UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

AN ADAPTIVE FTM ALGORITHM

ALONSO FERNANDEZ MIRA

Thesis supervisor: RENÉ SERRAL GRACIÀ (Department of Computer Architecture)

Degree: Bachelor's Degree in Informatics Engineering (Information Technologies)

Bachelor's thesis

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Chapter 1

Abstract

In recent years there has been a surging requirement for precise and effective indoor localization measurements due to the emergence of innovative technologies such as the Internet of Things (IoT) [1] and smart home devices. Traditional localization technologies like GPS have been proven not to be precise in indoor environments. A new approach based on WiFi RTT (Round Trip Time) called FTM protocol (Fine Time Measurement) has gained popularity over the last years due to its low-cost implementation and wide distribution especially indoors.

Accuracy issues within FTM have been pointed out by studies such as [2] and [3]. In an environment where the signal bounces into walls and buildings, the signal is received through different paths. As the sampling interval of the radios is below the delay between the different paths, the receiver often sees a mixture of the signals over different paths. Therefore, it is not possible to distinguish the direct path (if it exists) from the reflected signals. The receiver measures a mixture of different signals with different amplitudes, phases, and probably also different frequencies.

Recent studies [2] analyzed the increase in accuracy by choosing different burst sizes and multiple positions. As expected, increasing the number of bursts in an FTM frame leads to a reduction in the error. However, this comes at the cost of consuming more channel time. The increase in accuracy is not directly proportional to the amount of channel time required to perform it. This thesis' objective will be to develop an algorithm capable of dynamically adapting its parameters to ensure, simultaneously, channel time optimization and precision in real-case scenarios.

Contents

Chapter 2

Introduction

Indoor location and ranging applications have become indispensable in various domains, from asset tracking in manufacturing facilities to smart building solutions. The demand for pinpoint accuracy in determining positions and distances within indoor environments has led to the development of Fine Time Measurement (FTM) protocol. This protocol, initially designed for ranging purposes, has evolved to encompass a wide range of indoor location-based applications [4], [5].

However, indoor environments introduce a myriad of challenges, factors such as signal reflections, interference, and varying signal propagation characteristics can lead to suboptimal performance. The complexities of indoor environments render traditional outdoor navigation systems, such as GPS, unreliable for indoor location and ranging [6]. This limitation has driven the exploration of alternative technologies and protocols, with FTM protocol emerging as a promising solution exclusively designed for indoor applications. Traditional FTM protocol have been designed with static parameters set during initial configurations and held constant throughout a session. While this approach may suffice in stable, well-understood environments, it falls short in dynamic, unpredictable indoor settings.

To address these challenges, this thesis embarks on an exploratory journey to investigate the profound effects of varying sets of parameters within FTM protocol. We focus on key parameters such as burst size, burst duration, ftm per burst, burst exponent, and delta ftm, the goal is to understand how these parameters, governing various aspects of FTM protocol, impact indoor location and ranging applications across different scenarios.

The objective goes beyond optimizing indoor location and ranging precision. It also involves making judicious and responsible use of the limited resource of channel time. Wasting precious channel time can lead to inefficiency, congestion, and ultimately an unsatisfactory user experience. Responsible channel time utilization

becomes a critical criterion for assessing the performance of FTM protocol in indoor localization and ranging.

We aim not only to examine the impact of these parameters in controlled scenarios but also to develop an algorithm that can dynamically adapt them in real-time, designed exclusively to the unique requirements of indoor localization and ranging applications.

The subsequent chapters provide a comprehensive overview of the parameters and methodologies employed in my research, specifically designed for indoor ranging applications. Through case studies and practical implementations, we aim to demonstrate the efficacy of dynamic parameter adaptation in FTM protocol within these specific contexts. These findings will illuminate the potential for reducing errors, enhancing precision, and ensuring that the limited resource of channel time is used responsibly in the realm of indoor localization and ranging, even as they find application in scenarios such as the Internet of Things (IoT).

This thesis encapsulates the evolving landscape of FTM protocol for indoor localization and ranging within the ever-changing digital environment, addressing the unique challenges of indoor positioning and ranging.

Chapter 3

Fundamentals

3.1 WiFi

Wi-Fi is a set of network protocols that enable wireless communication between devices such as laptops, smartphones, and more to connect to the internet. Wi-Fi uses radio waves to transmit data over the medium, when the data is received, it is interpreted based on a predefined set of parameters the way this information is transmitted is by using *Modulation Coding Schemes*, a standard that defines how data is formatted and encoded for transmission in wireless communication systems. It is used to determine the data rate of a wireless connection and is based on several parameters of a Wi-Fi connection between two stations [7].

