therefore, for any cumulated graph, cluster centers can be defined and the mean cluster radius r_i compared between clusters. Then, outliers (points more than 2σ away from the cluster center) can be removed from each cluster, not only within each cluster individually, but also for individual pairs, where deviation from the mean pair distance can be used to identify matrix combinations that surface large individual error. As a matrix of size m is used to compute positions, the position for each device is compared through each iteration, and the difference between the coordinates found at each iteration can be graphed, as shown in Figure 5.18 for building 1. As the matrix size gets smaller, the noise increases, and so do deviations between computed positions, as can be seen on the right side of Figure 5.18. We found that starting from matrices as large as possible was computationally more efficient (as there are less large matrix combinations than small matrix combinations), and that accurate location was found with matrix of 6 or more for building 1. Smaller matrices only marginally improved the results,

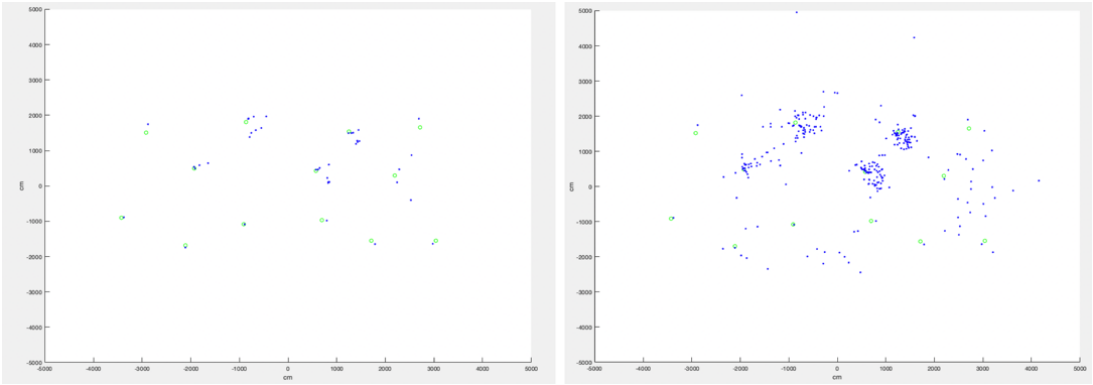


Figure 5.16 – size 9 matrix positions (left), and size 8 matrix positions (right), before outlier filtering, in building 1.

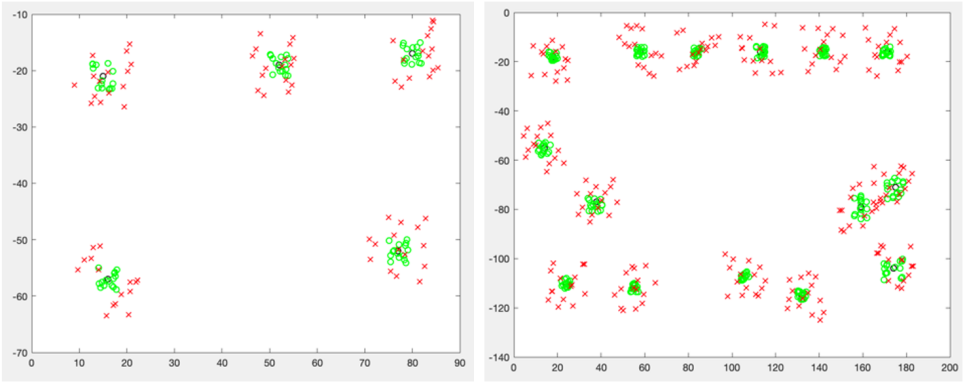


Figure 5.17 – Bldg. 3 (left, partial) and Bldg 5: position graph resulting from matrices built with APs in LoS (green circles) or nLoS (red crosses).

and the computation cost increased. The same logic was observed for the other buildings, where matrices containing at least half the floor devices (or more) produced better results.

After removing the outliers, each cluster center is re-computed. The final cluster centers are displayed in Figure 5.19. Ground truth positions are green circles, the computed cluster position for each device is represented as a red star. The maximum error is observed at 1.1143 meter in building 1, 1.2194 meter in building 2, and 0.9621 meter in building 5. In building 3 and 4, error is small in areas at the same height (0.7304 meter and 0.6812 meters respectively). When treating the APs / devices at different height as different floor levels, this result can be sufficient to move on to the next phase.

At this stage, the relative device positions are estimated, but their orientation is not known. However, using the ranging information from one or more mobile phones, walking outside the building, to two or more FTM devices, and sending to the device the GPS location of the mobile phone as seed, the graph can be rotated to its correct orientation and the phone GPS location

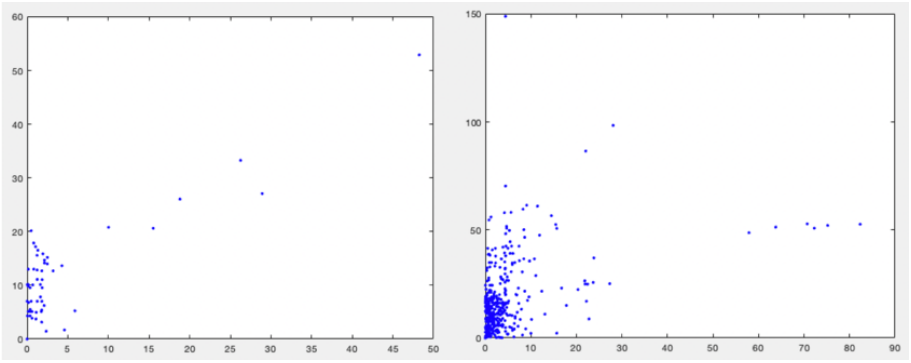


Figure 5.18 – device 9 position deviations with matrix of 6 (left) and matrix of 5 (right).

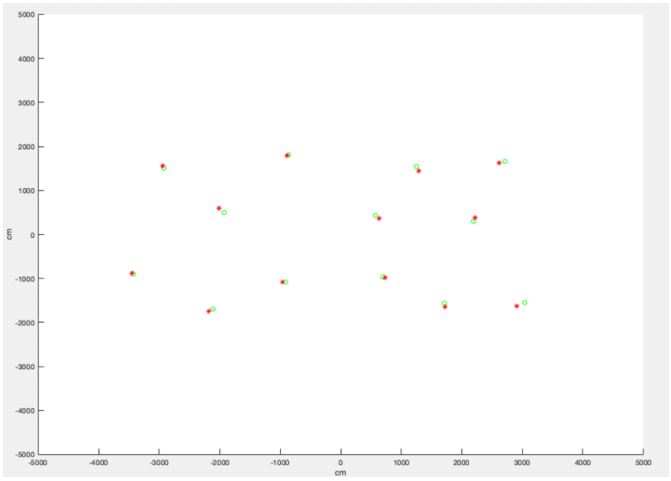


Figure 5.19 – cluster center position computation (red star) vs ground truth position (green circles) in Building 1.

can be used to populate the LCI values of all devices on the graph (and therefore all APs).

5.5.4 Comparison with other methods

Noisy EDM completion is a complex problem that has been explored in multiple contexts. We will limit our comparison to the main methods listed in [123]. Naturally, using classic MDS boils down to performing an eigenvalue decomposition and geometric centering. As the matrix is noisy, the error is large. Additionally, the algorithm interprets missing distances as 0 values, which introduces irreconcilable inconsistencies in the matrix in all dimensions. The effect is worsened when the projection is constrained into $\mathbb{R}^2$. This affect can be attenuated by constructing and overlapping partial matrices for which distances are known [124]. Even in this case, the effect of asymmetry can be dire.

Many optimizations methods aim at finding missing elements from a matrix, but that is as-

sumed to be noiseless. This assumption is necessary to maintain convexity, with the consequence of being unusable in our scenario. For incomplete and noisy matrices, a classical solution is to use semidefinite relaxation. This technique has multiple variants to adapt to different matrix sizes or sparsity scenarios, and both [124] and [125] are typical illustration of the associated reasoning. In all cases, the goal is to bound the rank of the Gram matrix to the target dimension space, thus constraining the number of positive and non-null eigenvalues. The process is efficient, especially for large matrices. As it proceeds iteratively, it also has the virtue of minimizing the error. In buildings where measurements errors are averaged over a large number of APs/FTM devices, the error may also be simplified to Gaussian. However, FTM works with a limited number of devices, making the limitation obvious. When the stretch is asymmetric, the error reaches a point where the method is not usable anymore. Building 5 is a typical example, where a strong obstacle is in the middle of the block, while light walls do not prevent the signal from reaching the neighboring classes. The result of classical MDS projection for this building is visible in Figure 5.20.

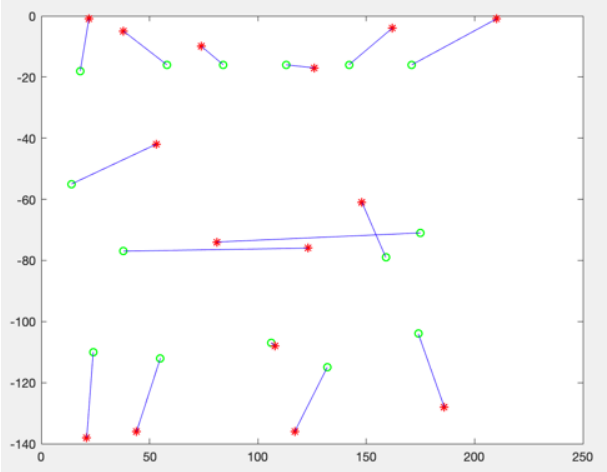


Figure 5.20 – classical MDS projection (after rotation) in building 5.

5.6 Conclusion

In this chapter, we examined a key limitation of FTM: the assumption that the location of the RSTA would be known, in places where no GPS allows for the RSTA position determination. The solution we propose relies on solving noisy Euclidian distance matrices (EDM). We showed that in the case of sensor time-of-flight measurements, measured distances are dilations of ground truth distance, but that the dilations are asymmetric, thus rendering classical EDM methods generally inaccurate, as they tend to assume noisy symmetry and therefore tend to center the error. We proposed a machine learning method for identifying and reducing noise asymmetry

based on evaluation of angles within overlapping triangles. We then proposed a second machine learning method, aimed at graphing the positions of sensors derived from distance sub-matrices, and at identifying and removing combinations that surface excessive distances, thus progressively removing edge asymmetries and reducing the distance errors. We also showed that this method outperforms standard EDM resolution methods.

With this method, deploying RSTAs (or configuring APs be play the RSTA role) becomes trivial. The implementer can deploy APs throughout the venue, then use the GPS seed of one device outside of the building (while verifying the accuracy of the GPS location reported) to simply cause all APs to dynamically learn their own geo-location. At this point, FTM is directly usable for indoor localization. In next chapter, we will look at another fundamental requirement of a modern localization technology: its ability to protect user privacy.

Chapter 6

Improving Privacy by Avoiding Fingerprinting

6.1 Introduction

At the end of the previous chapter, FTM is at a level where accurate ranging can be provided, and where AP can automatically learn their LCI. We are now ready to examine FTM more in depth. In particular, we noted before that FTM faced design limitations, where the protocol requires both sides to send frames (Section 4.3.1). This process may expose some information about the ISTA. When the ISTA is a personal device, the risk for privacy is an immediate concern. A natural reaction for implementers is to use Randomized and Changing MAC addresses (RCM) as the ISTA address when ranging (Section 3.3.6). The ISTA also does not communicate any STA-specific information (it just queries the RSTA for ranging), thus satisfying the appearance of privacy protection. This process begs the question: is RCM sufficient to protect privacy? It has been shown before that even when a station is unassociated, and even when that station randomly changes its MAC address over time, uniquely identifying the station is possible [126] by looking at the query messages that the station sends. Such unique identification (called in the literature and in this manuscript fingerprinting) relies on specific characteristics of the station transmissions, each element of which may not be unique, but that together allow for a probabilistically reliable unique identification of the transmitter. In particular, Cunche [127] showed that specific bits or information elements, and carrier characteristics were sufficient to uniquely identify 802.11 transmitters. Once alerted, chipset designers rushed to address the gaps that were based on software implementation. At lower level, Chapre et al. [128] also showed that an analysis of the physical characteristics of the transmission (CSI analysis) was sufficient to allow for such identification. Concerns about such identification have encouraged the development of recommendations to decouple device transmissions from unique patterns or identifiers (*e.g.*, IEEE 802E). At the same time, multiple machine learning strategies for indoor localization based on RSSI have been proposed, and [129] provides a good overview. Supervised learning techniques [130] have been shown to be useful, but neural networks [131] have also demonstrated high reliability. Machine learning techniques have been proposed to refine location, but also to

increase the reliability of device identification (*e.g.*, [132]). In our previous work [133], we have shown that such identification was not only possible, but that the performance of individual applications could also be predicted.

There is an ongoing race between researchers finding new ways to fingerprint a station, and international organisations that attempt to suppress the discovered gaps. Therefore, our first concern is to examine if FTM might be contributing factor to fingerprinting. In this chapter, we show that identification is possible, and propose a recurrent neural network-based pattern recognition implementation that provides unique station identification solely based on FTM exchanges. We show that FTM, as designed in 802.11-2016, dramatically increases the surface of privacy exposure, even when the station changes its MAC address for each exchange burst. We propose mitigation techniques, noting that the protocol itself would need important redesign to limit privacy exposures.

6.2 FTM identifiable patterns

A single ISTA will likely need several FTM exchanges with each RSTA in order to compute a reliable range value, thus providing a plurality of samples to work from. That same ISTA will likely need exchanges with more than one RSTA, possibly on different channels, in order to make a conclusion on its geographical location. At the same time, a single RSTA, for example an access point, may be exchanging ranging frames with more than a single ISTA, while at the same time serving multiple Wi-Fi data clients. As such, the time of each FTM exchange cannot be deterministic in nature, but still needs to be predictable.

6.2.1 FTM Burst Pattern

To allow for this combination of predictability and flexibility, recall that 802.11-2016 implements a negotiation mechanism by which the ISTA proposes exchanges parameters in the FTM Parameters information element of the initial FTM request frame, and the RSTA responds with its possible parameters in the initial FTM frame response, as detailed in Section 4.2. It is worth looking more in details into the FTM Parameters IE, represented in Figure 6.1.

One central scheduling element is the burst instance, which defines the time window during which FTM exchanges should occur. The burst instance is defined by a start time and a duration. The ISTA indicates a preferred start time by leaving the Partial TSF Timer no Preference bit unset, and by setting the Partial TSF Timer field. The field (when ISTA sets it) is 16 bit-long and expressed in TUs, thus offering a theoretical window of 65535 TUs for the start of the next burst instance. However, choosing such a wide range has limited incentive for the ISTA. In most observed implementations, the ISTA sends an initial FTM request frame when it is ready to begin the ranging exchange. There is no clear reason why an ISTA would want to initiate such

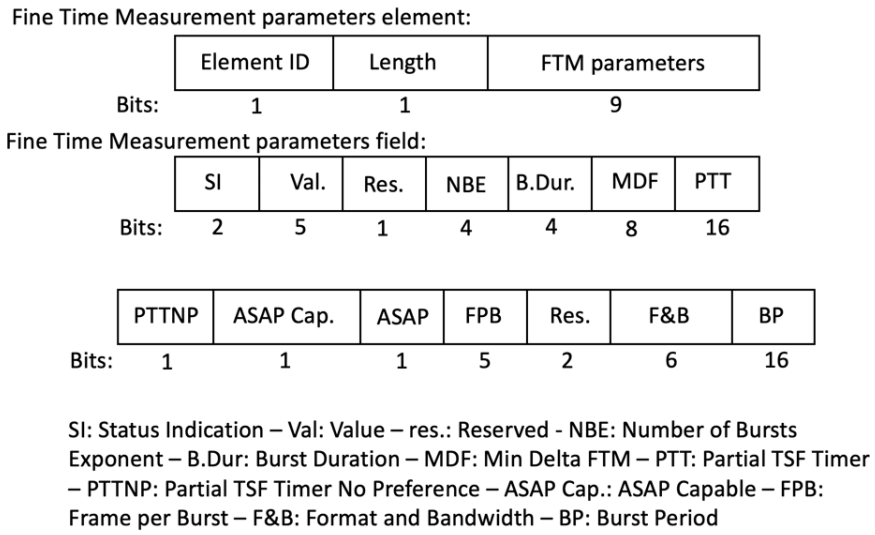


Figure 6.1 – The Fine Timing Measurement Parameters element, 802.11-2016 figure 9-576

FTM instance now, yet only start the ranging exchange more than a minute in the future. In fact, the ISTA Partial TSF timer is mostly ignored and the RSTA partial TSF Timer is used to determine the beginning of the first burst instance.

The ISTA may need to complete other tasks (*e.g.*, ranging with another RSTA, finishing a data exchange with an AP *etc.*), and therefore may need to delay the beginning of the next burst instance. In that case, the delay indicated in the Partial TSF Timer is likely to be implementation-dependent, where the ISTA sends the initial FTM request with a partial TSF timer set, only when the ISTA has determined the necessity of a partial TSF Timer, to indicate a preference when it will be available to start the FTM exchange. Therefore, it is likely that the partial TSF timer will be predictable for each ISTA.

Alternatively, the ISTA can set the ASAP bit and thus indicate that it is ready to start ranging as soon as the RSTA is ready. If the RSTA responds and indicates that it is ASAP-capable, then ranging can start immediately. The partial TSF timer value is limited in value when sent by the RSTA, and is intended to avoid issues resulting from clock synchronization drifts between the ISTA and the RSTA, and frame collision between the RSTA response and the first frame from the ISTA at the beginning of the burst. The Partial TSF Timer for the RSTA is also to read in comparison with the TSF Timer of the RSTA current frame. However, such timing constraints do not appear for the Partial TSF Timer set by the ISTA. For example, the ISTA could set the ASAP field to 1 or the Partial TSF Timer to 0, both indicating a readiness to start the exchange immediately. Implementations show both modes.

6.2.2 FTM Individual Burst Structure

The second element describing the burst instance is the burst duration. 802.11-2016 defines 10 possible values, each represented by a burst duration index: 250 (value 2) and 500 microseconds (3), then 1 (4), 2 (5), 4 (6), 8 (7), 16 (8), 32 (9), 64 (10) or 128 milliseconds (11). A value of 15 indicates that the ISTA has no preference. What factor can drive an ISTA to choose one burst duration over another? Theoretically, several elements may contribute to the duration determination. Among others, a higher channel utilization may incentivize the ISTA to choose a longer duration so as to increase its chances of completing enough exchanges with the RSTA. Similarly, a high Tx buffer level or a high signal stochasticity (high standard deviation of the RSTA or AP RSSI from the ISTA perspective) may also be deciding factors. However, it may often be computationally more economical for chipset or device vendors to implement a default burst duration, and add more bursts if more ranging samples are needed.

The goal of the burst is to achieve a number of FTM exchanges with the RSTA. The ISTA can indicate this target FTM exchange count within each burst with the FTMs Per Burst field. Values are coded over 5 bits, thus allowing from 1 to 31 FTM exchanges per burst. In most cases, the ISTA would not want all the FTM exchanges to occur within a small sub-segment of the burst, but be spread across the duration of the burst. To that effect, the ISTA can define a Min Delta FTM value, that indicates the minimum time between consecutive Fine Timing Measurement frames. It is measured from the start of an FTM frame to the start of the following FTM frame, in units of 100μs. The field is 8-bit long, thus allowing intervals up to 25.5ms between FTM frames. The value 0 indicates no preference by the ISTA.

6.2.3 FTM Burst Count and Bandwidth Parameter

The ISTA can also request how many bursts should occur within the FTM session. The Number of Bursts Exponent is a 4-bit long field, and is an exponent, thus allowing the values 0 to 14 (15 is reserved), and thus up to 16384 bursts. When more than one burst are requested, the ISTA can also specify an interval between bursts, by setting the Burst Period, that describes the interval between the beginning of a burst and the beginning of the next burst, in units of 100 ms. This 16-bit long field allows for up to 655.35 second intervals.

Last, the ISTA can specify the format and bandwidth of the exchange, from 20 MHz exchanges with standard OFDM frames to 160 MHz-wide exchanges with VHT frames. The ISTA can also request exchanges over the 60 GHz band. The format and bandwidth obviously match the ISTA capabilities compared to the RSTA capabilities expressed during the detection phase (*e.g.*, probe response). In most cases, the ISTA will attempt to use the widest possible band, because larger signals offer better accuracy in signal detection (see Section 4.4.1). Therefore, the format and bandwidth field may be indicative of the ISTA maximal capabilities, and is likely to always be the same for a given ISTA (provided that the RSTA supports the ISTA bandwidth

and MCS). However, the other fields offer a very wide set of possible combinations:

— Number of bursts (Burst count): $B_c = 2^c$ where $0 \leq c \leq 14$

— Burst duration: $2^n.B_D$
$B_D = 250$ and $n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

— Min Delta FTM: $min\Delta_{FTM} = 100\Delta_i$ where $i \in \mathbb{N}andi \leq 255$

— FTM per Burst: $0 \leq N_{FTMPB} \leq 31$
With the constraint: $min\Delta_{FTM}.N_{FTMPB} \leq 2^n.B_D$ (at least theoretically, we observed implementations where this constraint was ignored)

— Burst period: $B_P = 100.B_j$ where $B_j \in \mathbb{N}$ and $B_j \leq 65535$ with the constraint: $2^n B_D \leq 100B_j$

Although B_P can be 100 or 200, B_D is at least 250, therefore B_j cannot be 1 or 2, even if the protocol allows such values.

6.2.4 FTM Parameter Possible Permutations

Combining all these relative constraints together can be reduced to an extremal set problem, which amounts to a permutation with exclusions. The number of possible permutations is slightly above 200 billions. Therefore, FTM presents an unusual case in 802.11. When a large number of permutations exists, 802.11 often defines increments where initial values should be chosen within a smaller range, and increased (*e.g.*, doubled) when some conditions are detected. In other scenarios, 802.11 suggests that a number be randomly selected from a large range. However, no such framework appears for FTM. Stations can pick numbers as they wish, provided the above constraints are respected. Some combinations seem to produce similar outcomes. For example, it is not clear why an ISTA would prefer to exchange 16 frames over 4 bursts of 4 ms each instead of exchanging those same 16 frames over 8 bursts of 2 ms each.

To this large range of options with no clear individual use case, 802.11-2016 also stays silent on some critical elements. For example, the burst duration and number of frames per bursts are defined, but do not account for access delay of the initial FTM trigger frame. As a result, a burst duration limit may be reached before the expected count of FTM exchanges for that burst was completed. In that case, some vendors may choose to complete the FTM exchange count, thus extending the exchange beyond the burst duration. Some other vendors may choose to stop at the end of the burst, thus risking a smaller (or even an odd) number of FTM exchanges for that burst, with the consequence of insufficient sampling to achieve reliable ranging.

Given this large set of possible permutations, our hypothesis is that individual APs or STAs FTM exchange characteristics are the result of individual vendor implementation choices, with only a few variations matching pre-defined scenarios or conditions, such as buffer state or RF conditions. We hypothesize that a particular implementation can be recognized from its FTM

exchanges and the particular combination of thresholds it relies on. Coupled with individual station characteristics, such as clock drifts or other individual markers, these individual choices make it possible to fingerprint individual stations when enough FTM exchanges are collected.

6.3 A recurrent neural network to fingerprint FTM

In order to test our hypothesis, a simple parsing of various stations initial FTM request or initial FTM frames would not be sufficient. These frames do indicate preferred settings in the FTM Parameter IE and do allow for vendor identification. In an environment with a limited number k of stations, this information is obviously sufficient to uniquely identify the station if k is small [134] and the stations are diverse. In a denser environment or when similar stations are performing ranging in the same area, these elements alone would not allow for individual station identification with similar chipsets and operating systems. But competing traffic and processes, along with individual station state and hardware, can surface differences between two stations with the same chipset and operating system, even if the stations display the same FTM parameter IE values. It is clear, as exposed in the introduction of this chapter, that other frames, and the PHY layer of most frames, could also be used to perform such individual identification. But our objective is to determine if the pattern in FTM frames themselves could be used for such individual identification, including in a dense environment where more than one FTM station use a given chipset, driver, and operating system.

6.3.1 Structure Contrast Between Different Chipsets

The experimentation scenario we built accounted for this need, and involved diverse stations, with different form factors and chipsets, but also several stations with a similar chipset but various operating software update levels and competing applications. As shown below, we could uniquely identify similar stations with high accuracy. Such identification requires pattern recognition, while discarding environment-specific factors. For this purpose, we collected FTM exchanges from 11 different ISTAs, against 2 different APs operating as RSTAs, in 2 different environments: one office building floor with a mix of open spaces, cubicles and meeting rooms, and one shopping-mall-like open space. Samples were taken at 60 different location points. The experiment was repeated at 3 months intervals in both locations, to avoid time-and-environment-specific bias. The goal was to evaluate how an ISTA could be fingerprinted through a device acting as an RSTA. Therefore, only the elements that would be available to the RSTA were collected. For each measurement, the ISTA MAC address, the time of the ISTA initial FTM requests, the time of the RSTA FTM frames, the ToA and ToD values sent by the RSTA and the ISTA were recorded. Other traffic from the RSTA and other stations was filtered out.

When checking individual samples (for example, the frames from two different ISTAs posi-

tioned at a similar point on a map, and therefore at comparable distance from the same RSTA), individual ISTA identification proved trivial. A simple observation of individual parameters on a graph clearly shows that the implementation of individual chipsets on different operating systems results in specific and unique behaviors. For example in Figure 6.2, a side-by-side comparison shows that an ISTA with an intel chipset (8260) running on a Linux (Ubuntu 16.04) system (left side) displays different transmission signal characteristics than an ISTA using a Qualcomm chipset (8996) running on a Google Pixel phone running on Android 9 (right side). The graph shows the time interval between individual FTM frames sourced from the ISTA. Strong density is visible at 0.2 seconds for Pixel, with higher probability of frame for the interval 0.26 to 0.32 seconds. Density zones are visible for Intel in the 0.1-0.12, 0.16-0.20 and 0.23-0.31 second areas.

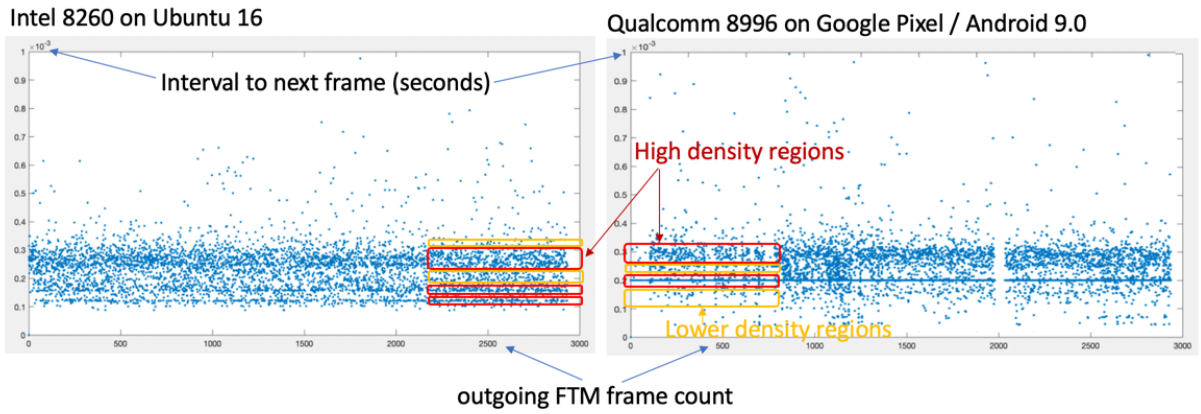


Figure 6.2 – Interval between outgoing FTM frames on 2 different ISTAs

Therefore, training a simple regression model to recognize different types of vendor settings would be considered trivial and our measurement showed obvious differences for this case. However, distinguishing stations using similar vendor chipsets would be more difficult, as their density regions would likely appear similar with superficial analysis. When the sample size is small (two ISTAs at a given position ranging to the same RSTA), individual differences can easily be seen, that relate to the particulars of each station at sampling time (competing upstream or downstream data traffic, differences in individual power levels or other). Performing recognition on such localized samples results in high identification probability in most cases. To allow for such recognition to be systematized, and avoid condition-specific triggers, we mixed the samples, concatenating, for each station, all samples from all locations in a single file. Each file contains the FTM frames exchanged with the ISTA, one column for each recorded field (ISTA MAC address, the time of the ISTA initial FTM requests, the time of the RSTA FTM frames, the ToA and ToD values sent by the RSTA). Other elements were filtered out, so as to focus solely on the FTM exchanges. Each file was then divided in slices of 200 frames each. Depending

on the station type and its FTM cycle and query rate, such 200-frame sample can represent anywhere between 0.80 seconds and 82 seconds of FTM exchanges.

6.3.2 Measuring the Effect of ISTA Parameter Choices

The individual impact in the RF environment of such ranging exchange sample is measurable. The duration DF_1 of an individual FTM exchange can be computed as follows:

$$\begin{aligned} DF_1 = ((K+1) \times (T_{IFTMR})) &+ (K-1) \times (EIFS + Random() \times aSlotTime)) \\ &+ ((K+1) \times T_{FTM\_1})) \\ &+ ((K-1) \times (EIFS + Random() \times aSlotTime)) \\ &+ T2FTM_1 + 2 \times aSIFSTime + 2 \times T_{ACK} \end{aligned} \tag{6.1}$$

Where K is the number of initial FTM request retransmissions that the ISTA may attempt, T_{IFTMR} is the duration of an individual initial FTM request, T_{FTM_1} is the duration of the RSTA FTM_1 frame, $T2FTM1$ is the interval between the ISTA initial FTM request and the RSTA FTM1 frame, and T_{ACK} is the duration of the ACK frame.

An individual FTM exchange can occur at any supported data rate. For simplicity, suppose a 6 Mbps exchange (we observed that 6 and 13.5 Mbps were common rates). An initial FTM request exchange, without retransmission, would use:

$$DIF_{6MBPS} = 92 + 16 + 44 = 152\mu s. \tag{6.2}$$

Then, within each burst, an individual FTM exchange pair will consume:

$$\begin{aligned} DF_1 = &((K+1) \times T_{ACCESSFTMR}) + ((K+1) \times (T_{IFTMR})) \\ &+ ((K-1) \times (EIFS + Random() \times aSlotTime)) \\ &+ ((K+1) \times (T_{FTM})) \\ &+ Min\Delta FTM - ((K-1) \times (EIFS + Random() \times aSlotTime)) \\ &+ T2FTM_2 + 2 \times aSIFSTime + 2 \times T_{ACK} \end{aligned} \tag{6.3}$$

Where $T_{ACCESSFTMR}$ is the estimated medium access time for the FTM request frame, including AIFS and possible EIFS cost, K is the maximum number of Fine Timing Measurement frame transmissions the STA might attempt, $T2FTM_2$ is the interval between the ISTA initial FTM request and the RSTA FTM2 frame, $Min\Delta FTM$ is the Min Delta FTM value and T_{ACK} is the expected duration of the expected ACK frame. For simplicity, suppose a 13.5 Mbps exchange

and no congestion or collision. An FTM frame pair exchange, without retransmission, would use:

$$DF_{13.5MBPS} = (68 + 16 + 44) + 43 + (68 + 16 + 44) = 299\mu s \tag{6.4}$$

Therefore, and ignoring the lower cost of the initial FTM request exchange, 200 frames representing 100 exchanges, would consume 29.9 ms. As each ranging measurement has limited accuracy, it is not uncommon to see implementations counting 2 to 4 bursts of 8 to 12 measurements for each ranging estimate and averaging the range to 4 APs or more to determine a position. As such, 200 frames represent approximately 2 consecutive ranging attempts. The training set is obviously larger and included around 40000 samples per stations for a total of about 9000 slices. Three stations implementing an Intel 8260 chipset on Ubuntu 16.04 were compared against one another, and against a reference device implementing another chipset (Pixel 1 on Android 9.0 with Qualcomm chipset 8996). Figure 6.3 shows the number of samples that were collected for the compared stations. Each element of the bar chart is one client, and the vertical axis is a count of collected sample for that client.

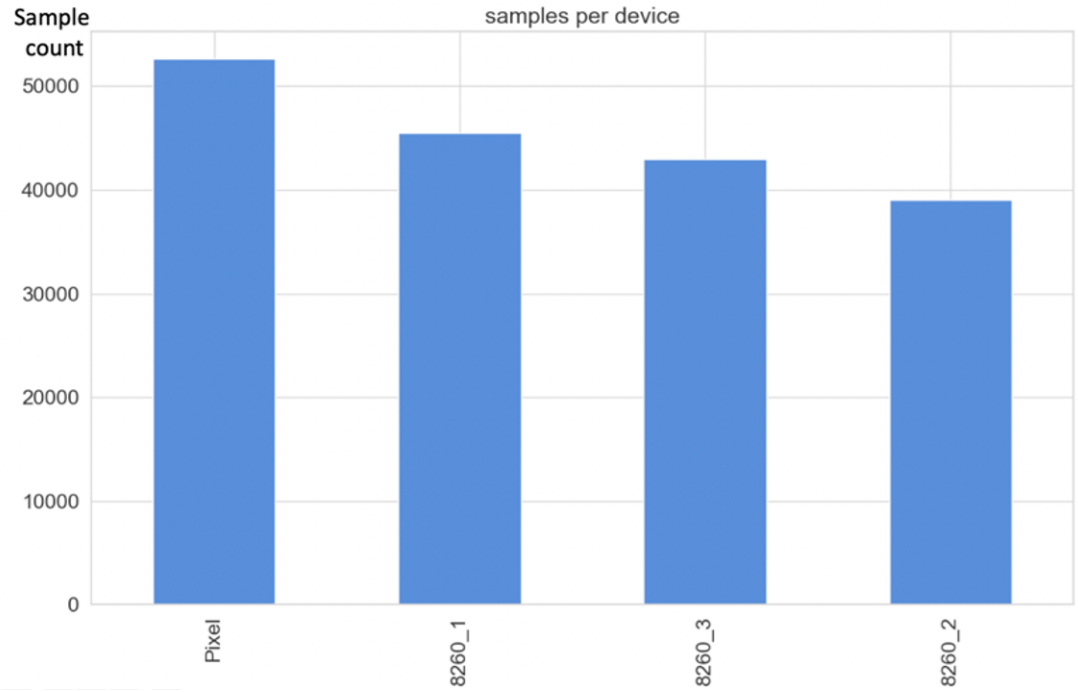


Figure 6.3 – Number of samples collected for same-chipset comparison experiment

8260_1 and 8260_2 use Intel 8260 chipsets inserted in Dell XPS 14 laptops. Both laptops run Ubuntu 16.04. 8260_2 has received maintenance operating system updates but no chipset or WiFi driver update), and also runs various applications (emails and others). 8260_1 has the initial 16.04 release, no update and no added programs. 8260_3 runs Ubuntu 16.04 on an IoT

board (https://fit-iot.com/web/products/fitlet2/).

An analysis of the FTM frames sent by each client already indicates pattern specificities for each device. Figures 6.4, 6.5 and 6.6 shows samples of FTM frames sent by three stations. The top (blue) graph shows the occurrence of ISTA FTM frames and RSTA responses. A spike structure indicates alternating ISTA and RSTA frames, while a flat structure indicates a repeat of the matching frame type. The center graph shows the ToD value (t_1) reported to the ISTA by the RSTA, and the lower graph shows the ToA t_4). Quite obviously, these valeus would not be visible to an attacker if the exchange was protected (*e.g.*, with PASN, PASN-FT (see next chapter), OWE or another mode), but the pattern is obvious, and the observer can measure the time between the AP frames to recombine the likely bound for t_1 and t_4. This method is used and exposed in more details in Section 7.4. For both graphs, the vertical axis represents the value transmitted in the frame. For all graphs, 200 samples are displayed, and the horizontal axis specifies which samples are shown.

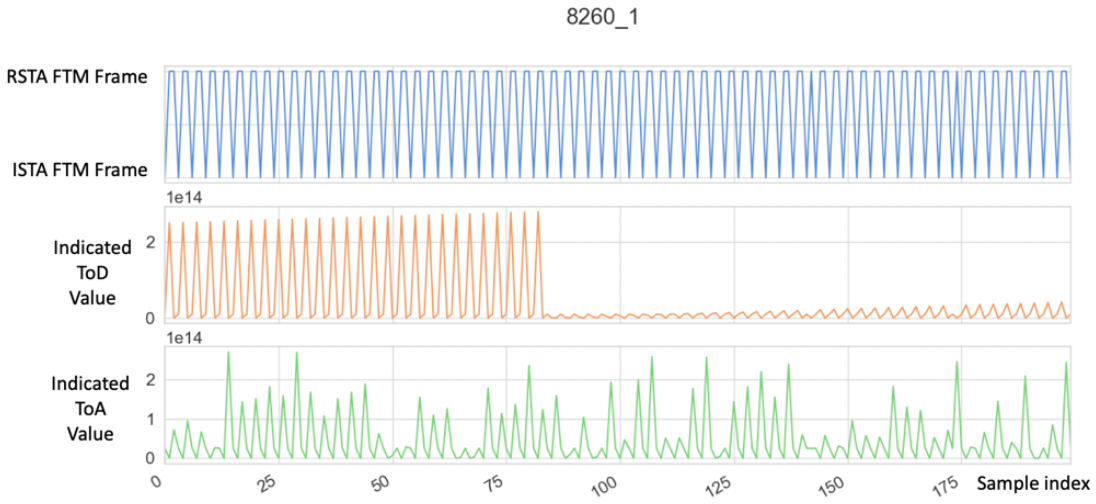


Figure 6.4 – FTM frames for client 8260_1. Top figure shows the ISTA request and RSTA responses, the second figure shows t_1 and the lower figure shows the t_4 values.

8260_1 and 8260_2 displayed similar visual patterns at the scale of 200 frames (therefore, only 8260_1 is displayed). The ToD shows an incremental value, while the ToA follows a stochastic pattern. However, 8260_3, also using the same chipset, use consistent values for both ToD and ToA. By contrast, the QCM behavior is very different. Large intervals are expressed for the burst start, ToD is only set at intervals, and ToA is also very stochastic. As the AP is the same in all cases, the only possible cause is a divergence in the ISTA behavior, in the processing structure of the AP messages, and the interval between frames that the ISTA requested.

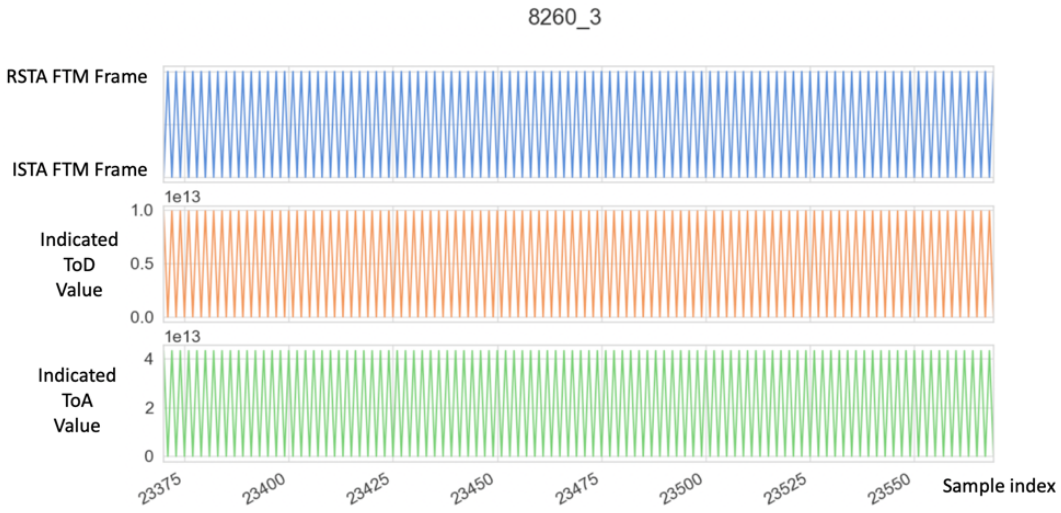


Figure 6.5 – FTM frames for client 8260_2. Top figure shows the ISTA request and RSTA responses, the second figure shows the t_1 and the lower figure shows the t_4 values.

6.3.3 Designing a Test To Identify 2 STAs running the Same Chipset

Consequently, it can be inferred that distinguishing a chipset from another would be fairly trivial, but distinguishing 2 stations implementing the same chipset on the same operating system may be more challenging for a human eye. However, the pattern of these frames may be accessible to a learning machine. Therefore, we injected these frames into a Long Short-Term Memory (LSTM) Neural Network implemented in TensorFlow [135]. The terminology for such structure is not standardized. In this chapter, we define the LSTM unit as being composed of a cell, an input set of vectors (input gate), an output gate and a forget gate. The advantage of such structure is that the unit can 'remember' the previous samples. This recurrent structure allows the unit to account for the previous frames (in our case) when evaluating the next one and thus better identify dependencies between frames (instead of considering each frame individually). Such networks are commonly trained for image recognition, but any structure that can be organized in slices that can be graphically represented can be analyzed using such networks. Such projection does not always surface a meaningful structure, but it did in this case, because the projection caused patterns along the time axis to be visible.