3.1.1 Concepts and components of a Wi-Fi network

Wi-Fi networks have become an integral part of our daily lives, enabling wireless communication between various devices. Several key components must be mentioned to understand these networks' infrastructure:

- **Client Devices**: Devices such as smartphones, tablets, laptops, IoT devices, and more belong to this category, these are vital elements of the network[8].
- **Access Point (AP)**: Networking device that allows wireless devices to connect to a wired network. They have a unique identifier called BSSID which is based on its MAC and a user user-friendly identifier SSID which is used to advertise the network to the end users. APs can handle multiple simultaneous connections, allowing users to move freely within the network coverage area without experiencing network interruptions [9]

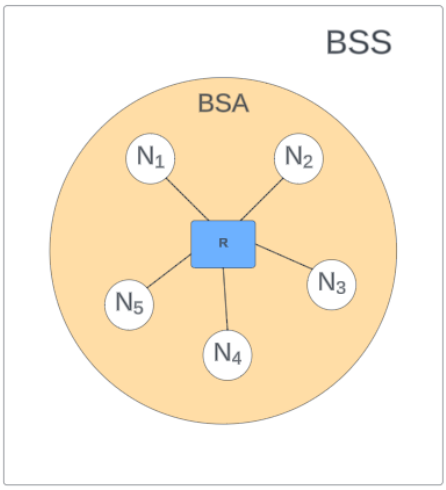

Figure 3.1 – Basic Service Set

- **Basic Service Set (BSS)**: Basic network topology, Figure 3.1, in which an Access Point (AP) serves as a gateway between the wired and wireless domains [8]
- **Basic Service Area**: Conceptual area in which members of a BSS may communicate [10].

3.1.2 Wi-Fi Channels and Bands

Wi-Fi protocols use two main frequency bands, 2.4GHz and 5GHz even though different standards include different frequency bands. The 2.4 band is then split into 14 different channels to split the band into smaller, more manageable parts.

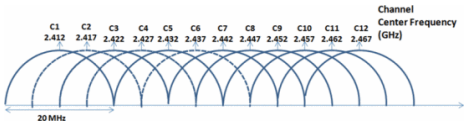

Figure 3.2 – WiFi Channels for 2,4GHz band [11]

This approach will help cope with issues such as congestion or interference. As can be seen in Figure 3.2, most of the 14 channels overlap each other, to avoid interferences only non-overlapping channels such as 1-5-9 will be used [11].

On the other hand, the 5GHz band does not have this overlapping problem, as a result, a higher transmission rate can be reached[12], channels are even combined to achieve higher speed [13], and the channels are usually less prone to suffer congestion. All these benefits come at the price of a lower reach[14] than the 2,4GHz band. When a station wants to transmit something over the wireless network, it

must wait until one of the channels is free. Choosing the right channel size for a wireless network band is not a trivial issue. The wider the channel, the more data can be pushed through it but an increase in the noise of the channel will also be perceived. Techniques such as CSMA/CD [15] are used to grant collision avoidance and fair distribution among the nodes of the channel time. If the Signal Noise ratio is reduced, the Modulation Coding Scheme rate will also be reduced.

3.1.3 Modulating Coding Schemes

MCS is an indexing used in Wireless LAN that defines the modulation technique for the binary strings of information sent over the wireless network and the Forward Error Correction (FEC) [16]. There are 32 MCS indexes from MCS-0 to MCS-31 [17]. Quadrature Amplitude Modulation (QAM) [18] is one of the most used modulation techniques to encode information in an RF wave. An RF wave has 3 main properties that we can control: amplitude, frequency, and phase. The phase shift represents how much the wave displaces horizontally.

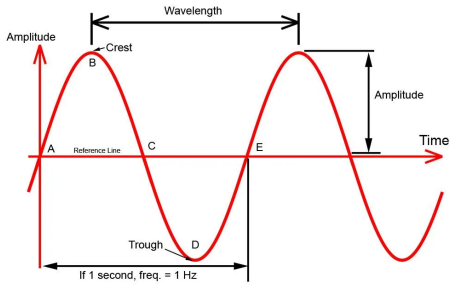

Figure 3.3 – RF properties

QAM is a combination of phase and frequency modulation [19]. The length of the symbols (unique combinations of phase and amplitude) determines the QAM modulation length, QPSK has 4 symbols with 2 bits/symbol, QAM16 has 16 symbols with 4 bits/symbol, and so on. These symbols can be mapped into a constellation diagram. Because of the noise, the transmitted symbols fluctuate between the ideal value, when the noise is too high, some symbols may be interpreted incorrectly.

The more symbols we want to include in our constellation diagram, the higher the SNR has to be to guarantee that the symbols will be interpreted correctly.