In our case, time slices form the x axis, and labelled ToD and ToA (as measured by the attacker) for each FTM frame type form the values of each slice dimension on the y axis. Formally, each of these elements forms the input layer. The model contains 2 fully connected layers and 2 LSTM layers, stacked on each other, with 64 units in each layer. The output Layer consists of 4 units, one for each type of station that needs to be identified. We follow there the deterministic school, where each possible output is its own container, as such approach is

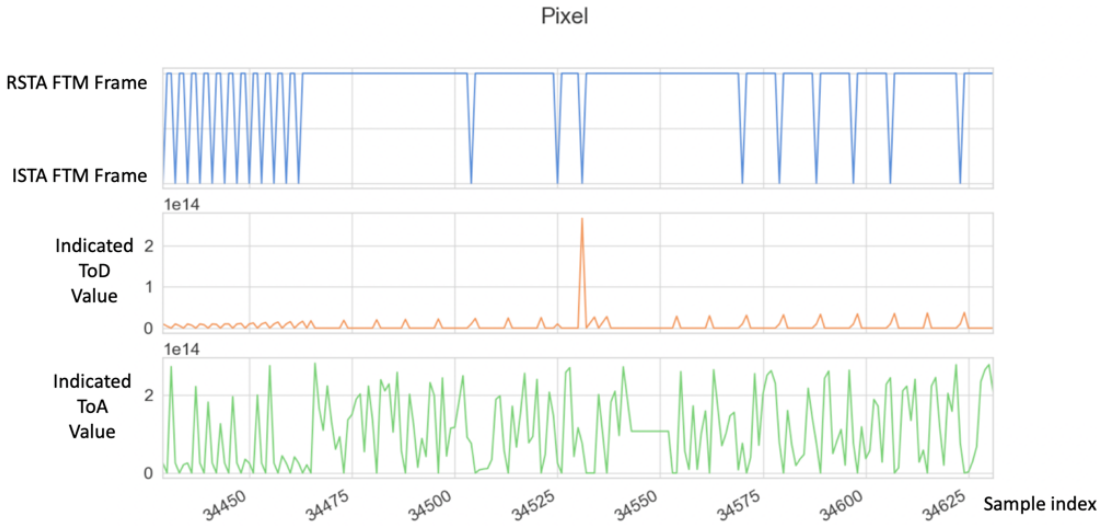


Figure 6.6 – FTM frames for client Pixel. Top figure shows the ISTA request and RSTA responses, the second figure shows the t_1 and the lower figure shows the t_4 values.

increasingly seen as more reliable than the default approach (where one container would be missing and low probability for the present containers is taken as signifying the last container). We use L2 regularization (0.00015) to limit overfitting, and a learning rate of 0.000025. The training runs over 80 epochs with a batch size of 32 slices.

6.4 Recurrent neural network evaluation results

An observation of one training session's progress over the various epochs shows accuracy reaching 95% with a loss at around 0.2, as shown in Figure 6.7. The learning curve shows that the model reaches accuracy as soon as 8 epochs, but needs up to 35 epochs to refine the weights to a level where losses stabilize and the model becomes fully efficient.

Once the training completes, the model is expected to recognize individual stations in 95% of cases. An easy way to visualize the success ratio for individual stations is to use a confusion matrix. The confusion matrix in Figure 6.8 shows, for each slice of 200 frames, which device the model thought was identified. That estimation is compared to the real device label in the training set. As expected, the Pixel device can be easily distinguished from the 8260 chipsets in almost 100% of cases. The 8260_3, implemented in a different form factor from the other 8260s is also recognized with almost 0 error. The 8260_1 and 8260_2 are differentiated correctly in close to 90% of cases, even though they are the same chipset on the same type of laptop with the same operating system. The same test performed with 6 similar stations (same OS, same chipset, various competing applications or update levels) still shows an accuracy close to 80%.

Figure 6.7 – Learning progression over epoch iterations

As the number of similar stations increases, the accuracy is expected to decrease, thus reducing the privacy exposure only when a large number of alike-stations are ranging at the same time, from the approximate same location, against the same RSTA.

This outcome confirms that local specificities, such as operating system updates and competing applications can generate specific constraints to the FTM operations performed by the chipset, resulting in differences in bursts, frame count per burst, timing choices and other FTM parameters that can be detected with only 200 exchanges. If a ranging evaluation against 4 APs consumes approximately 100 frames, a station could be recognized individually as early as the second exchange, then be tracked individually, even if it changes its MAC address.

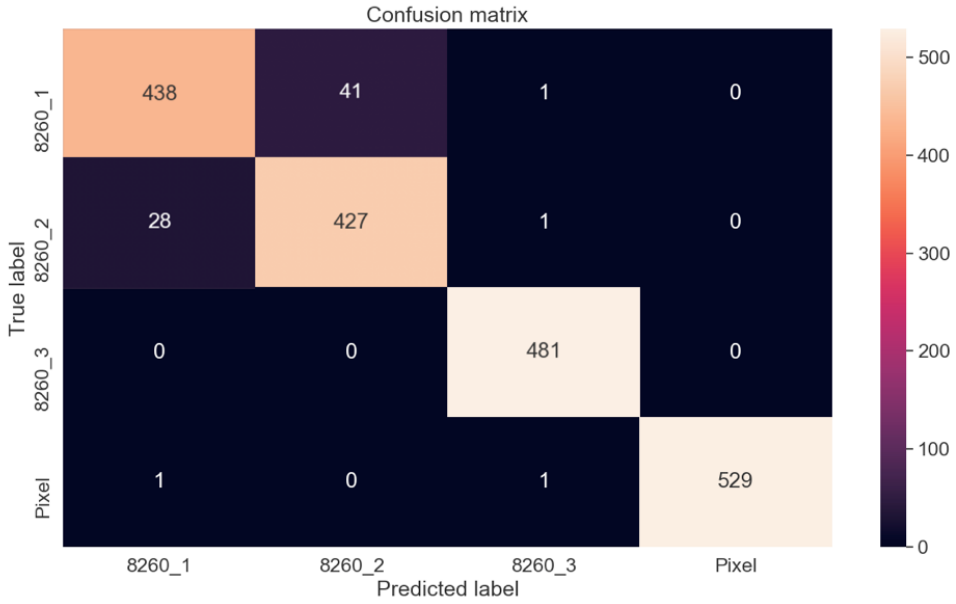


Figure 6.8 – Confusion matrix for label efficiency determination

6.5 Fingerprinting Remediation

Such identification is unfortunate for user privacy. Fast identification requires the attacker to store and process the first 2 exchanges, and also to continuously track the target station. For non-associated stations, frequent rotation of the RCM MAC address (*e.g.*, at each FTM session against each AP, see Section 3.3.6) may mitigate the efficiency of the collection. However, this protection is only partially effective. By nature, fingerprinting is a probabilistic determination and thus depends on the sample size. If the sample size includes hundreds of devices of largely the same type (*e.g.*, a crowd at the entrance of a stadium, all with the same types of smartphones, and all using FTM to find their assigned seat), then RCM may bring some protection, as within a small zone, the difference between devices' patterns may not be sufficient to obtain a good probability for the identification of individual stations.

However, this case of flash mob is likely to be the rarer case. In many public venues, people enter by small groups and there may only be a few of them actively using FTM to find their way to a PoI. Therefore, the sample size will often be small. In this context, using the RSSI of the target station as a filter is efficient, and defeats RCM protection. We know from Sections 1.3.1 and 3.3.6 that RSSI is a poor data point to compute distance. However, the attacker's goal is not to compute the distance to the victim's station, but merely to continue tracking the same station as it rotates its MAC address. As Figure 1.2 illustrates, the RSSI curve is stiff at short range, and becomes monotonic as distance increases. We also know that the signal is noisy. Thus the attacker only needs to consider in the sample all stations (irrespective of their MAC

address) which RSSI was measured to be in a given range (lower RSSIs allow for a broader range, to account for the monotonicity of the signal curve) to reduce the number of possible candidates which FTM parameters are inserted into the LSTM engine. If, in a given cell, and at a given second, 11 stations or less (as in our experiments) are actively using FTM ranging, the deterministic identification of individual stations remains possible, even as stations rotate their MAC addresses.

6.5.1 Analysing the Causes for Fingerprinting

In order to reduce the exposure of FTM to fingerprinting, it is essential to understand why the vulnerability occurs in the first place. A careful examination of the experiment above shows three different contributing factors:

1. The FTM burst parameters are observable, even when the FTM frames are protected (encrypted). This is because, even if the attacker cannot see the content of the frame, the result of the exchange is a specific structure (number of bursts, frames per burst, interval between frames within a burst). Observing the result of the parameter negotiation is equivalent to observing the negotiation itself, as both express the same quantities. The only exception would be if the AP systematically overrode the ISTA parameters, but we saw in Section 4.3.2 that this would reduce the efficiency of the method.
2. FTM provided no guidance on what burst structure to use in what scenario (RF conditions *etc.*), leaving individual vendors to make arbitrary choices on what parameters the ISTA should request. Without a common view and common parameter requests, fingerprinting will continue to allow the identification of individual chipset types.
3. Within the ISTA operating system, the competition between traffic types causes different patterns, where application data is interleaved with FTM exchanges, in a way that can be identified. Here again, without a common agreement between vendors on the type of priority given to the various calls to the chipset, fingerprinting will continue to be possible.

6.5.2 Reducing Fingerprinting

A solution to limit fingerprinting would then aim at limiting the individual FTM parameter choices operated by various chipsets and operating systems. At the same time, the pressure of other traffic needs to be accounted for. Thus an efficient solution space would have the following properties:

— All ISTAs would always request the same parameters. As this requirement is too rigid, a relaxation of this rule could be a set of parameter families that match groups of RF conditions (*e.g.*, in a quiet channel, always request 2 bursts of 8 frames at short intervals; in a noisy channel, request 4 bursts of 4 frames at longer intervals).

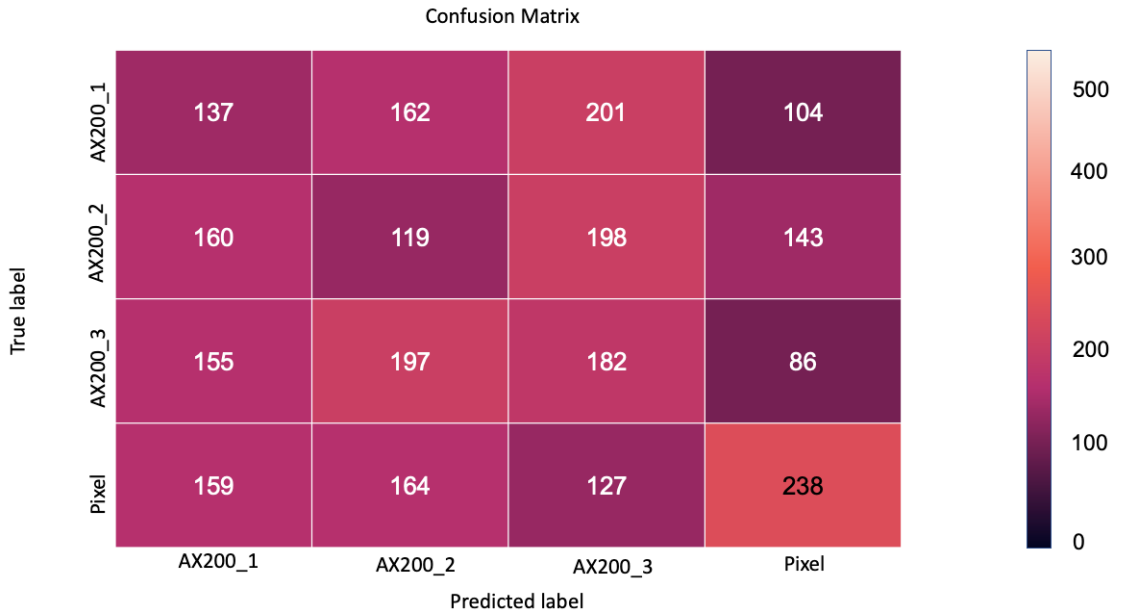


Figure 6.9 – Confusion matrix for label efficiency determination in recent AX200 chipset

— The RSTA would only override downward. In other words, the RSTA could use its own RF conditions to override the ISTA requested parameters, but only to slow down the transmission (thus reacting to poor RF conditions at the RSTA level). The RSTA would thus never increase the number of burst, the number of frames per burst, or the tempo of the exchange (reacting to good RF conditions at the RSTA level).

— The burst structure would be short enough that competing traffic would not need to be inserted within a given FTM session, and so that clock drift could not easily be measured within a given session. For example, an FTM session planned over 82 seconds (as observed in Section 6.3.1) will necessarily cause interruption for data traffic insertion. By contrast, a short session (few tens of milliseconds) may not cause data interrupts (except if a voice call is present, for which an insertion pattern could be designed).

As can be seen, the issue goes beyond the standard itself, and touches the implementation of the standard. The Wi-Fi Alliance is the place where implementation discussions and interoperability tests occur. As we published these vulnerabilities [10] in 2019, the Wi-Fi Alliance Location group was actively working on improving the performances of FTM. Naturally, all vendors are receptive to privacy and security issues, and our work was well received.

The 8260 is an older chipset, for which the manufacturer had no plan to continue driver development and OS integration. However, another chipset, the AX200, was in active development. Figure 6.9 shows the confusion matrix comparing laptops using the AX200 and the Pixel, with the latest driver and OS integration versions. As can be seen, it has become much more

difficult to fingerprint an individual station.

6.6 Conclusion

It is often said that outdoor localization has mostly been solved, but indoor localization is left to be solved. FTM can contribute to solve this issue. This chapter examined the various parameters defined for 802.11 FTM, and showed that a large combination of multiple possible parameters, each allowing a wide range of possible values, with few protocol constraint as to which value should match which ranging context, resulted in a scenario where individual vendors would be left to choose the parameter subsets that they would find reasonable to the various ranging use cases that they envisioned. As such, individual chipsets could easily be recognized from a few FTM frames. We also showed that, for similar devices implementing the same chipsets, individual software variations rendered the unique identification of each device possible through a neural network. We suggested that parameter range unification among vendors, along with proper MAC and timer randomization, might help alleviate these identification risks. Fortunately, the privacy exposure caused by the elements detailed in this chapter also alarmed the community of vendors implementing FTM, and at the time we write these lines, we are pleased to report that fingerprinting issues are addressed in all systems implementing FTM.

In the next chapter, we will turn our attention to the security of the FTM exchanges, to examine if an attacker could attack FTM the way GPS location is attacked.

Chapter 7

Protecting the FTM Exchanges

7.1 Introduction

The outcome of chapter 4 is workable FTM solution, that provides a ranging accuracy in the order of 1.5 meters. At the outset of chapter 5, we also have a framework where FTM can reasonably be deployed in an indoor environment, and where the APs automatically find their LCI coordinate values. Then, at the conclusion of last chapter, we have a solution where an observer cannot casually fingerprint a client with limited efforts, by just watching the tempo of its FTM exchanges. At this point, an FTM deployment can provide indoor ranging to any supporting mobile device. However, we saw in Section 4.3.2 that FTM had design limitations, namely that frames had to be exchanged in both directions, including in cases where the ISTA would not have an association to any APs in the network. This last chapter will then look more carefully at these limitations, and examine whether this design poses specific challenges for FTM localization. We will see that FTM, as designed in 802.11-2016, exposes the mobile device to severe risks of ranging and location attacks. We call these GPS-like attack, because they copy in spirit attacks that were developed against the GPS solution. However, we will also see that these risks can be mitigated with improvements to the Standard.

7.2 GPS Attacks and Remediation

Attacks against FTM, a new protocol, have not been studied much yet. However, attacks against GPS, also using ToF to known sources, have been widely studied and can be used as a starting point for our examination. Similar to FTM, the GPS signal available for civilian usage is neither encrypted nor authenticated. The same limitation is currently true for the other systems, Gallileo, GLONASS, Michibiki, BeiDou and NavIC, although protection schemes have been proposed for the BeiDou protocol [136]. The simplest attacks aim at completely jamming the GPS signal [137], or preventing selected satellite signals from reaching the station [138]. These starvation attacks are efficient, but of course trivial to detect. More elaborate attacks spoof the identity of one or more valid satellites [139], and provide signals which timing and timestamps cause the station to misevaluate its distance to the source [140] and compute an incorrect location [141].

In some cases, these attacks can be detected. A simple technique is to use Receiver Autonomous Integrity Monitoring (RAIM) [142]. With this technique, the client uses a rolling subset of satellites, and alerts if any particular satellite contribution provides results inconsistent with the others. This is a simple outlier detection system. We will see that a similar system can provide a partial protection for FTM, but it is insufficient on its own. Its efficiency in the case of GPS is also challenged when the attacker carefully injects wrong values progressively. Therefore, additional techniques are needed, that look at each individual signal. For example, Foruhandeh *et al.* [143], fingerprint each satellite signal, and recognize the impersonation by matching the received signal against the expected signature. Other techniques use fusion (see section 2.4.3 to compare the GPS-computed location with estimations from other sources, like cellular towers [144] or accelerometers and other sensors internal to the station [145]. More complex, multi-factor detection methods naturally augment the reliability of the detection [146], for example when neural networks combine all contributing elements together to determine if one of them is surfaced as an outlier [147].

All these techniques mitigate the surface attack, but do not completely remove it. A large trend is to apply to the civil GPS the same authentication and encryption scheme as military grade GPS [148], thus disabling the spoofing risk, and complement security with resilience features to disable the jamming risk [149]. We will see that, if similar attacks affect FTM, the protection modes offered for GPS only imperfectly apply to the protection of the FTM case.

7.3 GPS-like Attacks on FTM

7.3.1 FTM Location framework

First, we need to recall from Section 4.2 that 802.11 FTM focuses on the ranging exchange (not the position computation). From its ranging exchange with the RSTA, the ISTA obtains the (t_1, t_4) values and uses them with its own (t_2, t_3) timers to compute its distance to the RSTA, retaining the shortest ToF from each burst. The ISTA also requests the RSTA LCI. After having exchanged with different RSTAs, the ISTA can combine distances and LCIs to compute its own location. Depending on the implementation, the ISTA uses the three sphere method (Section 1.3.3) or a matrix method (Section 1.3.4), possibly (but often) augmented with a filter like Kalman (Section 1.5). For the context of this chapter, one key element is that the Kalman gain is used to decide how much of the new estimated location should rely on a prediction based on the previous location and the user estimated trajectory, and how much should rely on the new (noisy) values. This aspect will take its importance as we inject invalid measurements. For this technique like the previous ones, because the measured distances are noisy [150], using more than 5 to 6 responders offers diminishing incentive, as ranging to each additional RSTA increases the energy and ranging (airtime) cost with a decreasing accuracy gain.

7.3.2 Ranging Attacks

FTM exchanges do not require association or any link security because an ISTA will need to range against multiple AP/RSTAs, but can associate to only AP at a time. An ISTA will thus range against any AP announcing FTM support without further verification, and FTM is vulnerable to AP/RSTA impersonation, the equivalent of GPS' satellite spoofing. 802.11az, the 802.11 Amendment for Enhancements for Positioning that expands FTM, does define a Pre-Association Security Negotiation (PASN), that can be used to protect the exchanges and will be examined in depth in Section 7.6, but we will conclude that PASN, as designed in 802.11az, does not protect well against GPS-like attacks.

Also, as the ISTA is only in control of t_2 and t_3, a vector of attack is obviously to feed the ISTA with invalid ranges by forging custom t_1 and t_4 values. In the implementations we tested, some ISTAs consume the range irrespective of its likelihood in the real world (*e.g.*, a distance of 15 km to the AP). Others use some filtering, but with limited effect. For example, in [151], the authors ignore ranges 50 percent larger or smaller than the range established in the previous burst. As long as $(t_4 - t_1) \geq (t_3 - t_2)$ and the $(t_4 - t_1)$ interval is relatively consistent from one sample to the next, the ISTA will use the returned numbers.

7.3.3 Position Attacks

Another common point is that all methods use both the distance and the location (LCI) returned by the RSTA. Thus an attack equivalent to invalid (t_4, t_1) values is to send an invalid LCI value for one or more RSTAs. Here again, we have not found an implementation that discards unrealistic LCIs (*e.g.*, one RSTA reporting to be in Sydney Opera House, the others in the Louvre museum). All tested implementations simply do their mathematical and computational best to minimize the error from these various distance and location elements.

7.4 FTM GPS-like attack experiments

We tested these various possible attacks in a FTM deployment. In an open space free from objects (to avoid localities related to obstacles or reflections, and typical of a shopping-mall setting), 5 APs are deployed along a 75-meter walk path. APs are positioned 24.8 meters from each other, at 10.74 meters from the walking path, at 2.9 meters height, in an alternating fashion represented in Figure 7.1. This structure allocates to each AP a 750-square-meter cell, a typical Wi-Fi density in public venues. At each 50 cm interval, the ISTA collects 100 range samples against all detectable APs. The ISTA location is then computed using the three sphere method (method 1), the distance matrix least squares resolution method (with 4 to 6 RSTAs, method 2) and the position estimation based on a Kalman filter (method 3), in Matlab. The ranging tests are run with a Pixel 3, a Compulab ISTA and a laptop running Windows 10 (and Intel AX200

WiFi card) for the ISTA side, and a Google Wi-Fi AP, a Compulab Responder and a Cisco Catalyst 9120 access point for the RSTA side. As results are comparable for all combinations, the Pixel vs Cisco 9120 figures are presented here.