3.1.4 Wi-Fi Standards

The roots of WiFi can be tracked down to the 20th century when notable scientists such as Nikola Tesla and Guglielmo Marconi started experimenting with this concept. The foundation of modern Wi-Fi standards was laid by the Institute of Electrical and

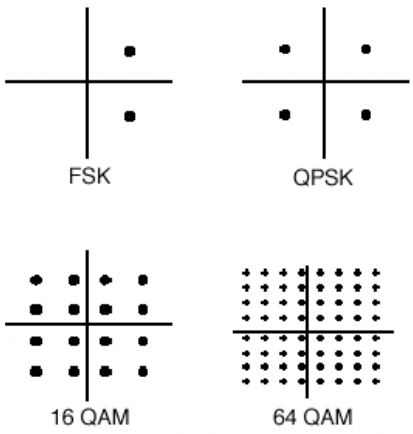

Figure 3.4 – Constellation diagrams

Electronics Engineers (IEEE) with the introduction of the 802.11 family of standards. These standards [20] established the framework for wireless networking and have since seen multiple iterations and enhancements.

- **802.11a**: The original 802.11 standard [21], introduced in 1997, operated in the 2.4 GHz frequency band and provided a maximum data rate of 2 Mbps. This marked the dawn of Wi-Fi technology.
- **802.11b**: Significant improvement in data rates reaching up to 11Mbps by introducing different modulating techniques [22]
- **802.11g**: Introduced in 2003 [23] it was a great milestone, increasing data rates up to 54Mbps.
- **802.11n**: The first modern WiFi standard, introduced in 2009, introduced the 5GHz band [24], reaching this was data transfer rates of up to 600Mbps. This was a turning point as it set the basis for integration with wireless devices such as smartphones, laptops, and tablets.
- **802.11ac**: In 2013, the 802.11ac standard made its debut, focusing on the 5 GHz frequency band [25]. This standard offered blazingly fast data rates, reaching up to 3.5 Gbps. With wider channels and more efficient data encoding, 802.11ac was tailored for high-bandwidth applications like streaming high-definition video and online gaming.
- **802.11ax**: Introduced in 2019 [26], one of its main innovations is the ability to support multiple devices simultaneously by using Orthogonal Frequency Division Multiple Access (OFDMA).

3.2 Indoor localization

Although Global Position Systems (GPS) and wireless E911 services [27] seem to solve the location and ranging problem, there are certain scenarios where these technologies cannot cope with the introduced error by specific physical phenomena. Hence, the results obtained under certain circumstances cannot be considered reliable, and new approaches to solve these edge cases must be explored. Among these edge cases, indoor conditions have become one of the main focus of researchers. Due to the exponential implementation and introduction of wireless technologies over the last few years, new approaches to indoor localization have been explored [28].

1. *Received Signal Strength Indicator*

   The Received Signal Strength-based technique [29] is one of the most basic, low-cost, and wide coverage methods up to date. On top of that it does not require any sort of specific hardware to apply these measurements. RSS is the actual signal strength received measured in decibel-milliwatts (dBm) or milliWatts (mW), the higher the RSS, the lower the distance between the receiving (Rx) and the transmitting (Tx) stations. Provided a simple path-loss propagation model [30], the distance d can be estimated as follows:

   $$RSSI = -10n \log_{10} d + A \tag{3.1}$$

   where n is the path loss exponent and A is the actual RSSI value at a reference value from the receiver.

   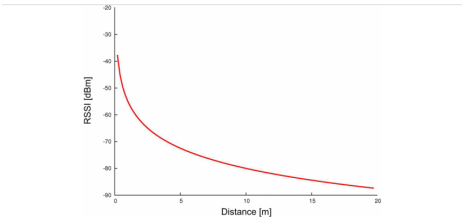

   Figure 3.5 – Expected relationship between RSSI and distance [31]

   However, most RSSI location-based protocols have been proven [32] [33] to be underperforming in real-case scenarios. Even though this method is cost-effective and easy to implement, it suffers from limitations in accuracy due to the physical phenomena of the environment.

2. *Fingerprinting*

The Fingerprinting location technique [34] uses the received signal strength (RSS) at the sampling locations to build a "radio map" for the target environment, this technique can be performed by any Access Point that supports Received Signal Strength. As a consequence, it is relatively simple to configure and does not require any specific hardware. Location fingerprinting performs well for non-line-of-sight (NLOS) circumstances and LOS environments are not required [35], but turns out to be only suitable for certain concrete scenarios where the conditions never change. This method comprehends two major steps:

(a) *Calibration Phase*: The area to be ranged in decomposed in a rectangular rid full of points where each one of these points will hold the RSS information of all the measurements collected during this phase, this data structure is called *Radio Map*[36]

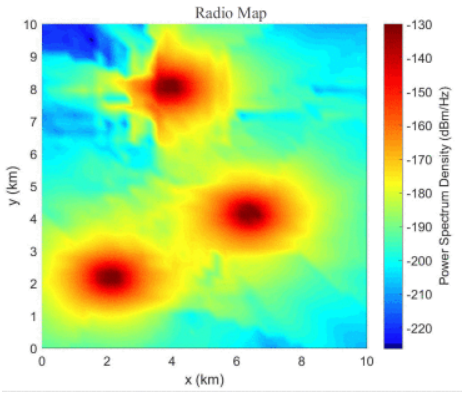

Figure 3.6 – Radio map obtained mapping the RSS in different positions in a venue [36]

(b) *Online*

Once the Radio Map is constructed, it can be used to compare the RSS we are experiencing from a device to estimate our position in the Map. With our RSS value, we can obtain a series of possible candidates for the position, deciding which one of the candidates is more likely to be the better approximation is a complex topic that has been a line of research through the last few years [35] [37], amongst them, probabilistic methods [38], K-Nearest-Neighbor and Neural Networks seem to be some of the most effective and promising methods.

3. *Angle of Arrival*

The Angle of Arrival (AoA) location technique estimates the position of a signal source by measuring the angle at which the signal arrives at a receiver

[39]. This method is commonly used in various applications such as wireless communication, radar systems, and localization systems [40]. Understanding how the AoA technique works involves analyzing the angular direction of the incoming signal and processing the measurements to determine the source's location. The AoA technique typically involves an array of antennas [39] or sensors that capture the incoming signal from different spatial angles. By processing the signals received by the array, the angle of arrival of the signal can be estimated. This estimation provides valuable spatial information about the location of the signal source. The angle of arrival (AoA)–based localization

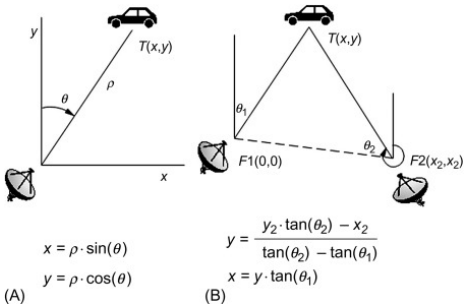

Figure 3.7 – The AoA is calculated based on the angles perceived by the APs' RF [41]

can be visualized with the help of Figure 3.7. For example, a graph depicting the AoA-based localization with two known access points (APs) shows how the angle measurements are utilized to determine the location of the signal source.

4. *Phase of Arrival*

The Phase of Arrival technique estimates the distance between nodes and the localization of a specific node based on the Phase Shift of different RF signals [42]. When employing PDoA for localization, the distance is predominantly influenced by the wavelength and the initial phases of the signals transmitted from the antennas [43]. For a single antenna, the distance is calculated as follows [44]:

$$d = \frac{\lambda}{2} \cdot (\frac{\phi}{2\pi} + n) \tag{3.2}$$

$$\lambda = \frac{c}{f} \tag{3.3}$$

where λ is the wavelength and ϕ is the phase shift and n is an integer. The phase shift can be expressed as:

$$\phi = \phi_{int} + \phi_{prop} = 2\pi \cdot (\frac{2d}{\lambda} - n) \tag{3.4}$$

The only way of obtaining a value for 2 is using at least 2 antennas canceling the ϕ_{int} component, for antennas at the same distance it can be expressed as:

$$d = \frac{c \cdot \Delta\phi}{4\pi\Delta f} \tag{3.5}$$

Whereas, for antennas placed at different distances from the observer it is expressed as:

$$d_1 = \frac{\lambda}{2} \cdot (\frac{\phi_1}{2\pi} + n_1) \tag{3.6}$$

$$d_2 = \frac{\lambda}{2} \cdot (\frac{\phi_2}{2\pi} + n_2) \tag{3.7}$$

$$\Delta d = d_1 - d_2 = \frac{\lambda \cdot \Delta\phi}{4\pi} + \frac{\lambda \cdot (n_1 - n_2)}{2} \tag{3.8}$$

A second antenna must be used to provide triangulation[43].

5. *Time Difference of Arrival*

   The Time Difference of Arrival (TDoA) localization method is a technique used to determine the position of an object by measuring the difference in signal arrival times at multiple spatially separated receivers. This method is based on the principle of multilateration and is characterized by high accuracy of localization [45], allowing for object tracking within a long effective range. TDoA is a passive technique that exploits the difference in signal arrival times at multiple receivers to localize and track emitting objects. The method does not require the time that the signal was sent from the target, only the time the signal was received and the speed that the signal travels [46]. Once the signal is received at two reference points, the difference in arrival time can be used to calculate the difference in distances between the target and the two reference points. The TDoA method is used in various applications, including underwater mobile node localization [47], and indoor positioning systems [48]. It is also used in real-time location systems (RTLS) and with Ultra-Wideband technology (UWB) [49]

3.3 FTM protocol

Since the appearance of WiFi technology, researchers have shifted their interest to new applications in this field. Over the years, several approaches towards indoor ranging and location using WiFi have been pursued [50]. These ideas were embraced enthusiastically by the community and after various upgrades [51] [52] [46] this approach has become a key positioning technology next to GPS used by mobile

devices. It was only in 2016 when the IEEE standardized these ideas into the so-called Fine Time Measurement Protocol [53].

This protocol is based on the *Time of Flight (TOF)* approach, the *Initiating Station* measures the time the packet has been traveling through the medium, in Figure 3.8 a simple example of a simple packet exchange and its ToF calculation can be seen. Notice how the processing time between packets is subtracted and the overall result is divided by two as the packet does a Round Trip.