7.4.1 Attack vectors

For an attacker, the different attacks listed in the previous section present consequential feasibility differences:

— **Invalid t_1 and t_4**: in most systems, the timestamp is computed the chipset DSP microcode. Control from the operating system is limited. Without such control, one option is to capture over-the-air an FTM exchange (*e.g.*, using Wireshark), edit the file to change the victim target MAC address and insert the t_1 and t_4 values of choice (*e.g.*, using WireEdit), then use tools like TCPreplay to replay the AP response to the ISTA. This attack requires some level of preparation. Its outcome is to mislead the ISTA on its real distance to the location (LCI) reported by the attacker's AP.

— **Invalid LCI**: the effect is also to mislead the ISTA on its distance to a reported location, this time by providing valid (t_1, t_4) but invalid RSTA location. The LCI is provided by the operating system (*e.g.*, hostapd.conf file in Linux), and is therefore easy to modify, making invalid LCI injection much easier than t_1 and t_4 modification, unless the attacker has access to the DSP microcode.

— **Session hijacking**: the ISTA and RSTA exchange dialog token values. An attacker inserting into a valid dialog between an ISTA and a RSTA, for example to substitute the attacker response to the valid RSTA response, would need to provide the correct token value in the response. Failure to do so would cause the ISTA to ignore the frame. However, the systems we tested do not implement a complex token system. Some use a linear suite $(1, 2, 3,$ *etc.*) Others always use 0 as the token value. Additionally, 2 of the 3 RSTAs tested allow the user to define the MAC address. A simple injection attack is therefore to program the attacker RSTA with the victim RSTA MAC address, and let the local system perform FTM (responding to the ISTA tokens). The effect is ranging confusion, as the ISTA receives different distances (and LCIs) from what the ISTA assumes to be single device.

In the experiments below, we found that t_1 and t_4 manipulation provided similar outcomes as LCI manipulation, but at the cost of a much higher implementation complexity. Thus, the LCI attack outcomes are presented. The session hijacking also provided interesting observations.

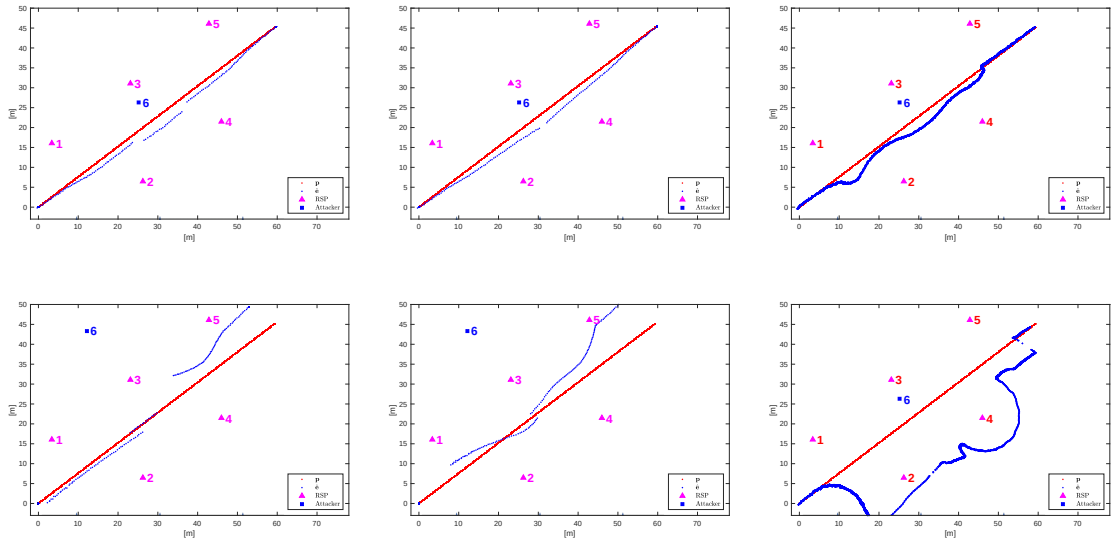


Figure 7.1 – Attacker AP sending 5-m (top), and 35-m biased LCI (bottom), with location computed using three-sphere method (left), matrix resolution (center) and Kalman filter estimation (right). Victim real position is marked p, forged path e, real AP responders with a triangle and attacker AP with a square.

7.4.2 Inserting an Invalid RSTA

Throughout these experiments, an attacker AP (AP6) is introduced. The experiments show that the attacker's AP physical position is not critical for this phase. When AP6 provides RSTA service and valid values, the ISTA computed position matches the actual position, regardless of the computation method, as expected. Next, the LCI sent by the attacker AP is modified to return an incorrect value, as shown in Figure 7.1.

With a small bias (*e.g.*, 5 meters), the ISTA computed position drifts by at most the bias value, and reaches its maximum near the attacker AP's position. Quite naturally, the drift increases with the bias. However, a large bias may cause a filter mechanism to detect the attacker AP as an outlier (providing values not compatible with the position determined from the other RSTAs). Additionally, large bias (*e.g.*, 35 meters) may render the position impossible to determine. With the three spheres technique, the reported positions make that the spheres do not always intersect, resulting in no location results for several points. With an extended Kalman filter, the path becomes incoherent (even if the ISTA continues to compute a position, a normal user would soon identify that one cannot walk in a straight line and yet be on such convoluted path). The reason for such incoherence lies in the way the Kalman filter technique is applied. As the goal is to resolve standard Euclidean distance equations (in the form $\tilde{d} = \sqrt{(u_i - u_y)^2 + (v_i - v_y)^2 + (w_i - w_y)^2} + b_i$) the extended Kalman filter technique seeks derivatives of the state matrix and the distance measurement matrix before applying the

Kalman process, and determining the relative weight given to the observation and the prediction when computing the next likely position (see Section 1.5). The filter becomes better at predicting the next state when noise is comparable across RSTAs. But when the ISTA switches to a faulty RSTA, suddenly providing incoherent values (as it is the case with this attack model), then the measurements suddenly largely exceed the range of expected errors. The Kalman gain soon increases the weight of measurements over prediction, but the reactive process, coupled to the fact that derivatives are sought from ranging against a different contributor, causes the resulting state matrix to display sudden changes of direction as the new contributor data becomes dominant. It is worth noting that this effect is known for the Extended Kalman Filter (it is not an optimal estimator in cases when one AP provides values beyond the expected noise range).

7.4.3 Spoofing valid RSTAs

In this phase, the attacker impersonates a valid RSTA MAC address. Such action causes both the valid RTSA and the attacker RSTA to respond to the ISTA queries for ranging. A confusion attack would have the attacker RSTA respond with different parameters than the valid RSTA. Depending on the implementation, such response might cause the ISTA to ignore the valid RSTA parameters, or to fail during the ranging exchanges (as the exchanges do not match the parameters that the ISTA recorded).

A more interesting attack is to let the valid RSTA respond, then have the attacker RSTA insert FTM ranging frames within the valid exchanges. The ISTA then undergoes more exchanges than it expects (*e.g.*, receiving 16 frames in a burst where it expects 8). On all observed ISTAs, the client considers the exchanges that can take place within the defined burst duration (and ignores RSTA FTM messages beyond the expected end of the burst), even if the burst contains more than the expected count of exchanges. On all observed RSTAs, the FTM exchanges also stop at the end of the burst duration (but each RSTA does not attempt more than the expected count of exchange within each burst). Thus, it seems that current implementations allow the ISTA to perform more exchanges than agreed upon, within the limit set by the burst duration. The net effect is that the ISTA receives half its ranges from the valid RSTA, and half from the attacker RSTA. The ISTA retains the shortest distance in the burst, as explained in Section 4.4. All observed ISTAs also do not consider the LCI as a fixed object. In other words, in the ISTAs we observed, each return to the channel causes the ISTA to query for the LCI again as part of the FTM exchange. In this circumstance, the ISTA receives 2 sets of LCIs. All the observed ISTAs record both received LCIs in their logs, but continue to display the second one for the burst analysis.

Therefore, in order to be preferred to the valid RSTA, the attacker first needs to make sure that the distance offered to the attacker RSTA is less than to the valid RSTA (as the ISTA retains the shortest distance in the burst). The attacker can hard code that distance in the t_1

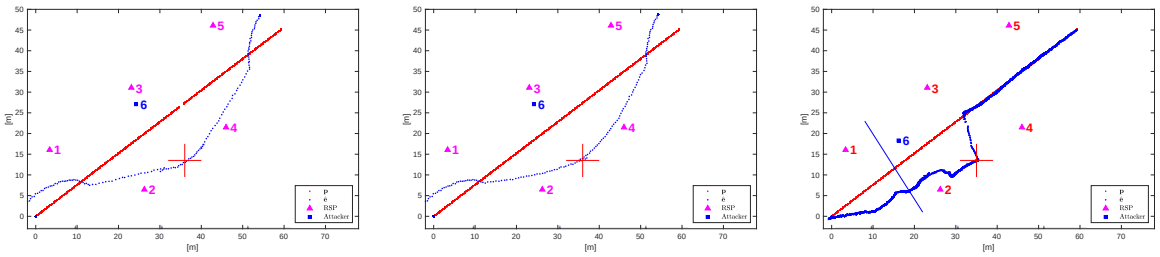


Figure 7.2 – Leading a victim toward a target point, with the three-sphere method (left), least squares (center) and Kalman filter (two LCI set technique, right).

and t_4 values, or make sure to position the attacker AP closer to the victim walking path than the real AP. In a public venue where walking paths and valid APs positions are commonly known, choosing the right AP to target (*e.g.*, an AP away from the public walking path) makes that phase trivial. Then, the attacker can modify the LCI at each exchange to lure the victim away from the intended path. Replaying the LCI value several times can ensure that the attacker's value is received after the valid AP LCI in each burst, and thus preferred.

With these precautions in place, the effect observed is that the attacker's AP is substituted to the valid RSTA for most bursts. Although the ISTA ranges to both RSTAs, the values from the attacker are statistically retained more often, thus effectively resulting in valid AP suppression and replacement.

7.4.4 Leading the victim to a target location

We now show that careful parameter injection can be used to lead the ISTA to a location of the attacker choosing. Such possibility may have dire consequences in settings where FTM is used for business-critical navigation (computer-on-wheels in healthcare, guided robots in factories, *etc.*)

Naturally, each location computation technique incorporates a different set of noisy distances and parameters from which location is computed. Therefore, an efficient attack should account for the type of localization equation in use by the victim device. In most cases, the victim will not choose the formula, but use the method incorporated in the operating system or the navigation app of choice. It is expected that a small set of large actors will provide the bulk of the multilateration algorithms. Thus, it is likely that knowing the victim device will allow the attacker to determine the localization method in use.

To illustrate such approach, the attacker apparatus is positioned near AP3 (marked AP 6 in Figure 7.2). The apparatus is comprised of 3 Compulab RSTAs, set to AP1, AP3 and AP5 channels and MAC addresses respectively. The apparatus is closer to the walking path than AP1, AP3 and AP5 between the marks on the path in Figure 5.6. Because the attacker now impersonates 3 systems, and because the number of expected contributing RSTAs is limited as

explained in Section 7.3.1, the outlier filtering system fails (no single isolated outlier AP).

The goal of the attacker is then to provide ranging values pushing the victim toward a point between AP2 and AP4 (marked with a cross in the figure). The task becomes trivial with Equation (1.10) and the three-sphere technique. To build LCIs with that method, any impersonated AP is chosen to assume the position at the origin and provides any arbitrary reference LCI value. Then, from the known distance from the apparatus to the point where the victim should be led (and its matching coordinates y), the system of equation can be solved to find the LCIs to be announced by the other impersonated APs. From Equation (1.10):

$$u_j^2 - 2u_j u_y + d_i^2 - d_j^2 = 0. \tag{7.1}$$

The only unknown is u_j, which can be found as:

$$u_j = \frac{2u_y \pm \sqrt{4u_y^2 - 4(d_i^2 - d_j^2)}}{2}, \tag{7.2}$$

where $(0, 0, 0)$ is the LCI of choice for the first impersonated AP. Out of the 2 possible solutions, choosing the smaller u_j limits the risk of outlier detection. Here, $v_j = v_i = 0$. Then:

$$v_k^2 - 2v_k v_y - 2u_k u_y + d_i^2 - d_k^2 + u_k^2 = 0. \tag{7.3}$$

Both u_k and v_k are unknown in this system, but by expressing v_k as a function of u_k:

$$v_k = \frac{2v_y \pm 2\sqrt{v_y^2 - d_i^2 + d_k^2 - u_k^2 + 2u_k u_y}}{2}. \tag{7.4}$$

The attacker can choose an arbitrary value of u_k so that:

$$u_k^2 - 2u_k u_y - v_y^2 + d_i^2 - d_k^2 \leq 0. \tag{7.5}$$

In other words, u_k is an arbitrary number in the range $[p, q]$ that satisfies $pq = -v_y^2 + d_i^2 - d_k^2$ and $p + q = -2u_y$. As all other values are known to the attacker, the determination is a simple factoring exercise.

Once the attacker's APs are positioned, the victim system will measure noisy distances, but the optimal solution will lead the victim toward the intended point. As the computation includes the contribution of valid APs at some stage of the path, the location result is increasingly biased toward the attacker's APs data as the victim advances on the path, as can be seen on the left of Figure 7.2.

The same logic is applicable to the multi-sphere (least squares) technique. Let's suppose the more complex case, 6 contributors, including 3 valid APs (i, j, k) and the three attacker APs

(a_1, a_2, a_3). In Equation (1.16), for each AP under the attacker control, y is the intended target destination (we write y_i), while for each valid AP, y is the true victim position (we write y_t), which can be known by deciding the real location of the victim when the location app computes the intended target position. The distances $\tilde{d_i}$ to all contributing APs are known from the (t_1, t_4) values measured at that real position. Thus, we can rewrite the minimization goal as:

$$\begin{aligned} \min[&(||y_t - i|| - \tilde{d_i})^2 + (||y_t - j|| - \tilde{d_j})^2 \\ &+ (||y_t - k|| - \tilde{d_k})^2 + (||y_i - a_1|| - \tilde{d}_{a_1})^2 \\ &+ (||y_i - a_2|| - \tilde{d}_{a_2})^2 + (||y_i - a_3|| - \tilde{d}_{a_3})^2] \end{aligned} \tag{7.6}$$

The only unknowns are therefore the announced positions i of the attacker's APs that minimize the error at the target position. A simple solution is to insert two arbitrary positions, and let the system solve for the third. This solution is functional, but might output a LCI for the third AP far from the others. In order to minimize the risk of outlier detection, an efficient approach is to wrap this algorithm into another gradient descent structure. With this method, the first 2 APs are set at initial arbitrary positions, the third AP location is found, then a loop runs, where step-wise changes to the initial 2 AP positions are made so as to minimize the differences between all 3 resulting LCIs. Once such system is found, the victim can be led to the intended location, as displayed in the center part of Figure 7.2. The Kalman filter case is slightly different. The solution proposed for the least squares approach above is also functional with the Kalman method, misleading the ISTA to compute its position as the target location. However, as can be seen in the lower right part of Figure 7.1, the effect is also to cause an incoherent trajectory (reporting a sharp turn while the user is walking along a straight line). To avoid this risk, an additional step is to identify intermediate positions where the attacker would want to smoothen the curve, determine the optimal victim position at that point, and generate a set of attacker APs LCIs accordingly. The attacker announces a first set of LCIs (or timestamps), then a next set as the victim passes key points. The effect of such modification can be seen in the right part of Figure 7.2, with an initial set of forged LCIs, then a second set announced as the victim passes the traversal (blue) line. The process needed for such attack in the Kalman Filter case is similar in concept to the Least Square cases, where a loop computes the LCI set that minimizes the differences between the spoofed APs announced positions, but is more laborious, as the range at which the announced LCIs are efficient in fooling the victim is limited. It is likely that the effort involved will match the value of leading the victim to the target point without raising suspicion.

7.5 Limitations of Existing Protection Techniques

FTM is not the only location solution relying on distances, and not the only 802.11 technology where data validation is required, so one could think that the problem is well-known and possibly solved through these other techniques. However, we will now see that thes protections do not apply well to FTM. 802.11 also does not offer viable solutions in its current form.

7.5.1 Applicability Limitation of GPS-attack Solutions

In particular, GPS-attack protections that apply fusion techniques prove inefficient for FTM. These techniques use a primary source, and one or more secondary (less accurate or reliable) sources. The values coming from the secondary sources are used to spot outliers in the primary source, but then the primary source data is still used to contribute to the location computation. In other words, the secondary sources augment, but do not replace, the primary source. GPS is commonly not available inside buildings, and cell tower triangulation is also usually not possible. The STA can collect information from movement sensors (gyroscopes *etc.*), but their validity dissolves with distance [152], and a free-form indoor walking path gets complex very fast (faster than outdoor, on guided tracks like a road or sidewalk). Therefore, without an additional reliable source of reference, the STA cannot compensate for a progressive and continuous injection of invalid parameters. This difficulty is amplified by the fact that Wi-Fi ToF values are usually noisy. With obstacles and multipath, there is no correlation between the AP signal strength and the calculated distance, and the precision of measurement is low. Therefore, fusion is not a sustainable solution for FTM.

Other GPS attack protection techniques that rely on signature and fingerprinting are also not transferable. With billions of Wi-Fi APs on the planet, and the assumption that a core use case for FTM is a human with a handheld device (*e.g.*, a smartphone) in an unfamiliar venue, the designers of FTM have not implemented any validation technique, and there is no global database that the ISTA could use to verify the identity of each AP in range.

Therefore, although the attack exposure of FTM is similar to that of GPS attacks, the particularities of FTM makes that the solutions for GPS-attack mitigation cannot be transparently transposed to the FTM case.

7.5.2 Limitations of IEEE 802.11 Solutions

Another avenue to explore could be to leverage 802.11 existing solutions. For example, 802.11 association can be followed with a mutual authentication mechanism, where both the STA and the infrastructure prove that they know a shared secret (which is a proof that the AP is legitimate). It could be suggested to use this mechanism to protect the FTM exchanges. However, besides the difficulty that each ISTA would need to be configured with credentials for

the local SSID, the entire procedure commonly takes 300 milliseconds (and can take more than one second when the 802.1X/EAP part relies on an external RADIUS server). Once association is complete, the STA can leave the AP cell and re-join with a re-association process, which delay is much shorter (80 to 100 ms).

This overhead can cause several problems. A first immediate issue is time consumption. In the FTM negotiation, the ISTA and the RSTA agree to meet on the channel at pre-determined points in time to exchange FTM frames in a burst. The designers understood that either side may be delayed. For this reason, the burst duration can last up to 128 ms. During that interval, each side will attempt to be on the channel and perform ranging. The ISTA will listen on the channel, waiting for the RSTA first FTM frame of the burst. The RSTA will send the first FTM frame, and wait for the first acknowledgement response from the ISTA (then retries repeatedly until the end of the burst when no acknowledgment is received). The ISTA and the RSTA also agree on an inter-burst interval, but if during that interval the ISTA goes to another channel and does not know if it will spend there 80 ms, 300 ms or more than a second before coming back, failed bursts are guaranteed to be a common occurrence. This would not be a reasonable design.

Another issue is the processing cost on the RSTA. The ISTA and RSTA negotiate a number of bursts (between 1 and 16384). If the ISTA reassociates for each new burst, then the AP potentially has to manage 16384 reassociation procedures per client and FTM session, each with a computing cost related to key verification (not to mention 80 ms inserted delay each time). A few stations performing rapid-fire ranging with multiple APs are susceptible to exhaust the APs computing resources. For these reasons, relying on 802.11 authentication has never been considered as a viable solution by the 802.11 designers of FTM.

These limitations do not mean that the ISTA cannot associate and perform FTM. The ISTA could associate with one AP, perform FTM with that AP, then go to other channels and perform unassociated and unprotected exchanges with the other APs. Quite obviously, this mode would offer very limited protection.

802.11az PASN is also an interesting direction to explore. PASN exists in two modes. One mode supposes the existence of pre-existing shared keying material between the ISTA and the RSTA. In that mode, the ISTA and the RSTA then form what 802.11 calls a Robust Security Network Association (RSNA) authentication, in that they create a secure link and authenticate each other. This mode is efficient to protect from spoofing, as an attacker AP would not have the valid keying material, and would therefore not be able to insert frames in the exchange between an ISTA and a valid RSTA.