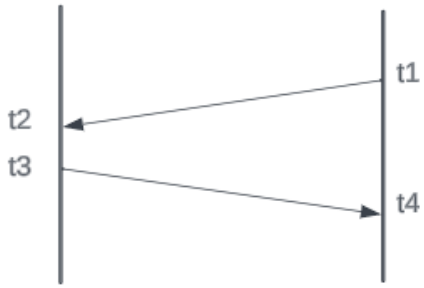

Figure 3.8 – Time of Flight

Once the ToF is known, the distance between the two nodes can be estimated as follows: $d = ToF \cdot 3 \cdot 10^8$

The FTM protocol allows for more precise ranging and localization measurements by obtaining the mean value of several estimations in each session. To start a session, the initiator node sends an FTM Request with the configuration of the parameters of the protocol embedded, if the *Responder Station* is up, it will answer with an ACK frame and only then, the *Responder Station* will start sending FTM frames that will be answered with ACK frames by the *Initiating Station*. The FTM frames the responder is sending to the initiator will contain the timestamps at which this station sent and received the previous frames. Every time the initiator receives the timestamps, an estimation of the distance between stations can be performed using the ToF approach, the RTT for n measurements is calculated as follows:

$$RTT = \frac{1}{n}[\sum_{k=1}^{n} t_4(k) - \sum_{k=1}^{n} t_1(k) - \sum_{k=1}^{n} t_3(k) + \sum_{k=1}^{n} t_2(k)] \tag{3.9}$$

an example of this exchange can be seen in the Figure 3.9.

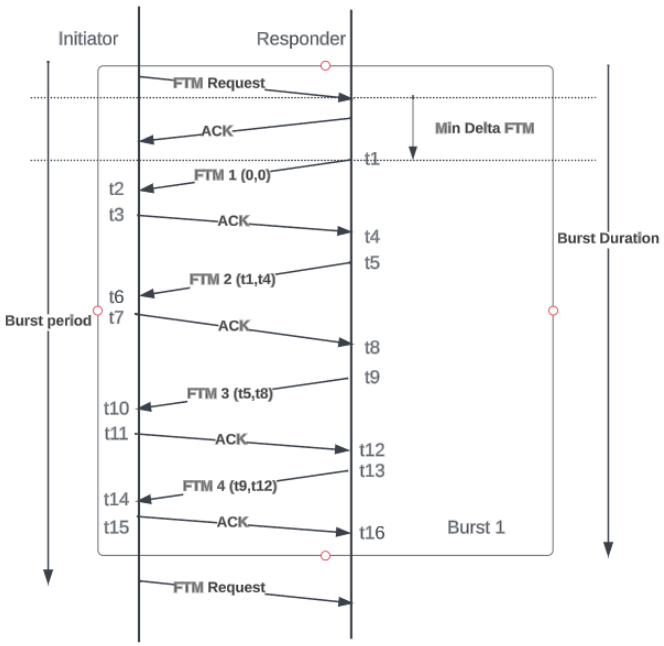

Figure 3.9 – FTM exchange

3.3.1 Parameters

The FTM protocol has several parameters that enable the Initiating Station to configure its behavior [54].

- **Burst Exponent**: Defines the number of times in which data is sent for each session (burst), each one of these bursts may contain multiple FTM messages.
- **Burst Duration**: Defines how long does a burst last.
- **Burst Period**: Defines how often is a burst triggered.
- **Ftm per Burst**: Defines how many FTM frames fit in each one of the bursts.
- **Min ΔFTM**: Specifies the time between two consecutive FTM frames.
- **TSF timer**: Clock shared between all the stations of the same BSS.
- **ASAP**: The ASAP Capable field indicates that the responding STA is capable of sending a Fine Timing Measurement frame as soon as possible; that is, the STA is capable of capturing timestamps associated with an initial Fine Timing Measurement frame and sending them in the following Fine Timing Measurement frame [54]

3.3.2 Error

The FTM protocol has several fonts of error [2], mainly related to the channel bandwidth and the multipath effect.

1. *Channel Bandwidth*

   It is commonly believed that the resolution of a ToF measurement depends directly on the sampling rate [55]. This is known as range binning when matched filters [56] are used to estimate the time of arrival with a sampling rate up to $f_s = 2B$. The space is divided into range bins of size $\frac{c}{f_s}$, and each one of these bins introduces a uniform error of σ_s^2.

   $$\sigma_s^2 = \frac{c^2}{12 f_s^2} \tag{3.10}$$

   In the case of WiFi technologies, different channel bandwidths influence enormously the precision of our measurements.