One clear limitation of this mode is the requirement to pre-populate the ISTA with the keying material. This is easily possible in a private setting (*e.g.*, a factory with a single WLAN system). However, in a public venue, this constraint becomes difficult to overcome, even for a

device that is expected to be familiar with the venue (*e.g.*, a self-driving shuttle or delivery robot in a shopping mall). It is likely that there will be many APs and SSIDs (for example, an American shopping mall commonly includes 4 or 5 anchor large stores, and several dozens of small shops in between). APs may communicate over the back-end, but it is unlikely that the ISTA would have credentials for all stores and SSIDs. Even in a private setting, it is common to find zones with different SSIDs (for example department names in a hospital), forcing the implementers to configure multiple profiles on the device, even if the APs communicate with the same backend infrastructure (and thus even if the device credentials would be the same for all SSIDs).

Thus RSNA PASN provides an interesting, but incomplete direction for FTM protection. PASN also exists in a non-RSNA mode, where no initial keying material exists between the STA and the APs. In that case, a protected, but unauthenticated link is established ad-hoc between the STA and each AP, creating what some call "trust at first sight". The frames that the STA and the AP exchange (including FTM) are then encrypted. In this scenario, (t_1, t_4) and possibly LCI values can be protected from eavesdroppers. A similar unauthenticated but encrypted link is formed between the ISTA and each and any other AP with which the ISTA needs to range. This mode simplifies the deployment (no pre-existing keying material required), but does not include any validation structure. Thus, an attacker can pretend to be an additional AP in the system, or can impersonate (spoof) a valid AP. The ISTA would then establish an unauthenticated but encrypted connection to that rogue AP, with no means to know that the AP is invalid and send forged (t_1, t_4) and LCI values. The only protections that this mode offers are exchange obfuscation (an observer cannot see the values exchanged between the ISTA and RSTA) and session hijacking protection (an attacker can spoof the identity of a valid AP at the time of a new FTM session establishment, but cannot insert rogue values once a session has started with a valid AP).

7.6 A Crowd-Wisdom FTM Attack Exposure Mitigation Solution

Clearly, RSNA PASN is a good option for environments with a single SSID and a device that is familiar with the venue (and thus can be programmed with keying material). However, in environments with a single WLAN infrastructure but multiple SSIDs, or multiple neighboring WLAN infrastructures, or environments with which the device is not familiar (thus does not possess any keying material), RSNA PASN shows its limitation, and non-RSNA PASN offers no real protection.

For these environments, it is difficult to establish trust when no APs are known. However, we propose a mechanism to reduce the likelihood that a given contributing AP would be an

attacker's. In a setting where there is no authentication, and no protected exchange, all exchanges are open to all abuses, and we do not believe that an easy remediation is possible. However, if an initial protection can be built between an ISTA and a first AP, we believe that partial further protection becomes possible. We write "partial" because without authentication, the protection stays limited, but it can be implemented in a way that makes the attack less trivial, and thus less attractive to an amateur attacker. Therefore, we next examine partial remediation options for cases where the ISTA is not associated to a network but can establish a protected link to APs.

7.6.1 Modified FTM AP Sorting Algorithm

One key assumption for this method is that the venue presents several APs announcing the same SSID. These APs communicate with each other (either in a mesh fashion, or because they are connected to the same management system, *e.g.*, a WLAN controller). Other APs may be present in the same RF space, forming individual islands of Wi-Fi coverage around individual or shared SSIDs. In this environment, it is unlikely that the attacker can become the dominant system, *i.e.*, deploy with impunity more APs than the valid network, without causing multiple rogue alarms on the main system and being detected. Most managed Wi-Fi systems can identify AP MAC address spoofing or SSID impersonation. An attacker can deploy one or a few temporal (physical or virtual) RSTAs for the duration of the attack, but is unlikely to have deployed a system larger than the venue legitimate WLAN infrastructure.

When initiating an FTM session, the ISTA needs to first establish a list of possible RSTAs. The ISTA starts by scanning all channels (standard 802.11 discovery) and establishes the list of channels and AP MAC addresses (Basic Service Set identifiers, BSSIDs) offering RSTA services. In traditional FTM, the ISTA would then directly start ranging against the first AP (*i.e.*, the AP with the strongest signal, or the AP on the lowest channel in the band). We now propose instead that the ISTA sorts the BSSIDs by signal level, then considers BSSIDs within the same signal level (RSSI) range (*e.g.*, within 6 dB of each other) as being a single system (for a reason that will become apparent later). We then propose that the ISTA operates a second sort, grouping the BSSIDs by their announced SSID. Thus, we propose that the ISTA starts by ranging against the RSTAs representing the largest system (max BSSID count for the given SSID). This first step is precautionary and does no offer any strong protection, but decreases the likelihood of ranging against a lone attacker, as will be seen.

We then suggest that the ISTA selects one BSSID at random within the largest SSID group, and first establishes a protected link to the RSTA, using PASN (likely, the non-RSNA PASN flavor). The goal of such link is to protect the exchange from eavesdropping. Although the largest group is likely to be a valid system, the attacker may impersonate one AP of the valid system, and thus it is possible that the first AP may belong to the attacker.

7.6.2 PASN and 802.11r

We then propose an augmentation of PASN, that we call PASN-FT, to allow secure link pre-establishment to other APs. FT, or 802.11 Basic Service Set (BSS) Fast Transition (FT), was defined in the 802.11r-2008 amendment (integrated in the 2012 version of the 802.11 Standard), and is intended for RSNA, fast roaming key exchanges for associated STAs. In this mode, a STA first establishes a secure association (RSNA) with an AP. During that process, the 802.11 choreography allows for AP and STA mutual validation (as they both have to prove to the other that they have the right temporal keying material). The 802.11-FT process incorporates two major changes to the previous 802.11 association process:

— The AP advertises a Mobility Domain Element (MDE), which is a string representing the domain, *i.e.*, the set of APs between which fast transition will be possible. The string is commonly an arbitrary set of characters (it does not need to have a meaning, and just needs to be common between APs participating to the same group). When roaming, the STA selects APs that advertise the same MDE.

— 802.11r establishes a new key hierarchy. Upon a STA first association, the WLAN infrastructures establishes a first Pairwise Master Key (PMK-R0). This key is derived from the Master Session Key (MSK), which is formed on the client side and the infrastructure side through the regular authentication process defined for 802.11. In a non-FT mode, the PSK is directly derived from the MSK (the PSK is the first 256 bits of the MSK). With FT, the PMK-R0 is derived by also integrating other elements, such as the value of the domain in the MDE, the SSID name, the STA MAC address and the identifier of the first entity with which the client establishes this first keying material (this can be the first AP MAC address, or a value for a centralized WLAN controller; this entity is later identified as the holder of the PMK-R0, or R0KH-ID). Then, for each AP, a PMK-R1 is established, built from the PMK-R0 value, the MAC address of the client and the MAC address of the target AP. When a STA needs to establish a communication with a first AP, it is provided the elements it needs to compute PMK-R0 from the MSK (that the client should be able to derive during the authentication phase, from its credentials or a pre-shared key). The client can then compute the PMK-R1 to associate with any AP in the domain, if the AP MAC address is known. From the PMK-R1, other keys are derived (temporal unicast keys).

This key hierarchy allows BSS-FT to enable a fast transition mode. When a STA needs to roam to a neighboring AP, a non-FT mode would mean that the STA should deassociate from the current AP, associate to the next AP, then undergo the full authentication exchange in order to derive a new PMK. This process can take a long time, as detailed in Section 7.5. With FT, the keying material required for association to the next AP can be derived while the STA is still associated to the first AP. BSS-FT allows two modes for that process:

— With the Over-the-air mode (OTA), the STA sends to the next AP an FT authentication request, that includes the PMK-R0 Name, the MDE, and a Fast Transition Information Element that includes the R0KH-ID. These elements allow the next AP to determine if it can build a PMK-R1. If the answer is positive, the next AP responds with an FT authentication response, that includes the elements the STA needs to derive the PMK-R1 value for the next AP.

— With the Over the Distribution System method (Over-the-DS), the STA first identifies the target next AP, then sends to its current AP an action frame requesting the establishment of keying material with the target AP. The request also includes the PMK-R0 Name, the MDE value and the R0KH-ID. The current AP should relay this request over the wire to the target AP. Similar to above, the target AP should determine if it can build the PMK-R1, and reply through the current AP (over the wire) if the answer is positive, with a frame containing the elements the STA needs to derive the PMK-R1 value for the next AP.

In both cases, the STA is then ready to communicate securely with the next AP. At any time, the STA can deassociate from the current AP, and send a reassociation request to the next AP, mentioning the PMK-R1 Name, the MDE, the R0KH-ID and the MAC of the target AP. The STA also mentions a message integrity check (MIC) that proves that the STA has the right keying materiel. The next AP replies in kind, and data communication can resume immediately.

7.6.3 PASN FT

802.11 BSS-FT was intended for associated STAs. However, we propose to adapt FT principles to the PASN case by adding to PASN exchanges the FT elements that do not strictly force the STA to undergo an association.

Thus, in this method, all APs part of the same infrastructure include the MDE in their probe responses and beacons. This informs the STA about which APs are claiming to be part of the same domain.

In PASN, the STA seeking to establish a secure link sends a first PASN 802.11 authentication frame to the first AP. The frame may include base Authentication and Key Management (AKM) parameters (if there is a pre-existing keying material that the STA can use), but also includes an ephemeral public key that the STA wishes to use (the STA also generates internally the matching private key). We add the MDE value to this frame.

In PASN, the AP responds with a second PASN authentication frame that includes the AP temporal public key (the AP also generates internally the matching private key), and optionally base AKM parameters (if the STA included them). We also add the MDE value to this frame.

In PASN, the STA then responds with a third PASN 802.11 authentication frame, that serves as an acknowledgment to the exchange. We reuse this frame unchanged.

At this point, the STA has established keying material with the AP and can undergo protected exchanges. With PASN, the secure link is established with a single AP at a time (the AP with which the STA wishes to communicate). We augment this procedure by allowing an Over-the-DS key pre-establishment with other APs. An OTA mode would also be possible, but presents limited added value. From an airtime consumption standpoint, the amount of frames to exchange would be equivalent to a direct PASN exchange with the next AP (thus bringing no airtime consumption, and no process time consumption, advantage). From a security standpoint, both the Over-the-DS and OTA modes prove that that the first AP has a trusted backend relationship with the second AP, thus that they belong to the same infrastructure. The OTA mode thus does not surface specific advantages.

The Over-the-DS PASN FT is illustrated in Figure 7.3, and works as follows:

— When in need to communicate securely with a second AP, the STA sends to the current AP a PASN FT authentication frame wrapped in a protected (robust) action frame. The frame resembles the PASN first frame, but also includes the MDE, and a Fast Transition Element (FTE) that includes the target AP BSSID (MAC address), and also the client intended MAC address (called S0KH-ID) for exchanging with the next AP. This last element offers an interesting additional protection. The STA can decide to use a different MAC address for its dialog with the next AP (using locally administered MAC addresses). As the frame is encrypted with the current AP public key, this value is obfuscated from an eavesdropper's view. This way, the infrastructure can keep track of the client queries while observers do not see a single STA.

— The current AP forwards this frame over the backend to the next AP. The next AP responds over the DS with a frame that resembles the PASN second frame, but also includes the MDE, the second AP MAC address, the target STA current MAC address (used by the STA to send the first PASN FT frame to the current AP) and the S0KH-ID. The frame also includes the FTE that includes a timeout value. This value tells the STA the interval for which the current keying material will be valid.

— The current AP relays the frame to the STA. The STA validates the frame components, and returns an acknowledgement frame. The frame resembles the PASN third authentication frame, but also includes the FTE that mentions the next AP MAC address, and the client intended MAC address (S0KH-ID).

At this point, the STA is ready to communicate securely with the next AP. Within the timeout interval specified by the next AP, the STA can switch to the next AP channel and directly send protected data (using as a source MAC the S0KH-ID value). The STA can this way pre-establish secure links with a multiplicity of APs, then switch to their respective channel in turn to proceed to protected FTM exchanges.

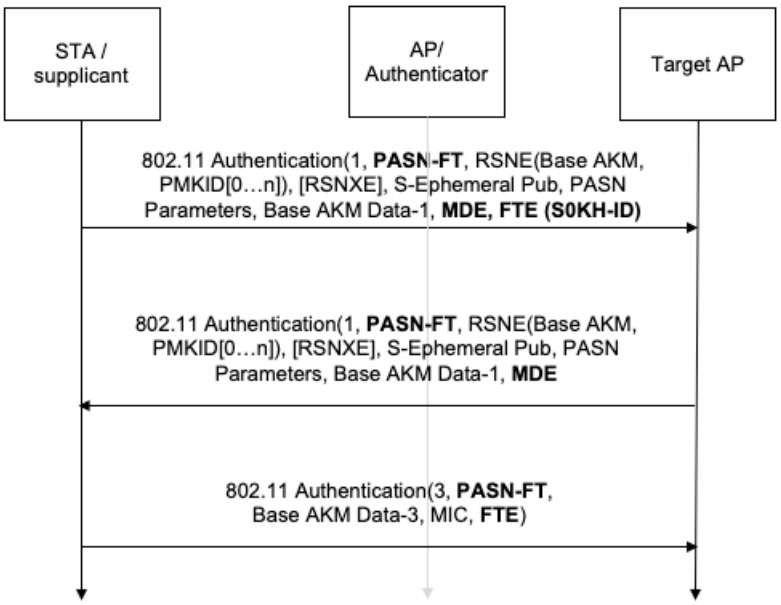


Figure 7.3 – Proposed PASN-FT choreography. The elements that diverge from strict PASN are labelled in boldface.

This process does not guarantee that no attacker will insert in the exchange. However, this methods greatly limits the risks, as four permutations and scenarios are possible:

Current AP \ Next AP	Legitimate	Attacker
Legitimate	(1)	(2)
Attacker	(3)	(4)

1. The APs can communicate over the backend and the STA can successfully establish a secure connection to the next AP.
2. The legitimate AP does not have a trusted relationship to the attacker AP, and rejects the PASN-FT request from the STA.
3. The attacker AP may attempt to pass the PASN FT request to the legitimate AP, but they do not have a backend trusted relationship. The attempt fails and the STA does not receive a response from the next AP.
4. The APs may communicate over the backend, and the STA may be able to successfully establish a secure connection with the next AP.

It should be noted that, because the STA merges RSTAs with the same RSSI as indicated above, scenario (4) only succeeds if the attacker has positioned 2 different physical APs. Scenarios (2)

and (3) do not directly allow the STA to determine that one AP is illegitimate. From the STA viewpoint, the PASN FT process failed, possibly because one of the APs is illegitimate, or because both APs are legitimate but in disjoint systems (a less common, but possible case).

However, as the process repeats with more APs, the STA surfaces groups of APs that have a backend trusted relationship, and outliers APs (AP_O) that are not trusted by others (because requests made to an AP_O to PASN FT toward other APs will usually fail, and requests made to other APs to PASN FT toward an AP_O will also usually fail, unless the other AP is also an AP_O). The STA can then use these groups of largest APs having a trusted relationship as the set of RSTA from which location is computed. These are likely to be legitimate, unless the attacker is the dominant system in the venue.

7.7 Experimental Validation

We tested this method in two different environments. The first setting is similar to the attack test setup described earlier (5 legitimate APs, and the attacker emulating one to three APs), and leads to the following observations:

1. The attack fails in 100% of cases where the attacker presents a single AP. This is likely because that AP cannot form a group large enough to be usable (and of course the AP also fails to establish PASN-FT with the other APs).
2. When the attacker emulates three APs, the attack fails if the APs are all emulated from the same physical system (*e.g.*, virtual APs on the same laptop, or physical APs at the same location). Despite RF signal stochasticity, all APs then present an RSSI in the same range and get merged by the filtering procedure (thus leading the STA to the same conclusion as above).
3. When the attacker deploys 3 non-co-located APs (see Figure 7.1 bottom), the attack fails for any localization method using 4 APs or more (center, matrix-based localization). This outcome is expected. For localization method using 3 APs, the attacker system becomes self-sufficient and may partially succeed. With the 3-sphere and matrix methods, the STA temporarily follows the attacker's data, then suddenly jumps back to the correct trajectory as soon as contributors from the valid system are introduced (left). When using Kalman filtering, the slide toward the correct path is progressive, as the system arbitrates between the observed and the computed values (right).
4. The above attack succeeds only if the attacker system is within the first 3 APs to be attempted by the STA, and if the attacker system is large enough to be entirely sufficient for the STA calculations (*i.e.*, 3 APs for the 3-sphere methods, and up to 6 APs for other methods). The STA then forms 2 groups of non-compatible sets (the legitimate APs, and

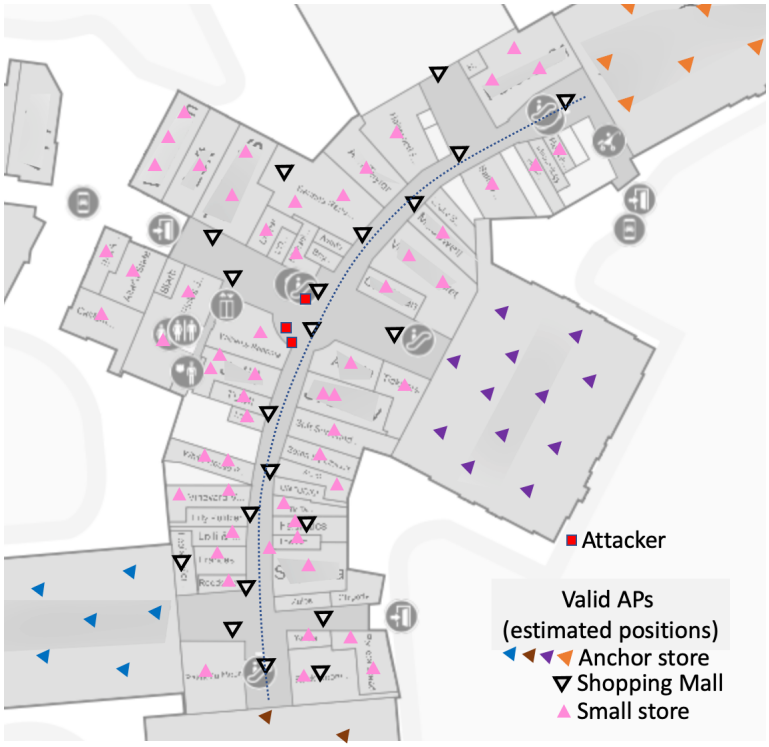


Figure 7.4 – Attack attempts with PASN-FT.

the attacker's APs) and can then randomly consider the location by the valid AP set, or the attacker's AP set.

The second setting is a shopping mall with 4 major store anchors and a multiplicity of smaller stores, each with an individual SSID. The mall is set on two floors, and Figure 7.4 represents the ground floor. A user walks along the main corridor path, from bottom to top (represented by a dashed line).

In this environment, without changes, the attack fails in all attempts. One likely reason is that, at any point of the path, 9 to 11 APs can be heard that form groups larger than 4 APs, from a combination of the main mall Wi-Fi and one of the anchors'. In such environment, the attacker is unable to establish a system large (and distributed) enough to compete with one of the valid groups.

Forcing the ISTA to ignore the main mall Wi-Fi (supposing a targeted denial of service) does not allow for an attack vector either, because at least one of the anchors' Wi-Fi can be detected from any location along the path (commonly with 4 APs or more). In some areas (mid-point on the walking path), only one anchor's SSID is detectable, and its APs are all in the same direction (to the right). Location precision dilution occurs in this zone if only the anchor's SSID is used. Adding the smaller stores RSTAs restores the precision. Even when an individual store does not allow for PASN-FT with others, the anchor's RSTAs serve as a reference and the attacker is

also identified as an outlier (when injecting forged LCI or (t_1, t_4) values causing an error larger than that resulting from computing location with valid RSTAs).

Forcing the ISTA to also ignore the anchors' Wi-Fi systems provides only a mild attack vector. Although each store display a specific SSID, in effect, many of the smaller stores use APs managed by the mall (and therefore would be displaying the same MDE value), even as most stores complement the system with home or small-business grade APs. These other APs may not communicate with one another over the wire. It is only when removing all small store APs that communicate over the wire that the attack succeeds. However, such a dramatic scenario (all major stores, and the entire shopping mall main WLANs are disabled, only isolated single-AP SSIDs remain) is unlikely in a real environment.

7.8 Conclusion

In this chapter, we have shown that 802.11 FTM is vulnerable to ranging and location attacks. As the client does not know in advance the APs, and as the exchanges are neither authenticated nor protected, an attacker can easily insert an additional AP that provides invalid ranging or position (LCI) information. The client has limited ability to distinguish the attacker's from valid APs, and tends to integrate the data provided by the attacker, if it is not excessively implausible, into its location computation.

The existing IEEE 802.11 Standard, along with a modification of the PASN protocol to allow for fast transition between APs, can be used to mitigate the attack exposure. The effect of these simple changes is to turn the attack from "trivial" to "challenging". There is still a weak vector left, where the attacker is one of the first APs picked by the ISTA and has deployed a system as large as the main WLAN. Further modification of the technology, where deeper key exchanges occur between the client and a trusted infrastructure, will be needed to solve this last limitation.

Conclusion

Concluding Remarks

The explosion of smartphones and other personal devices has changed the way location is consumed. A device in hand, users expect accurate navigation outdoor ("blue dot"), and have extended this expectation to the indoor environment. With most people carrying a personal device, businesses can track individual key resources, or obtain statistics on foot traffic, all in near real time.

However, these possibilities are also met with new constraints, where the location of a personal device steps into privacy territory. Therefore, a modern localization technology should offer the incentive of a blue dot for the end user, but with the guarantee of privacy protection, letting the device share its location only through an opt-in mechanism. The technology should also offer advantages to the infrastructure owner, allowing for the location of assets where and when needed, and anonymized foot traffic patterns where required.

In this study, we saw how outdoor localization techniques solved these challenges. GPS only focuses on the blue dot, and therefore respects privacy while providing an accuracy of a few meters. However, it is often said that GPS was an accident of History, where a system that was designed for a very specific and small set of customers (the US army) was extended to the general public because of fortuitous circumstances. Thus GPS only benefit the end user. By contrast, LTE technologies offer location for both sides, and have implemented multiple techniques to solve the various use cases. It may not be entirely technically true that all requirements of a modern localization protocol are fulfilled, but the difficulty for an attacker to insert into a licensed spectrum makes that privacy challenges are dimmed in the LTE world.

As the user enters a building, GPS and LTE may start suffering from performance degradation. Other techniques then need to be added to maintain location accuracy. Among all techniques, those that rely on RF communications present a great potential, primarily because they are already being deployed for other purposes (*e.g.,* data communication). Naturally, 802.11 is a fundamental candidate, as most indoor environments where smartphones are likely to be found also are likely to have an 802.11 setup. Despite a long history, the indoor localization techniques based on 802.11 exchanges have become challenged. The efficiency of the blue dot solutions are limited by the infrastructure short life-cycle. The localization techniques use by venue owners were using frames that clients send less and less, making infrastructure-based location an imperfect technique in search of a replacement option. In this context, FTM appears as a great

candidate. Developed in the last few years, the protocol could benefit from the 21^{st} century requirements for location determination. Based on 802.11, the protocol has a great potential for wide adoption.

However, we saw that the protocol suffers from multiple design limitations. Focusing on the blue dot scenario, the protocol lacked attractiveness for infrastructure owners, and we saw how we helped fill this gap [4] by adding an ranging feedback from the ISTA to the RSTA.

FTM also made the assumption that, in a place where no GPS signal would be available, the geo-location of the APs would somehow still easily be known. This assumption is unfortunately untrue almost everywhere. We thus proposed a solution to extend GPS from the outside [7] [9] [8], to seed location information where it is available, and then have the RSTAs dynamically learn their relative position to one another and automatically learn their geo-location.

Another issue we identified was that FTM provided no guidance on what parameters should be used for each ranging scenario. The outcome of this limitation was that each vendor would implement the parameters they though would be best, resulting in an ability for an attacker to fingerprint an individual station, just by looking at the station FTM exchange pattern. We offered a solution [10] to provide stricter parameter requirements, and today all stations implement similar FTM parameters, the same way, making fingerprinting based on FTM patterns a risk of the past.

A last issue we identified was that FTM exchanges were not, and could not easily be protected in the Standard context were the technology was developed. An effect of this limitation was that an attacker could not only attack FTM ranging and location, but could use these weaknesses to drive an unsuspecting device to a destination of the attacker's choice [11]. We proposed improvements to the 802.11 Standard and a stricter ranging procedure to mitigate this risk. These improvements are proposed for insertion into the 802.11 Standard [5] [6], and we hope that ranging and location attacks will become much harder as a result.

Perspectives and Future Work

At this stage, we believe that FTM has become a safe and precise technology that can be deployed to solve several indoor location use cases. We are happy to see that major infrastructure vendors are starting to deploy support of FTM on their access points.

Yet indoor location is far from being solved. FTM itself still suffers from unaddressed design limitations. For example, there is no procedure to exchange with the client device, so as to offer an option to opt-in. Without such mechanism, receiving the user consent is impossible, and device tracking feedback will face the obstacle of privacy laws in many countries for years to come. In a future work, we plan to suggest the insertion into the 802.11 Standard of such opt-in messaging within the initial FTM exchanges. Additionally, the passive mode supposes

that the RSTA local time is automatically translated into the station time, but we know how this conversion is difficult. Another axis of future work will be to suggest a inter-RSTA clock synchronization exchange structure, and a method for the ISTA to align its time to that of the RSTA constellation.

Passive mode is great in public venue, but FTM does not include a messaging system for the APs to express the nature of the environment where they are deployed. This limitation adds deployment shortcomings. For example, in hazardous and other dangerous areas, there is no mechanism for the infrastructure to express to the station that tracking in real time is mandatory for user safety. Adding such environmental information is also part of our envisioned future work. Last, FTM has no mechanism to protect the AP LCI. Such protection could establish a different LCI accuracy level based on the identity of the requesting station, providing high accuracy LCI to staff and personnel, and only low accuracy to unassociated stations. Thus, although we believe that FTM has reached a point where it is a good solution for indoor positioning, there is still a lot of work to be done to make a "good" protocol a "modern" protocol.

We also think that indoor localization will be solved like in the outdoor case, with a fusion of techniques. Clearly, GPS and LTE signals do penetrate buildings to some levels, but their availability is limited, and deploying relays inside, solely for the purpose of improving indoor localization, sounds like an expensive proposition. Thus it is likely that other indoor technologies will continue to be developed, and will offer methods that will complement FTM to provide a good location experience. Among them, BLE has brought a lot of promises, and its adoption in the 2010 decade has been stellar. However, the requirements to manage BLE tags battery (and the fact that these tags can easily be stolen) has limited the adoption. The development of BLE support on Wi-Fi APs (with directional virtual BLE tags) may be an interesting alternative. In parallel, UWB has long promised great accuracy, down to the 10-cm scale. However, UWB (based on IEEE 802.15) has not been widely deployed for purposes other than location. The cost of such an overlay deployment has limited the adoption of this technology to specific verticals. The growing support for UWB on customer smartphones (initially for peer-to-peer exchanges) may change the dynamic of the market. It may very well be that UWB chips start becoming common on Wi-Fi APs as well. At that time, a lot of work will become possible to realise the fusion of technologies. BLE, with its ability to provide proximity, could be a first level to augment the accuracy of FTM ranging. Then, in zones where high accuracy would be required, UWB could be selectively enabled to zoom in on the user exact location. The process by which each technology is activated or paused, is used as a primary or a complementary technique for location, the exchanges between the infrastructure and the end device, and the algorithm required to provide the best location ("just accurate enough") are still to be designed, and we are starting to investigate such design. We hope to be among those who will tell how these problems were solved, to a generation that will not know that indoor localization was once a challenge.

BIBLIOGRAPHY

[1] H. Ye, T. Gu, X. Tao, and J. Lu, « F-Loc: Floor localization via crowdsourcing », *in 2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, 2014, pp. 47–54. DOI: 10.1109/PADSW.2014.7097790.

[2] J. Henry, *CCNP Wireless IUWMS Quick Reference*. CiscoPress, 2010.

[3] E. Lindskog, N. Kakani, B. R., J. Lansford, and J. Rosdahl, *Client Positioning using Timing Measurements between Access Points*, 2013.

[4] W. Q., H. C., M. M., A. T., P. V., R. K.S., N. S., S. K., A. A., and H. J., « Text Proposal for ISTA-2-RSTA LMR Feedback », Tech. Rep., May 2019. [Online]. Available: https://mentor.ieee.org/802.11/dcn/19/11-19-0331-05-00az-text-clarification-on-ista-to-rsta-lmr.doc.

[5] J. Henry, S. Orr, N. Bhandaru, and T. Derham, *FT-Friendly PASN*, 2021.

[6] J. Henry and S. Orr, *Fast Transition for Opportunistic Wireless Encryption (FT-OWE)N*, 2021. [Online]. Available: https://datatracker.ietf.org/doc/draft-henry-ft-owe/.

[7] J. Henry, N. Montavont, Y. Busnel, R. Ludinard, and I. Hrasko, *A Geometric Approach to Noisy EDM Resolution in FTM Measurements*, 2021. DOI: 10.3390/computers10030033. [Online]. Available: https://www.mdpi.com/2073-431X/10/3/33.

[8] J. Henry, Y. Busnel, R. Ludinard, N. Montavont, and I. Hrasko, « Auto-localisation en intérieur à l'aide de mesures FTM », *in CORES 2021 – 6ème Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*, La Rochelle, France, Sep. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03220746.

[9] J. Henry, N. Montavont, Y. Busnel, R. Ludinard, and I. Hrasko, « Sensor Self-location with FTM Measurements », *in 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2020, pp. 1–6. DOI: 10.1109/WiMob50308.2020.9253395.

[10] J. Henry and N. Montavont, « Fingerprinting using Fine Timing Measurement », Nov. 2019, pp. 49–56, ISBN: 978-1-4503-6905-3. DOI: 10.1145/3345770.3356736.

[11] J. Henry, Y. Busnel, R. Ludinard, and N. Montavont, « Ranging and Location attacks on 802.11 FTM », working paper or preprint, May 2021. [Online]. Available: https://hal-imt-atlantique.archives-ouvertes.fr/hal-03241630.

[12] H. Khan, M. N. Hayat, and Z. Ur Rehman, « Wireless sensor networks free-range base localization schemes: A comprehensive survey », *in 2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, 2017, pp. 144–147. DOI: 10.1109/C-CODE.2017.7918918.

[13] M. K., L. M., and H. P., *Towards Smart Surroundings: Enabling Techniques and Technologies for Localization*. Springer, 2005.

[14] S. Sivasakthiselvan and V. Nagarajan, « Localization Techniques of Wireless Sensor Networks: A Review », *in 2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 1643–1648. DOI: 10.1109/ICCSP48568.2020.9182290.

[15] N. A. Alrajeh, M. Bashir, and B. Shams, « Localization Techniques in Wireless Sensor Networks », *International Journal of Distributed Sensor Networks*, vol. 9, *6*, p. 304 628, 2013. DOI: 10.1155/2013/304628.

[16] M. Berber and S. Hekimoglu, « WHAT IS THE RELIABILITY OF CONVENTIONAL OUTLIER DETECTION AND ROBUST ESTIMATION IN TRILATERATION NETWORKS? », *Survey Review*, vol. 37, *290*, pp. 308–318, 2003. DOI: 10.1179/sre.2003.37.290.308.

[17] P. Wang and Y. Luo, « Research on WiFi Indoor Location Algorithm Based on RSSI Ranging », *in 2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, 2017, pp. 1694–1698. DOI: 10.1109/ICISCE.2017.354.

[18] M. A. Cheema, « Indoor Location-Based Services: Challenges and Opportunities », *SIGSPATIAL Special*, 10–17, 2018. DOI: 10.1145/3292390.3292394.

[19] C.-H. Kao, R.-S. Hsiao, T.-X. Chen, P.-S. Chen, and M.-J. Pan, « A hybrid indoor positioning for asset tracking using Bluetooth low energy and Wi-Fi », *in 2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2017, pp. 63–64. DOI: 10.1109/ICCE-China.2017.7990996.

[20] M. A. Al-Ammar, S. Alhadhrami, A. Al-Salman, A. Alarifi, H. S. Al-Khalifa, A. Alnafessah, and M. Alsaleh, « Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms », *in 2014 International Conference on Cyberworlds*, 2014, pp. 245–252. DOI: 10.1109/CW.2014.41.

[21] G. Marini, « Towards Indoor Localisation Analytics for Modelling Flows of Movements », *in Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, Association for Computing Machinery, 2019, 377–382, ISBN: 9781450368698. DOI: 10.1145/3341162.3349306.

[22] Y. Wei, S.-H. Hwang, and S.-M. Lee, « IoT-Aided Fingerprint Indoor Positioning Using Support Vector Classification », *in 2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 973–975. DOI: 10.1109/ICTC.2018.8539594.

[23] D. Wang, T. Wang, F. Zhao, and X. Zhang, « Improved Graph-Based Semi-Supervised Learning for Fingerprint-Based Indoor Localization », *in 2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6. DOI: 10.1109/GLOCOM.2018.8647621.

[24] B. Berruet, O. Baala, A. Caminada, and V. Guillet, « DelFin: A Deep Learning Based CSI Fingerprinting Indoor Localization in IoT Context », *in 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–8. DOI: 10.1109/IPIN.2018.8533777.

[25] P. Pascacio, S. Casteleyn, J. Torres-Sospedra, E. S. Lohan, and J. Nurmi, « Collaborative Indoor Positioning Systems: A Systematic Review », *Sensors*, vol. 21, *3*, 2021, ISSN: 1424-8220. DOI: 10.3390/s21031002. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/1002.

[26] E. Shakshuki, A. A. Elkhail, I. Nemer, M. Adam, and T. Sheltami, « Comparative Study on Range Free Localization Algorithms », *Procedia Computer Science*, vol. 151, pp. 501–510, 2019, The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.04.068. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919305307.

[27] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, « Indoor Positioning Algorithm Based on the Improved RSSI Distance Model », *Sensors*, vol. 18, *9*, 2018, ISSN: 1424-8220. DOI: 10.3390/s18092820. [Online]. Available: https://www.mdpi.com/1424-8220/18/9/2820.

[28] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, « Using RSSI value for distance estimation in wireless sensor networks based on ZigBee », *in 2008 15th International Conference on Systems, Signals and Image Processing*, 2008, pp. 303–306. DOI: 10.1109/IWSSIP.2008.4604427.

[29] W. Debus, *RF Path Loss and Transmission Distance Calculations*. Axonn, 2006.

[30] S. Wibowo, M. Klepal, and D. Pesch, « Time of Flight Ranging using Off-the-self IEEE802. 11 WiFi Tags », Jan. 2009.

[31] S. Lanzisera, D. T. Lin, and K. S. J. Pister, « RF Time of Flight Ranging for Wireless Sensor Network Localization », *in 2006 International Workshop on Intelligent Solutions in Embedded Systems*, 2006, pp. 1–12. DOI: 10.1109/WISES.2006.329127.

[32] L. Barclay, *Propagation of radiowaves*. Iet, 2003, vol. 2.

[33] P Series, « Effects of building materials and structures on radiowave propagation above about 100 MHz », *Recommendation ITU-R*, pp. 2040–1, 2015.

[34] R. A. Dalke, C. L. Holloway, P. McKenna, M. Johansson, and A. S. Ali, « Effects of reinforced concrete structures on RF communications », *IEEE Transactions on electromagnetic compatibility*, vol. 42, *4*, pp. 486–496, 2000.

[35] J. Dattorro, « CONVEX OPTIMIZATION † EUCLIDEAN DISTANCE GEOMETRY 2e », *in*. Dec. 2019. [Online]. Available: https://meboo.convexoptimization.com/access.html.

[36] M. Powell, « The volume internal to three intersecting hard spheres », *Molecular Physics*, vol. 7, *6*, pp. 591–592, 1964.

[37] K. D. Gibson and H. A. Scheraga, « Volume of the intersection of three spheres of unequal size: a simplified formula », *The Journal of Physical Chemistry*, vol. 91, *15*, pp. 4121–4122, 1987.

[38] C. C. et al., « Calculation of Weighted Geometric Dilution of Precision », *Journal of Applied Mathematics*, vol. 2013, *953048*, 2013.

[39] B. K. Horn, *Projective Geometry Considered Harmful*. [Online]. Available: http://people.csail.mit.edu/bkph/articles/Harmful.pdf[Accessed20072020].

[40] ——, *What is wrong with so-called 'linear' photogrammetric methods?* [Online]. Available: http://people.csail.mit.edu/bkph/articles/Orientation_2D_Illustration.pdf[Accessed20072020].

[41] S. Bouley, C. Vanwynsberghe, T. Le Magueresse, and J. Antoni, « On an array localization technique with Euclidean distance geometry », *in Euronoise 2018*, Hersonissos, Greece, May 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01820078.

[42] E. P. Simoncelli, *Least Squares Optimization*. [Online]. Available: http://www.cns.nyu.edu/~eero/math-tools19/Handouts/leastSquares.pdf[Accessed20072020].

[43] D. Young, C. Keller, D. Bliss, and K. Forsythe, « Ultra-wideband (UWB) transmitter location using time difference of arrival (TDOA) techniques », *in The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, 2003, 1225–1229 Vol.2. DOI: 10.1109/ACSSC.2003.1292184.

[44] R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato, « TDOA location system for IEEE 802.11b WLAN », *in IEEE Wireless Communications and Networking Conference, 2005*, vol. 4, 2005, 2338–2343 Vol. 4. DOI: 10.1109/WCNC.2005.1424880.

[45] J. Xu, M. Ma, and C. L. Law, « AOA Cooperative Position Localization », *in IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, 2008, pp. 1–5. DOI: 10.1109/GLOCOM.2008.ECP.720.

[46] B. Barodia, « Performance analysis of MUSIC algorithm for DOA estimation », *spectrum*, vol. 30, *20*, p. 10, 2017.

[47] N. Deligiannis, S. Louvros, and S. Kotsopoulos, « Optimizing Location Positioning Using Hybrid TOA-AOA Techniques in Mobile Cellular Networks », *in Proceedings of the 3rd International Conference on Mobile Multimedia Communications*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, ISBN: 9789630626705.

[48] G. Welch and G. Bishop, *An introduction to the Kalman Filter*, Jul. 2006. [Online]. Available: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf.

[49] A. Shareef and Y. Zhu, *Localization Using Extended Kalman Filters in Wireless Sensor Networks*, Apr. 2009. [Online]. Available: https://digitalcommons.library.umaine.edu/gradstudent_pub/5.

[50] G. Pipelidis, N. Tsiamitros, C. Gentner, D. B. Ahmed, and C. Prehofer, « A Novel Lightweight Particle Filter for Indoor Localization », *in 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pp. 1–8. DOI: 10.1109/IPIN.2019.8911744.

[51] J. Canales, « Navigating the History of GPS », *Nature Electronics*, vol. 1, 12 2018. DOI: https://doi.org/10.1038/s41928-018-0187-9.

[52] T. Yunck, C.-H. Liu, and R. Ware, « A history of GPS sounding », *Terrestrial, Atmospheric and Oceanic Sciences*, vol. -1, p. 1, Jan. 2001. DOI: 10.3319/TAO.2000.11.1.1(COSMIC).

[53] A. Mohan, *Calculating Position from Raw GPS Data*, Jul. 2017. [Online]. Available: https://www.telesens.co/2017/07/17/calculating-position-from-raw-gps-data/\#Step_1_Determining_the_Position_of_a_Satellite.

[54] S. R., G. A., and B. S., « Geometry of GPS dilution of precision: revisited », *GPS Solutions*, vol. 21, pp. 1747–1763, Oct. 2017. DOI: 10.1007/s10291-017-0649-y.

[55] D. Hanbay, N. Rahemi, M. R. Mosavi, and A. A. Abedi, « Accurate Solution of Navigation Equations in GPS Receivers for Very High Velocities Using Pseudorange Measurement », *Advances in Aerospace Engineering*, vol. 2014, Jun. 2014. DOI: 10.1155/2014/435891.

[56] N. Dvorecki, O. Bar-Shalom, L. Banin, and Y. Amizur, « A Machine Learning Approach for Wi-Fi RTT Ranging », *in Proceedings of the 2019 International Technical Meeting of The Institute of Navigation*, Jan. 2019, pp. 435–444. DOI: 10.33012/2019.16702.

[57] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, « A Survey of Enabling Technologies for Network Localization, Tracking, and Navigation », *IEEE Communications Surveys Tutorials*, vol. 20, *4*, pp. 3607–3644, 2018. DOI: 10.1109/COMST.2018.2855063.

[58] R. Yang, Z. Song, and X. Xi, « Self-assisted first-fix method for GPS receiver with autonomous short-term ephemeris prediction », *IET Radar, Sonar & Navigation*, vol. 13, *11*, pp. 1974–1980, 2019.

[59] L. Hyun, K. Kyu, and S. Seok, « Assisted SBAS Global Navigation Satellite System Operation Method for Reducing SBAS Time to First Fix », *Journal of Advanced Navigation Technology*, vol. 24, *2*, pp. 92–100, Apr. 2020.

[60] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, « Survey of Cellular Mobile Radio Localization Methods: From 1G to 5G », *IEEE Communications Surveys Tutorials*, vol. 20, *2*, pp. 1124–1148, 2018. DOI: 10.1109/COMST.2017.2785181.

[61] M. Posluk, J. Ahlander, D. Shrestha, S. M. Razavi, G. Lindmark, and F. Gunnarsson, *5G Deployment Strategies for High Positioning Accuracy in Indoor Environments*, 2021. arXiv: 2105.09584 [cs.NI].

[62] L. Logan, C. Davids, and C. Davids, « Determining the Indoor Location of an Emergency Caller in a Multi-story Building », *in 2020 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, 2020, pp. 1–6. DOI: 10.1109/CQR47547.2020.9101391.