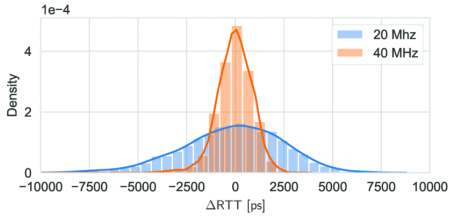

Figure 3.10 – Bandwidth error analysis [2]

   Using the equation 3.10, for WiFi using 20MHz bandwidth, a uniform error of $(4.68m)^2$ is introduced. On the other hand, using a channel bandwidth of 40MHz , the uniform error shrinks down to $(2.16m)^2$. The distribution of the error on the measurements can be seen in Figure 3.10

2. *Multipath*

   In a real-case scenario, wireless signals are influenced by the multipath effect. The radio frequency signal bounces off surfaces in the environment and as a consequence, the receiver perceives the signal as arriving through many different paths as can be seen in Figure 3.11. All of these reflected components arriving at the receiver (but the Line of Sight component) have different phases and amplitudes due to their reflection and diffraction, this effect is called fading [57].

   Multipath affects the ToF measurements even if a direct Line of Sight component is received as it is not uncommon for reflected components to have

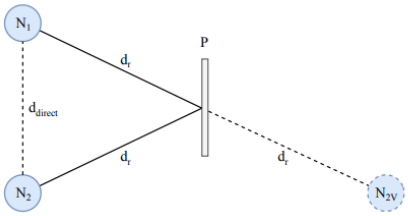

Figure 3.11 – A reflective surface P creates a multipath environment with known propagation paths[58]

higher power than the actual LoS component [59]. Receiving multiple signals simultaneously entangles the proper detection of packet arrivals. Take a closer look at the results obtained in [60], where several ToF measurements were taken in a car parking where the LoS component was not guaranteed due to cars, columns, and other objects; it can be seen in Figure 3.12 how there is a tendency to underestimate the distance between the stations because of the effect of multipath (note that the ground truth has been subtracted and hence the channel bandwidth is not affecting the measurements).

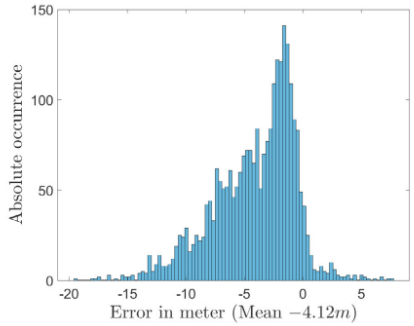

Figure 3.12 – Multipath error analysis [2]

3.4 ns3

As networks become more complex and further research on different network-related technologies keeps emerging, the need for reliable, scalable, and highly precise network simulators is critical [61]. Even though we are witnessing rapid development in large-scale testbeds [62], simulation still plays a huge factor in research as it provides, among many other properties, quick reproducibility in research. For a long time, ns2 simulator[63] was the tool mostly used in research, and despite its success in the industry and broad extension, the idea of a new version of this simulator started to sound around 2005. The main focus of this new version was to make the

models used to represent abstractions of the networks more realistic compared to the actual software implementation of different network devices. In July 2006, ns3 network simulator was released. ns3 is a network simulator working at the packet level, it is implemented in C++, which makes it easier to embed C code used in many physical factors, relating to this paper, it is relevant to note that ns3 does not natively support the Fine Time Measurement Protocol[2].

3.4.1 Fine Time Measurements in ns3

The Fine-Time Measurement (FTM) protocol, introduced in IEEE 802.11-2016, uses radio frequency-based time-of-flight estimation, enabling precise indoor ranging and positioning. FTM-ns3 is an extension presented in [2] for the widely used ns-3 network simulator that supports the 802.11 FTM protocol. Using results from experiments with real FTM hardware, error models have been developed to analyze the impact of channel bandwidth and multipath channel propagation on the performance of FTM-based ranging or localization schemes. The FTM-ns3 extension allows researchers to study the most relevant factors influencing the precision of FTM-based ranging, introduces an empirical error model based on extensive lab experiments, and evaluates the FTM-ns3 system in a proof-of-concept study for ranging and localization

3.4.2 Error Models

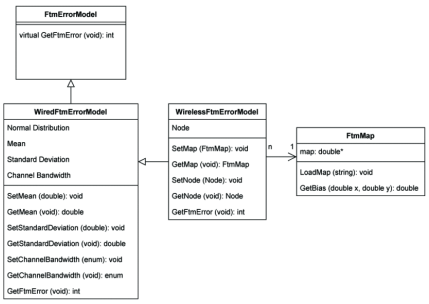

Figure 3.13 – Error Model diagram class [2]

FTM-ns3 supports different error models: wireless, wired, and wireless signal strength 3.13. These different options are modeled specifically to represent the physical phenomena that affect the precision of our measurement in real life. In this study, we will exclusively employ the wireless model, as it more accurately reflects the specific use case we intend to investigate, this error model will be dependent on the Channel Bandwidth and the Multipath effect as expressed in 3.11, where the

received RTT is expressed as an addition of the actual RTT and the error components.

$$RTT' = RTT + h + w \tag{3.11}$$

3.4.3 Mobility Models

In ns-3, MobilityModel objects track the evolution of position concerning a Cartesian coordinate system. The mobility model is typically aggregated to an *ns3::Node* object and queried using GetObject<MobilityModel>() [64]. The base class MobilityModel is subclassed for different motion behaviors. There are several types of mobility models in ns-3, each representing different motion behaviors:

1. *ConstantAccelerationMobilityModel*: The current acceleration does not change once it has been set and until it is set again explicitly to a new value.