[63] S. Fischer, « Observed Time Difference of Arrival (OTDOA) Positioning in 3GPP LTE », Qualcomm, Tech. Rep. [Online]. Available: https://www.qualcomm.com/media/documents/files/otdoa-positioning-in-3gpp-lte.pdf.

[64] F. Wang and X. Liu, « Empirical Analysis of TOA and AOA in Hybrid Location Positioning Techniques », *IOP Conference Series: Materials Science and Engineering*, vol. 466, p. 012 089, Dec. 2018. DOI: 10.1088/1757-899x/466/1/012089. [Online]. Available: https://doi.org/10.1088/1757-899x/466/1/012089.

[65] Z. Zhang, S. Kang, and X. Zhang, « A Bayesian Probabilistic AOA Localization Algorithm », *in 2020 IEEE 8th International Conference on Information, Communication and Networks (ICICN)*, 2020, pp. 96–100. DOI: 10.1109/ICICN51133.2020.9205101.

[66] E. Y. Menta, N. Malm, R. Jäntti, K. Ruttik, M. Costa, and K. Leppänen, « On the Performance of AoA–Based Localization in 5G Ultra–Dense Networks », *IEEE Access*, vol. 7, pp. 33 870–33 880, 2019. DOI: 10.1109/ACCESS.2019.2903633.

[67] L. Wang and M. Zawodniok, « RSSI-based localization in cellular networks », *in 37th Annual IEEE Conference on Local Computer Networks - Workshops*, 2012, pp. 820–826. DOI: 10.1109/LCNW.2012.6424069.

[68] I. Perpetual, A. Ekanem, and N. Aloziem, « Comparison of Path Loss Prediction Performance of Egli Model and Lee Model For Cellular Network Signal Along A Dual Carriage Way In Uyo », Oct. 2019. DOI: 10.13140/RG.2.2.10711.62882.

[69] V. Kumar, « RSS-RSQ Based Base Station and Mobile Station Localization in 3GPP UMTS Cellular Network », *in 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2019, pp. 1–7. DOI: 10.1109/ICECCT.2019.8868991.

[70] J. Silaa, H. Jazri, and H. Muyingi, « A study on the use of mobile computing technologies for improving the mobility of Windhoek residents », *African Journal of Science, Technology, Innovation and Development*, vol. 0, *0*, pp. 1–15, 2021. DOI: 10.1080/20421338.2020.1838083.

[71] T. Mangla, E. Showalter, V. Adarsh, K. Jones, M. Vigil-Hayes, E. Belding, and E. Zegura, *A Tale of Three Datasets: Towards Characterizing Mobile Broadband Access in the United States*, 2021. arXiv: 2102.07288 [cs.NI].

[72] J. Roth, M. Tummala, J. Mceachen, and J. W. Scrofani, « Location Privacy in LTE: A Case Study on Exploiting the Cellular Signaling Plane's Timing Advance », *in HICSS*, 2017.

[73] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, « Trajectory Recovery From Ash: User Privacy Is NOT Preserved in Aggregated Mobility Data », *in Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17, Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, 1241–1250, ISBN: 9781450349130. DOI: 10.1145/3038912.3052620. [Online]. Available: https://doi.org/10.1145/3038912.3052620.

[74] R. Langley, « Propagation of the GPS Signals », *in*. Apr. 2008, pp. 103–140, ISBN: 978-3-642-72013-0. DOI: 10.1007/BFb0117680.

[75] C. Ma, G. Jee, G. Macgougan, G. Lachapelle, S. Bloebaum, G. Cox, L. Garin, and J. Shewfelt, « GPS Signal Degradation Modeling », 2001.

[76] M. B. Kjærgaard, H. Blunck, T. Godsk, T. Toftkjær, D. L. Christensen, and K. Grønbæk, « Indoor Positioning Using GPS Revisited », *in Pervasive Computing*, P. Floréen, A. Krüger, and M. Spasojevic, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 38–56, ISBN: 978-3-642-12654-3.

[77] A. Davidson and C. Hill, « Measurement of building penetration into medium buildings at 900 and 1500 MHz », *IEEE Transactions on Vehicular Technology*, vol. 46, *1*, pp. 161–168, 1997. DOI: 10.1109/25.554748.

[78] A. Turkmani, J. Parsons, and D. Lewis, « Measurement of building penetration loss on radio signals at 441, 900 and 1400 MHz », *Journal of the Institution of Electronic and Radio Engineers*, vol. 58, S169–S174, 1988. DOI: DOI:10.1049/jiere.1988.0064.

[79] E. H. Walker, « Penetration of radio signals into buildings in the cellular radio environment », *The Bell System Technical Journal*, vol. 62, *9*, pp. 2719–2734, 1983. DOI: 10.1002/j.1538-7305.1983.tb03201.x.

[80] L. Lee, M. Jones, G. S. Ridenour, S. J. Bennett, A. C. Majors, B. L. Melito, and M. J. Wilson, « Comparison of Accuracy and Precision of GPS-Enabled Mobile Devices », *in 2016 IEEE International Conference on Computer and Information Technology (CIT)*, 2016, pp. 73–82. DOI: 10.1109/CIT.2016.94.

[81] P. Teunissen, « A proof of Nielsen's conjecture on the GPS dilution of precision », *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, *2*, pp. 693–695, 1998. DOI: 10.1109/7.670364.

[82] R. Langley, « Dilution of Precision », 1999.

[83] X. Liu, S. K. Nath, and R. Govindan, « Gnome: A Practical Approach to NLOS Mitigation for GPS Positioning in Smartphones », Jan. 2018, pp. 163–177. DOI: 10.1145/3210240.3210343.

[84] L. Ma, C. Zhang, Y. Wang, G. Peng, C. Chen, J. Zhao, and J. Wang, « Estimating Urban Road GPS Environment Friendliness with Bus Trajectories: A City-Scale Approach », *Sensors*, vol. 20, *6*, p. 1580, Mar. 2020, ISSN: 1424-8220. DOI: 10.3390/s20061580. [Online]. Available: http://dx.doi.org/10.3390/s20061580.

[85] S. Mun, J. An, and J. Lee, « Robust Positioning Algorithm for a Yard Transporter Using GPS Signals with a Modified FDI and HDOP », *Int. J. Precis. Eng. Manufacturing*, pp. 1107–1113, 19 2018. [Online]. Available: https://doi.org/10.1007/s12541-018-0131-y.

[86] D. Kim, J. Youn, T. Kim, and G. Kim, « DEVELOPMENT OF GPS SATELLITE VISIBILITY SIMULATION METHOD UNDER URBAN CANYON ENVIRONMENT », *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, pp. 431–435, Jun. 2019. DOI: 10.5194/isprs-archives-XLII-2-W13-431-2019.

[87] P. Ivanov, M. Raitoharju, and R. Piché, « Kalman-Type Filters and Smoothers for Pedestrian Dead Reckoning », *in 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 206–212. DOI: 10.1109/IPIN.2018.8533753.

[88] H. Rizk, A. Shokry, and M. Youssef, « Effectiveness of Data Augmentation in Cellular-based Localization Using Deep Learning », *in 2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–6. DOI: 10.1109/WCNC.2019.8886005.

[89] S. Qiu, Z. Wang, H. Zhao, K. Qin, Z. Li, and H. Hu, « Inertial/magnetic sensors based pedestrian dead reckoning by means of multi-sensor fusion », *Information Fusion*, vol. 39, pp. 108–119, 2018, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2017.04.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253517302701.

[90] D. Wu, L. Xia, J. Geng, and Q. Peng, « Robust Adaptive Extended Kalman Filtering for Smart Phone-based Pedestrian Dead Reckoning Systems », *in 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 2018, pp. 1–8. DOI: 10.1109/UPINLBS.2018.8559880.

[91] H. Aly, A. Basalamah, and M. Youssef, « Accurate and Energy-Efficient GPS-Less Outdoor Localization », vol. 3, *2*, 2017, ISSN: 2374-0353. DOI: 10.1145/3085575. [Online]. Available: https://doi.org/10.1145/3085575.

[92] H. Leppäkoski, J. Collin, and J. Takala, « Pedestrian Navigation Based on Inertial Sensors, Indoor Map, and WLAN Signals », *Journal of Signal Processing Systems*, vol. 71, Jun. 2012. DOI: 10.1007/s11265-012-0711-5.

[93] C. Zhao and B. Wang, « A UWB/Bluetooth Fusion Algorithm for Indoor Localization », *in 2019 Chinese Control Conference (CCC)*, 2019, pp. 4142–4146. DOI: 10.23919/ChiCC.2019.8865457.

[94] K. Zhang, C. Shen, and Q. Zhou, « A combined GPS UWB and MARG locationing algorithm for indoor and outdoor mixed scenario », *Cluster Computing*, pp. 5965–5974, 22 2019. [Online]. Available: https://doi.org/10.1007/s10586-018-1735-9.

[95] K. A. Kumar and O. Dhadge, « A Novel Infrared (IR) Based Sensor System for Human Presence Detection in Targeted Locations », *International Journal of Computer Network and Information Security*, vol. 10, pp. 34–40, 2018.

[96] E. M. Gorostiza, J. L. Lázaro Galilea, F. J. Meca Meca, D. Salido Monzú, F. Espinosa Zapata, and L. Pallarés Puerto, « Infrared Sensor System for Mobile-Robot Positioning in Intelligent Spaces », *Sensors*, vol. 11, *5*, pp. 5416–5438, 2011, ISSN: 1424-8220. DOI: 10.3390/s110505416. [Online]. Available: https://www.mdpi.com/1424-8220/11/5/5416.

[97] D. Arubla and S. Ljubic, « Indoor Localization Based on Infrared Angle of Arrival Sensor Network », *Sensors*, vol. 20, *21*, 2020. DOI: 10.3390/s20216278.

[98] M. F. Keskin, A. D. Sezer, and S. Gezici, « Localization via Visible Light Systems », *Proceedings of the IEEE*, vol. 106, *6*, pp. 1063–1088, 2018. DOI: 10.1109/JPROC.2018.2823500.

[99] B. Zhou, A. Liu, and V. Lau, « Joint User Location and Orientation Estimation for Visible Light Communication Systems With Unknown Power Emission », *IEEE Transactions on Wireless Communications*, vol. 18, *11*, pp. 5181–5195, 2019. DOI: 10.1109/TWC.2019.2934107.

[100] J. Qi and G.-P. Liu, « A Robust High-Accuracy Ultrasound Indoor Positioning System Based on a Wireless Sensor Network », *Sensors*, vol. 17, *11*, 2017, ISSN: 1424-8220. DOI: 10.3390/s17112554. [Online]. Available: https://www.mdpi.com/1424-8220/17/11/2554.

[101] J. Park, H. Kim, J. Yoon, H. Kim, C. Park, and D. Hong, « Development of an Ultrasound Technology-Based Indoor-Location Monitoring Service System for Worker Safety in Shipbuilding and Offshore Industry », *Processes*, vol. 9, *2*, 2021, ISSN: 2227-9717. DOI: 10.3390/pr9020304. [Online]. Available: https://www.mdpi.com/2227-9717/9/2/304.

[102] Y. Yaslan and B. Gunsel, « A context-aware mobile application framework using audio watermarking », *Multimedia Systems*, vol. 26, p. 323, 3 2020.

[103] j. moutinho, d. freitas, and r. e. araújo, « steganography for indoor location », *journal of the audio engineering society*, Jun. 2019.

[104] J. Rabadan, V. Guerra, R. Rodríguez, J. Rufo, M. Luna-Rivera, and R. Perez-Jimenez, « Hybrid Visible Light and Ultrasound-Based Sensor for Distance Estimation », *Sensors*, vol. 17, *2*, 2017, ISSN: 1424-8220. DOI: 10.3390/s17020330. [Online]. Available: https://www.mdpi.com/1424-8220/17/2/330.

[105] W. H. FOY, « Position-Location Solutions by Taylor-Series Estimation », *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-12, *2*, pp. 187–194, 1976. DOI: 10.1109/TAES.1976.308294.

[106] K. Yang, J. An, X. Bu, and G. Sun, « Constrained Total Least-Squares Location Algorithm Using Time-Difference-of-Arrival Measurements », *IEEE Transactions on Vehicular Technology*, vol. 59, *3*, pp. 1558–1562, 2010. DOI: 10.1109/TVT.2009.2037509.

[107] K. Yang, G. Wang, and Z.-Q. Luo, « Efficient Convex Relaxation Methods for Robust Target Localization by a Sensor Network Using Time Differences of Arrivals », *IEEE Transactions on Signal Processing*, vol. 57, *7*, pp. 2775–2784, 2009. DOI: 10.1109/TSP.2009.2016891.

[108] P. Wu, S. Su, Z. Zuo, X. Guo, B. Sun, and X. Wen, « Time Difference of Arrival (TDoA) Localization Combining Weighted Least Squares and Firefly Algorithm », *Sensors*, vol. 19, *11*, 2019, ISSN: 1424-8220. DOI: 10.3390/s19112554. [Online]. Available: https://www.mdpi.com/1424-8220/19/11/2554.

[109] Z. Deng, H. Wang, X. Zheng, X. Fu, L. Yin, S. Tang, and F. Yang, « A Closed-Form Localization Algorithm and GDOP Analysis for Multiple TDOAs and Single TOA Based Hybrid Positioning », *Applied Sciences*, vol. 9, *22*, 2019, ISSN: 2076-3417. DOI: 10.3390/app9224935. [Online]. Available: https://www.mdpi.com/2076-3417/9/22/4935.

[110] B. Rudić, M. A. Klaffenböck, M. Pichler-Scheder, D. Efrosinin, and C. Kastl, « Geometry-Aided BLE-Based Smartphone Positioning for Indoor Location-Based Services », *in 2020 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2020, pp. 1–4. DOI: 10.1109/ICMIM48759.2020.9299009.

[111] K. Urano, K. Hiroi, T. Yonezawa, and N. Kawaguchi, « An End-to-End BLE Indoor Location Estimation Method Using LSTM », *in 2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, 2019, pp. 1–7. DOI: 10.23919/ICMU48249.2019.9006638.

[112] S. Monfared, T.-H. Nguyen, L. Petrillo, P. De Doncker, and F. Horlin, « Experimental Demonstration of BLE Transmitter Positioning Based on AOA Estimation », *in 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 856–859. DOI: 10.1109/PIMRC.2018.8580796.

[113] *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*, Piscataway, NJ: IEEE, 1990.

[114] B. Zhou, W. Ma, Q. Li, N. El-Sheimy, Q. Mao, Y. Li, F. Gu, L. Huang, and J. Zhu, « Crowdsourcing-based indoor mapping using smartphones: A survey », *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 177, pp. 131–146, 2021, ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2021.05.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271621001313.

[115] Y. Polk, M. Linser, M.Thomson, and B. Aboba, « Dynamic Host Configuration Protocol Options for Coordinate-Based Location Configuration Information », RFC Editor, RFC 6225, Jul. 2011, pp. 1–36.

[116] M. Kjaergaard and et al., « Indoor Positioning Using GPS Revisited », *in International Conference on Pervasive Computing*. 2010, pp. 38–56.

[117] L. Liberti, *Distance Geometry and Data Science*, 2019. arXiv: 1909.08544 [cs.LG].

[118] T. Eren and et. al., « Rigidity, computation, and randomization in network localization », *IEEE INFOCOM*, vol. 4, pp. 2673–2684, 2004.

[119] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, « Convex position estimation in wireless sensor networks », *in Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, 2001, pp. 1655–1663.

[120] D. Jacobs, « Multidimensional Scaling: More complete proof and some insights not mentioned in class », 2016. [Online]. Available: http://www.cs.umd.edu/~djacobs/CMSC828/MDSexplain.pdf[Accessed20032020].

[121] J. de Leeuw, « Applications of Convex Analysis to Multidimensional Scaling », 2005. [Online]. Available: https://escholarship.org/uc/item/7wg0k7xq[Accessed20032020].

[122] J. Tsang and R. Pereira, « Taking All Positive Eigenvectors Is Suboptimal in Classical Multidimensional Scaling », *ArXiv*, vol. abs/1402.2703, 2016.

[123] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, « Euclidean Distance Matrices: Essential theory, algorithms, and applications », *IEEE Signal Processing Magazine*, vol. 32, *6*, pp. 12–30, Nov. 2015.

[124] E. Candes and Y. Plan, « Matrix Completion With Noise », *Proceedings of the IEEE*, vol. 98, *6*, 925–936, Jun. 2010, ISSN: 1558-2256. DOI: 10.1109/jproc.2009.2035722. [Online]. Available: http://dx.doi.org/10.1109/JPROC.2009.2035722.

[125] X. Guo, L. Chu, and X. Sun, « Accurate Localization of Multiple Sources Using Semidefinite Programming Based on Incomplete Range Matrix », *IEEE Sensors Journal*, vol. 16, *13*, pp. 5319–5324, Jul. 2016.

[126] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, « Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms », *in Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS '16, Xi'an, China: ACM, 2016, pp. 413–424, ISBN: 978-1-4503-4233-9. DOI: 10.1145/2897845.2897883. [Online]. Available: http://doi.acm.org/10.1145/2897845.2897883.

[127] M. Cunche, « I know your MAC Address: Targeted tracking of individual using Wi-Fi », *Journal of Computer Virology and Hacking Techniques*, vol. 10, pp. 219–227, Nov. 2013. DOI: 10.1007/s11416-013-0196-1.

[128] Y. Chapre, A. Ignjatović, A. Seneviratne, and S. Jha, « CSI-MIMO: An efficient Wi-Fi fingerprinting using Channel State Information with MIMO », *Pervasive and Mobile Computing*, vol. 23, Jul. 2015. DOI: 10.1016/j.pmcj.2015.07.002.

[129] H. Ahmadi and R. Bouallegue, « Exploiting machine learning strategies and RSSI for localization in wireless sensor networks: A survey », *in 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jun. 2017, pp. 1150–1154. DOI: 10.1109/IWCMC.2017.7986447.

[130] T. Mukherjee, M. Duckett, P. Kumar, J. Devin Paquet, D. Rodriguez, M. Haulcomb, K. George, and E. Pasiliao, « RSSI-Based Supervised Learning for Uncooperative Direction-Finding », *in*. Dec. 2017, pp. 216–227, ISBN: 978-3-319-71272-7. DOI: 10.1007/978-3-319-71273-4_18.

[131] K. D. Yüksel and B. U. Töreyin, « A deep learning and RSSI based approach for indoor positioning », *in 2018 26th Signal Processing and Communications Applications Conference (SIU)*, May 2018, pp. 1–4. DOI: 10.1109/SIU.2018.8404614.

[132] W. Lowder, « Real-Time RF-DNA Fingerprinting of ZigBee Devices Using a Software-Defined Radio with FPGA Processing », M.S. thesis, Defense Technical Information Center, 2015.

[133] R. E. Barton, J. Henry, G. Ian Mc Garry, S. M. Orr, and S. Pandey, *System and method to facilitate troubleshooting and predicting application performance in wireless networks*, US Patent 20180242178, Aug. 2018.

[134] V. Primault, A. Boutet, S. B. Mokhtar, and L. Brunie, « The Long Road to Computational Location Privacy: A Survey », *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018, ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2873950.

[135] T. team, *Recurrent Neural Networks*, 2019. [Online]. Available: https://www.tensorflow.org/tutorials/sequences/recurrent.[Accessed20062019].

[136] R. Hu, L. Ju, and P. Chen, « A security transmission system for Beidou short message based on SM9 », *Journal of Physics: Conference Series*, vol. 1345, p. 022 014, Nov. 2019. DOI: 10.1088/1742-6596/1345/2/022014. [Online]. Available: https://doi.org/10.1088\%2F1742-6596\%2F1345\%2F2\%2F022014.

[137] K. Tugsad Seferoglu and A. Serdar Turk, « Review of Spoofing and Jamming Attack on the Global Navigation Systems Band and Countermeasure », *in 2019 9th International Conference on Recent Advances in Space Technologies (RAST)*, 2019, pp. 513–520. DOI: 10.1109/RAST.2019.8767871.

[138] R. Ferreira, J. Gaspar, N. Souto, and P. Sebastião, « Effective GPS Jamming Techniques for UAVs Using Low-Cost SDR Platforms », *Wireless Personal Communications*, vol. 115, pp. 27–32, Dec. 2020. DOI: 10.1007/s11277-020-07212-6.

[139] J. Su, J. He, P. Cheng, and J. Chen, « A Stealthy GPS Spoofing Strategy for Manipulating the Trajectory of an Unmanned Aerial Vehicle », *IFAC-PapersOnLine*, vol. 49, pp. 291 –296, 22 2016.

[140] ACM, Ed., *A Practical GPS Location Spoofing Attack in Road Navigation Scenario*, Feb. 2017, pp. 85–90.