2. *ConstantPositionMobilityModel*: The current position does not change once it has been set and until it is set again explicitly to a new value.

3. *ConstantVelocityMobilityModel*: The current speed does not change once it has been set and until it is set again explicitly to a new value.

4. *ConstantVelocityMobilityModel*: The current speed does not change once it has been set and until it is set again explicitly to a new value.

5. *GaussMarkovMobilityModel*: This is a 3D version of the Gauss-Markov mobility model. Unlike the other mobility models in ns-3, which are memoryless, the Gauss-Markov model has memory.

6. *HierarchicalMobilityModel*: This model allows for hierarchical mobility patterns

7. *RandomDirection2dMobilityModel*: mobility model that simulates nodes moving in a 2D space with random directions, the nodes under this mobility model pause for a specified amount of time in constant intervals of time.

The MobilityHelper class is used to assign positions and mobility models to nodes. Most ns-3 program authors typically interact with the mobility system only at configuration time. However, various ns-3 objects interact with mobility objects repeatedly during runtime, such as a propagation model trying to determine the path loss between two mobile nodes. The MobilityModel class also provides methods to get the current position and velocity of an object, get the distance from another mobility model, and set the position of the object.

3.4.4 Kalman Filer

The Kalman Filter recursively estimates the state variables in a noisy linear dynamic system by minimizing the mean-squared estimation error of the current state as noisy measurements are received and the system evolves in time. Each update provides the latest unbiased estimate of the system variables together with a measure of the uncertainty of those estimates in the form of a covariance matrix. Since the updating process is fairly general and relatively easy to compute, the Kalman filter can often be implemented in real-time[65].

This algorithm is widely used in a broad array of fields such as Global Positioning System (GPS) [66], robotics [67], computer vision [67], and neural networks [68].

Kalman filters are used to estimate states on linear dynamic systems, the *process model* defines the evolution of the state based on linear dynamical systems from time $k-1$ to time k as :

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1} \tag{3.12}$$

where F is the state transition matrix applied to the previous state vector x_{k-1}, B is the control-input matrix applied to the control vector, u_{k-1} and w_{k-1} is the process noise vector.

The process model is paired with the *measurement model* that describes the relationship between the state and the measurement at the current time step k as:

$$z_k = Hx_k + v_k \tag{3.13}$$

Where z_k is the measurement vector, H is the measurement matrix and v_k is the measurement state vector that is assumed to be zero-mean Gaussian with the covariance R. The objective of the Kalman Filter is to come up with an estimate of x_k at a time k giving the original estimate x_0, the series of measurements $z_0, z_1, ..., z_k$ and the system information F, B, H, Q and R [69].

The Kalman filter employs feedback control to estimate a process. It first estimates the process state at a given time and then incorporates feedback in the form of measurements. Consequently, the Kalman filter equations can be categorized into two groups: *time update equations* and *measurement update equations*. The time update equations are tasked with estimating the subsequent state, whereas the measurement update equations handle the feedback process by integrating new measurements into the estimate, thereby enhancing future estimations.

3.5 Related Work

Several studies have proposed different methods to improve the accuracy of FTM-based indoor positioning. For instance, a two-step fusion method of Wi-Fi FTM for indoor positioning has been proposed, which involves single-point positioning and a second step based on geometric principles [70]. Another study proposed a Wi-Fi RTT positioning approach based on Line of Sight (LOS) compensation and trusted Non-line-of-sight (NLOS) recognition, which improves the performance of indoor positioning [71].

In addition to FTM, other wireless signals such as Wi-Fi, ZigBee, RFID, BLE, UWB, LoRa, Sigfox, and NFC are widely used in indoor positioning research. Various algorithms for indoor positioning have been proposed, including proximity algorithms, triangulation algorithms, multipoint positioning algorithms, maximum likelihood algorithms, and fingerprint positioning algorithms [72].

Adaptive algorithms have also been proposed to improve the accuracy of indoor positioning. For instance, an adaptive calibration ranging algorithm based on Logarithmic Distance Path Loss Model (LDPLM) and Received Signal Strength Indication (RSSI) has been proposed, which consists of two parts: a coefficient adaptive algorithm and an error correction algorithm [73] [73]. Another study proposed an adaptive trilateration algorithm for RSSI-based indoor positioning systems, which adapts to changes in the indoor environment [74].

In conclusion, the field of indoor ranging and localization has seen a variety of methods and algorithms being proposed, with a focus on improving accuracy and adaptability. The adaptive algorithm for Fine Time Measurements proposed in this thesis aims to contribute to this body of work by dynamically adapting the parameters of the Fine Time Measurement Protocol for indoor ranging and localization.