[141] E. Horton and P. Ranganathan, « Development of a GPS spoofing apparatus to attack a DJI Matrice 100 Quadcopter », *The Journal of Global Positioning Systems*, vol. 16, pp. 1446–1464, 1 2018.

[142] S. Hewiston and J. Wang, « GNSS receiver autonomous integrity monitoring (RAIM) performance analysis », *GPS Solutions*, vol. 10, pp. 155–170, 2006.

[143] M. Foruhandeh, A. Z. Mohammed, G. Kildow, and R. Gerdes, « Spotr: GPS Spoofing Detection via Device Fingerprinting », *in Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec'20)*, Linz (Virtual Event), Austria, 2020. arXiv: 2005.087875 [eess].

[144] G. Oligeri, S. Sciancalepore, O. A. Ibrahim, and R. Di Pietro, « Drive Me Not: GPS Spoofing Detection via Cellular Network: (Architectures, Models, and Experiments) », *in Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '19, Miami, Florida: Association for Computing Machinery, 2019, 12–22, ISBN: 9781450367264. DOI: 10.1145/3317549.3319719.

[145] J.-H. Lee, K.-C. Kwon, D.-S. An, and D.-S. Shim, « GPS spoofing detection using accelerometers and performance analysis with probability of detection », *International Journal of Control, Automation and Systems*, vol. 13, pp. 951 –959, 4 2015.

[146] K. Hu and Y. Huang, « A Composite Detection Method for Direct GPS Deception Attack », *IOP Conference Series: Materials Science and Engineering*, vol. 790, p. 012 028, Apr. 2020. DOI: 10.1088/1757-899x/790/1/012028.

[147] E. Shafiee, M. R. Mosavi, and M. Moazedi, « Detection of Spoofing Attack using Machine Learning based on Multi-Layer Neural Network in Single-Frequency GPS Receivers », *Journal of Navigation*, vol. 71, *1*, 169–188, 2018. DOI: 10.1017/S0373463317000558.

[148] K. Wesson, M. Rothlisberger, and T. Humphreys, « Practical Cryptographic Civil GPS Signal Authentication », *NAVIGATION*, vol. 59, *3*, pp. 177–193, 2012. DOI: https://doi.org/10.1002/navi.14.

[149] A. D. Molina-Markham, « Probabilistic models for assured position, navigation, and timing », *in Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything*, M. C. Dudzik and J. C. Ricklin, Eds., International Society for Optics and Photonics, vol. 10643, SPIE, 2018, pp. 148 –167. DOI: 10.1117/12.2301254.

[150] B. K. Horn, « Doubling the Accuracy of Indoor Positioning: Frequency Diversity », *Sensors*, vol. 1789, p. 1489, Mar. 2020. DOI: 10.3390/s20051489.

[151] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, *Reference-PE- and-Measurements-DB-for-WiFi-Time-Based-Scalable-Location*, Accessed July 2020. [Online]. Available: https://github.com/intel/Reference-PE-and-Measurements-DB-for-WiFi-Time-based-Scalable-Location.

[152] W. Simoes, G. Machado, A. Sales, M. de Lucena, N. Jazdi, and V. de Lucena, « A Review of Technologies and Techniques for Indoor Navigation Systems for the Visually Impaired », *Sensors*, vol. 20, *14*, p. 3935, 2015. DOI: doi:10.3390/s20143935.

[153] L. Lamport, *LaTeX, A Document Preparation System: User's Guide And Reference Manual*. New York: Pearson Professional Education, 1994.

[154] M. Goossens, F. Mittelbach, and A. Samarin, *The LaTeX Companion*. Addison-Wesley, 1994, ISBN: 0201541998.

[155] A. Bensky, *Wireless positioning technologies and applications*. Boston: Artech House, 2008.

[156] W. Murphy and W. Hereman, « Determination of a position in three dimensions using trilateration and approximate distances », *Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, MCS-95*, vol. 7, p. 19, 1995.

[157] P. Ängskog, M. Bäckström, and B. Vallhagen, « Measurement of Radio Signal Propagation through Window Panes and Energy Saving Windows », *in Proceedings of Electromagnetic Compatibility (EMC), 2015 IEEE International Symposium*, 2015, pp. 74–79. DOI: http://dx.doi.org/10.1109/ISEMC.2015.7256135.

[158] P. Ali-Rantala, L. Ukkonen, L. Sydanheimo, M. Keskilammi, and M. Kivikoski, « Different kinds of walls and their effect on the attenuation of radiowaves indoors », *in IEEE Antennas and Propagation Society International Symposium. Digest. Held in conjunction with: USNC/CNC/URSI North American Radio Sci. Meeting (Cat. No.03CH37450)*, vol. 3, 2003, 1020–1023 vol.3. DOI: 10.1109/APS.2003.1220085.

[159] T.-H. Yi, H.-N. Li, and M. Gu, « Effect of different construction materials on propagation of GPS monitoring signals », *Measurement*, vol. 45, *5*, pp. 1126–1139, 2012, ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2012.01.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224112000425.

[160] W. Stone, *Electromagnetic Signal Attenuation in Construction Materials*, en, Oct. 1997. DOI: https://doi.org/10.6028/NIST.IR.6055.

[161] S. Yousuf and M. B. Kadri, « Information Fusion of GPS, INS and Odometer Sensors for Improving Localization Accuracy of Mobile Robots in Indoor and Outdoor Applications », *Robotica*, vol. 39, *2*, 250–276, 2021. DOI: 10.1017/S0263574720000351.

[162] B. T. Fang, « Trilateration and Extension to Global Positioning Navigation System », *Journal of Guidance, Control and Dynamics*, vol. 9, 715=717, 1986.

[163] Z. Li, D. Zhou, and Y. Huang, « Design of Outdoor Following Vehicle System Based on GPS-INS Fusion Navigation Algorithm », *in 2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, 2018, pp. 1285–1289. DOI: 10.1109/IMCEC.2018.8469395.

[164] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, Piscataway, NJ: IEEE, 2016.

[165] I. Borg and P. Groenen, *Modern Multidimensional Scaling Theory and Applications*. New York: Springer, 2005, ISBN: 038728981X. DOI: 10.1007/0-387-28981-X.

[166] Y. Yu and et al., « A Robust Dead Reckoning Algorithm Based on Wi-Fi FTM and Multiple Sensors », *IEEE Remote Sensing*, vol. 11, *5*, p. 504, 2019.

[167] L. Pan, C. Cai, R. Santerre, and X. Zhang, « Performance evaluation of single-frequency point positioning with GPS, GLONASS, BeiDou and Galileo », *Survey Review*, vol. 49, *354*, pp. 197–205, 2017. DOI: 10.1080/00396265.2016.1151628. eprint: https://doi.org/10.1080/00396265.2016.1151628. [Online]. Available: https://doi.org/10.1080/00396265.2016.1151628.

[168] X. Liu, S. K. Nath, and R. Govindan, « Gnome: A Practical Approach to NLOS Mitigation for GPS Positioning in Smartphones », Jan. 2018, pp. 163–177. DOI: 10.1145/3210240.3210343.

[169] D. Niculescu and B. Nath, « Ad hoc positioning system (APS) », *in IEEE Global Telecommunications Conference*, 2001, pp. 2926–2931.

[170] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, « Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications », *IEEE Communications Surveys Tutorials*, vol. 19, *2*, pp. 1327–1346, 2017.

[171] R. Liu, C. Yuen, T. Do, and U. Tan, « Fusing Similarity-Based Sequence and Dead Reckoning for Indoor Positioning Without Training », *IEEE Sensors Journal*, vol. 17, *13*, pp. 4197–4207, 2017.

[172] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, Piscataway, NJ: IEEE, 2016.

[173] H. Blunck, M. Kjærgaard, T. Godsk, T. Toftkjær, D. Lund, and K. Grønbæk, « Empirical analysis and characterization of indoor gps signal fading and multipath conditions », *International Technical Meeting of the Satellite Division of the Institute of Navigation (GNSS)*, Jan. 2009.

[174] F. Zafari, A. Gkelias, and K. K. Leung, « A Survey of Indoor Localization Systems and Technologies », vol. abs/1709.01015, 2017. [Online]. Available: `http://arxiv.org/abs/1709.01015`.

[175] H. Koyuncu and S.-H. Yang, « A Survey of Indoor Positioning and Object Locating Systems », *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 10, Jan. 2010.

[176] W Sakpere, M. Adeyeye-Oshing, and N Mlitwa, « A state-of-the-art survey of indoor positioning and navigation systems and technologies », *South African Computer Journal*, vol. 29, *3*, pp. 145–197, Dec. 2017, ISSN: 1015-7999. DOI: `10.1109/TSMCC.2007.905750`. [Online]. Available: `https://doi.org/10.18489/sacj.v29i3.452`.

[177] R. J. Fontana, « Recent System Applications of Short-Pulse Ultra-Wideband (UWB) Technology », *Microwave Theory and Techniques, IEEE Transactions on*, vol. 52, pp. 2087 –2104, Oct. 2004. DOI: `10.1109/TMTT.2004.834186`.

[178] M. Bedford and G. A. Kennedy, « Evaluation of ZigBee (IEEE 802.15.4) Time-of-Flight-Based Distance Measurement for Application in Emergency Underground Navigation », *Antennas and Propagation, IEEE Transactions on*, vol. 60, pp. 2502–2510, May 2012. DOI: `10.1109/TAP.2012.2189731`.

[179] S König, M. Schmidt, and C. Hoene, « Precise time of flight measurements in IEEE 802.11 networks by cross-correlating the sampled signal with a continuous Barker code », Dec. 2010, pp. 642 –649. DOI: `10.1109/MASS.2010.5663785`.

[180] T. Karalar and J. Rabaey, « An RF ToF Based Ranging Implementation for Sensor Networks », vol. 7, Jul. 2006, pp. 3347 –3352. DOI: `10.1109/ICC.2006.255233`.

[181] I. Casacuberta and A. Ramirez, « Time-of-flight positioning using the existing wireless local area network infrastructure », Nov. 2012, pp. 1–8, ISBN: 978-1-4673-1955-3. DOI: `10.1109/IPIN.2012.6418938`.

[182] W. Xinrui, T.-F. Lu, and C. Lei, « Synchronization and Time Resolution Improvement for 802.11 WLAN OWPT Measurement », *Lecture Notes in Engineering and Computer Science*, vol. 1, Mar. 2009.

[183] J. Yim, « Comparison between RSSI-based and TOF-based Indoor Positioning Methods », *International Journal of Multimedia and Ubiquitous Engineering*, vol. 7, Jan. 2012.

[184] L. Schauer, F. Dorfmeister, and M. Maier, « Potentials and limitations of WIFI-positioning using Time-of-Flight », Oct. 2013, pp. 1–9, ISBN: 978-1-4799-4043-1. DOI: 10.1109/IPIN.2013.6817861.

[185] A. I. Kayssi, K. A. Sakallah, and T. N. Mudge, « The impact of signal transition time on path delay computation », *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, *5*, pp. 302–309, May 1993, ISSN: 1057-7130. DOI: 10.1109/82.227370.

[186] K. Merchant, S. Revay, G. Stanchev, and B. Nousain, « Deep Learning for RF Device Fingerprinting in Cognitive Communication Networks », *IEEE Journal of selected topics in signal processing*, 2018.

[187] F. Cruz, « Near Real-Time RF-DNA Fingerprinting for ZigBee Devices Using Software Defined Radios », M.S. thesis, Air Force Institute of Technology, 2019.

[188] M. Kose, S. Tascioglu, and Z. Telatar, « RF Fingerprinting of IoT Devices Based on Transient Energy Spectrum », *IEEE Access*, 2019.

[189] L. d. A. Faria, C. A. d. M. Silvestre, M. A. F. Correia, and N. A. Roso, « GPS Jamming Signals Propagation in Free-Space, Urban and Suburban Environments », en, *Journal of Aerospace Technology and Management*, vol. 10, Jan. 2018, ISSN: 2175-9146. DOI: 10.5028/jatm.v10.870.

[190] L. d. A. Faria, C. A. d. M. Silvestre, and M. A. F. Correia, « GPS-Dependent Systems: Vulnerabilities to Electromagnetic Attacks », en, *Journal of Aerospace Technology and Management*, vol. 8, pp. 423 –430, Dec. 2016, ISSN: 2175-9146. DOI: 10.5028/jatm.v8i4.632.

[191] Yan Ge, Zhi Zheng, Bo Yan, Jiao Yang, Yuxuan Yang, and Huipeng Meng, « An RSSI-based localization method with outlier suppress for wireless sensor networks », *in 2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 2235–2239.

[192] M. L. Psiaki and T. E. Humphreys, « GNSS Spoofing and Detection », *Proceedings of the IEEE*, vol. 104, *6*, pp. 1258–1270, 2016.

[193] H. Zhao, L. Zhang, S. Qiu, Z. Wang, N. Yang, and J. Xu, « Pedestrian Dead Reckoning Using Pocket-Worn Smartphone », *IEEE Access*, vol. 7, pp. 91 063–91 073, 2019. DOI: 10.1109/ACCESS.2019.2927053.

[194] H. Aly, A. Basalamah, and M. Youssef, « Accurate and Energy-Efficient GPS-Less Outdoor Localization », *ACM Trans. Spatial Algorithms Syst.*, vol. 3, *2*, Jul. 2017, ISSN: 2374-0353. DOI: 10.1145/3085575.

[195] A. Shokry, M. Torki, and M. Youssef, « DeepLoc: A Ubiquitous Accurate and Low-Overhead Outdoor Cellular Localization System », *in Proceedings of the 26th ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL'18)*, New York, NY, USA: Association for Computing Machinery, 2018, ISBN: 9781450358897. DOI: 10.1145/3274895.3274909.

[196] H. Wang, Z. Wang, G. Shen, F. Li, S. Han, and F. Zhao, « WheelLoc: Enabling continuous location service on mobile phone for outdoor scenarios », *in 2013 Proceedings IEEE INFOCOM*, 2013, pp. 2733–2741. DOI: 10.1109/INFCOM.2013.6567082.

[197] K. Elawaad, M. Ezzeldin, and M. Torki, « DeepCReg: Improving Cellular-based Outdoor Localization using CNN-based Regressors », *in 2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6. DOI: 10.1109/WCNC45663.2020.9120714.

[198] W. Yan, L. Wang, Y. Jin, and G. Shi, « High accuracy Navigation System using GPS and INS system integration strategy », *in 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2016, pp. 365–369. DOI: 10.1109/CYBER.2016.7574851.

[199] B. Buchli and J. Sutton Felixand Beutel, « GPS-Equipped Wireless Sensor Network Node for High-Accuracy Positioning Applications », *in Wireless Sensor Networks*, G. P. Picco and W. Heinzelman, Eds., Springer Berlin Heidelberg, 2012, pp. 179–195.

[200] S. Shafeeque, G. S. N. Meedin, and H. U. W. Ratnayake, « Locating the Position of a Cell Phone User Using GSM Signals », *in Artificial Intelligence*, J. Hemanth, T. Silva, and A. Karunananda, Eds., Singapore: Springer Singapore, 2019, pp. 49–63.

[201] R. T. Reza and V. M. Srivastava, « Effect of GSM Frequency Band on Received Signal Strength and Distance Estimation from Cell Tower », *in 2017 10th International Conference on Developments in eSystems Engineering (DeSE)*, 2017, pp. 151–154. DOI: 10.1109/DeSE.2017.10.

[202] W. Tan, T. Anglea, and Y. Wang, « Analysis of Dead Reckoning Accuracy in Swarm Robotics System », *in 2018 13th World Congress on Intelligent Control and Automation (WCICA)*, 2018, pp. 860–864. DOI: 10.1109/WCICA.2018.8630531.

[203] S. Razzaq, M. Abd, and E. Muhssan, « Object Tracking based on GPS Technology », *International Journal of Advanced Research in Science, Engineering and Technology*, vol. 4, Mar. 2017.

[204] Y. Huang, Z. Zhang, S. Du, Y. Li, and Y. Zhang, « A High-Accuracy GPS-Aided Coarse Alignment Method for MEMS-Based SINS », *IEEE Transactions on Instrumentation and Measurement*, vol. 69, *10*, pp. 7914–7932, 2020. DOI: 10.1109/TIM.2020.2983578.

[205] S. Luo and Y. Li, « Data Fusion of Multiple Spatio-Temporal Data Sources for Improved Localisation in Cellular Network », *in 2018 IEEE International Conference on Big Knowledge (ICBK)*, 2018, pp. 456–463. DOI: 10.1109/ICBK.2018.00067.

[206] J. Winter and W.-C. Lee, « KPT: A Dynamic KNN Query Processing Algorithm for Location-Aware Sensor Networks », *in Proceeedings of the 1st International Workshop on Data Management for Sensor Networks: In Conjunction with VLDB 2004*, Association for Computing Machinery, 2004, 119–124, ISBN: 9781450377959.

[207] S. Wagner, N. Fet, M. Handte, and P. J. Marrón, « An Approach for Hybrid Indoor/Outdoor Navigation », *in 2017 International Conference on Intelligent Environments (IE)*, 2017, pp. 36–43. DOI: 10.1109/IE.2017.22.

[208] A. Yaeli, P. Bak, G. Feigenblat, S. Nadler, H. Roitman, G. Saadoun, H. J. Ship, D. Cohen, O. Fuchs, S. Ofek-Koifman, and T. Sandbank, « Understanding customer behavior using indoor location analysis and visualization », *IBM Journal of Research and Development*, vol. 58, *5/6*, 3:1–3:12, 2014. DOI: 10.1147/JRD.2014.2337552.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Localisation à l'intérieur des bâtiments : étude du standard 802.11 Fine Timing Measurement.

Mot clés : 802.11az, FTM, indoor location, ranging

Résumé : La localisation à l'intérieur des bâtiments reste problématique. Les signaux GPS ou cellulaires ne pénètrent pas bien les structures bâties, et les techniques Wi-Fi basées sur des signaux broadcast ont perdu de leur efficacité avec l'apparition de smartphones soucieux de conserver l'énergie de leur batterie et la vie privée de leurs utilisateurs. Dans ce contexte, le protocole Fine Timing Measurement (FTM), défini dans la révision 2016 du standard IEEE 802.11, apparut comme une solution viable pour produire sur l'écran d'un client mobile un point représentant la position de ce client sur le plan de l'étage. Malheureusement, la création de ce protocole a porté l'accent sur les échanges de trames plus que sur l'environnement ou les conditions dans lesquelles ces échanges prendraient place. L'objet de cette thèse est donc d'étudier FTM en profondeur, pour comprendre comment il manifeste les propriétés d'un protocole de localisation moderne, c'est à dire capable de résoudre les problèmes de navigation à l'intérieur d'un bâtiment, sans exposer les données privées de l'objet qui l'utilise, et tout en permettant à l'infrastructure de localiser des objets mobiles ou d'obtenir des statistiques sur le trafic.

Cette thèse fait apparaitre les forces et les faiblesses de FTM, et propose des améliorations des techniques d'implémentation et du Standard IEEE 802.11 pour compenser les faiblesses de la version initiale de FTM. En particulier, cette thèse propose une méthode pour automatiser le positionnement des points d'accès (qui servent de répondeurs pour les clients mobiles) et leur permettre d'apprendre automatiquement leurs coordonnées géographiques, qu'ils peuvent ensuite communiquer aux clients mobiles. Cette thèse propose aussi une méthode pour contrecarrer les effets d'une technique d'intelligence artificielle qui permettrait d'identifier chaque client mobile à partir de ses échanges FTM. Cette thèse propose enfin plusieurs améliorations du standard IEEE 802.11, pour protéger l'infrastructure tout en permettant aux opérateurs de réseaux de bénéficier aussi de la localisation que permet FTM.

Title: Indoor Location: Study on the IEEE 802.11 Fine Timing Measurement Standard

Keywords: 802.11az, FTM, indoor location, ranging

Abstract: Indoor location remains challenging. GPS and cellular signals do not always penetrate buildings well, and legacy techniques that relied in Wi-Fi broadcasts have lost their luster with the explosion of personal devices focused on privacy and battery efficiency. In this context, Fine Timing Measurement (FTM), defined in IEEE 802.11-2016, appeared as viable solution to provide a blue dot inside. However, the Standard merely focuses on the frame exchanges, without considering the environment or the conditions where they would occur. This thesis aims at analysing FTM in depth, to understand how it fits into the mold of a modern location protocol, solving indoor navigation while ensuring end device privacy, but also allowing the infrastructure to track assets or collect analytic about foot traffic.

This thesis surfaces the strengths and weaknesses of FTM, and proposes implementation techniques and Standard enhancements to overcome the critical shortcomings. Among them, a method is proposed to facilitate the automation of Access point (responding anchors) deployment, allowing them to automatically learn their geo-position. Another method is proposed to counteract the efficiency of a learning machine capable of fingerprinting client stations solely based on their FTM exchanges. An augmentation of 802.11 is suggested to limit the possibility of ranging and location attacks on FTM, and another augmentation to the Standard is designed to allow the infrastructure to also benefit from the ranging exchange.