Chapter 4

Simulation

The pursuit of robust and efficient wireless communication systems has induced the development and evaluation of various protocols and models. In this pivotal chapter, the core of our research is presented, presenting the results obtained through extensive simulations using the ns3 simulator. The focal point of our investigation lies in understanding the Fine Time Measurement (FTM) protocol under the influence of four distinct mobility models: *Fixed Position*, *Circle Velocity*, *Circle Mean*, and *Brownian*. As the preliminary results from our simulation indicate in figure 4.1, a quite noticeable difference in the error distribution can be observed when using different mobility models. Hence, finding out how the different parameters affect our measurements and estimating which model represents better the mobility mode of our initiating station is crucial to achieving a performance enhancement of the FTM protocol.

4.1 Mobility Model Overview

1. *Fixed Position*

   In the Fixed Position model, the receiving and initiating nodes are statically positioned throughout the simulation, maintaining a constant distance of 10m from each other. This model introduces a unique element of constancy, allowing us to examine the impact of unchanging spatial relationships on the adaptive FTM protocol. The static nature of the nodes enables the evaluation of the protocol under conditions of unvarying multipath scenarios.

2. *Circle Mean*

   Moving beyond the rigidity of Fixed Position, the Circle Mean model introduces dynamicity. The initiating node spins around the responding node at a fixed radius of 10m, making measurements at predefined positions. After completing

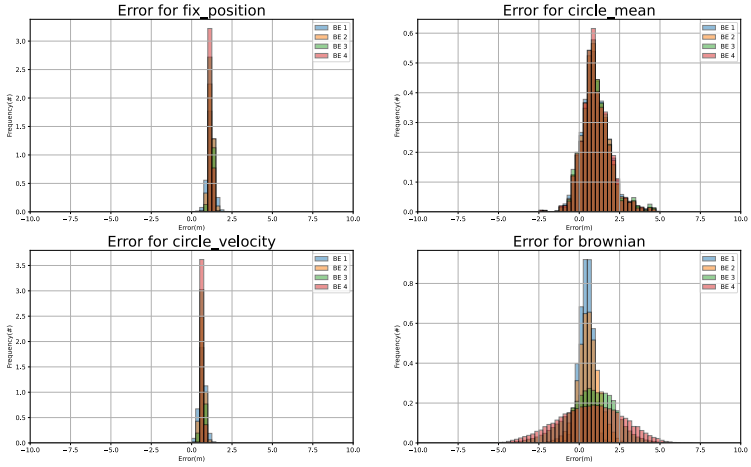

Figure 4.1 – Error Histogram for different mobility models

a circuit, the initiating node returns to the same set of positions, ensuring a consistent distance between nodes. This model enables us to explore the protocol's performance in scenarios where nodes traverse predefined paths.

3. *Circle Velocity*

   Building upon the Circle Mean model, the Circle Velocity introduces a dynamic twist by eliminating the pauses during measurement, the same radius (10m) is used in this model as well. The initiating node, while circling the responding node, does not come to a standstill during measurements. This dynamic behavior adds a layer of complexity, altering session times and measurement positions based on the chosen FTM parameters. Consequently, the effect of varying FTM-related parameters on the protocol is investigated under continuous motion.

4. *Brownian*

   Venturing into a more realistic realm, the Brownian model introduces randomness in the movement of the initiating node. It selects random directions, moves in a straight line until encountering the boundary of the 2D plane, and reflects its direction. The inherently unpredictable nature of Brownian motion adds a dynamic element to the distance between nodes, challenging the adaptability of the FTM protocol to scenarios with constantly changing spatial relationships.

4.2 Performace Metrics for Comparative Analysis

In the pursuit of a thorough evaluation of the Fine Time Measurement (FTM) protocol, a comprehensive set of performance metrics is employed. Each metric serves as a crucial lens through which the protocol's behavior is analyzed, shedding light on its efficacy under diverse conditions. The following subsections elaborate on each metric, untangling the complexity of their significance in our comparative analysis.

1. **Session Time**

   Session Time, a critical metric in our analysis, encapsulates the duration from the initiation to the conclusion of a communication session. Specifically, in the context of the FTM protocol, it delineates the time during which ranging results are received. Striking a balance is crucial; while high-frequency results (low session time) for dynamic scenarios are desired, avoiding oversampling during stationary phases is a big objective as well. This metric serves as a key indicator of the protocol's adaptability to varying temporal dynamics, offering valuable insights into its agility and efficiency.

2. **Error**

   Error, another pivotal metric, quantifies the disparity between the measured distance and the actual distance. The primary objective is to minimize this discrepancy, ensuring the precision and accuracy of the FTM protocol. A low error rate signifies the protocol's proficiency in providing reliable distance measurements, essential for the overall effectiveness of wireless communication systems.

3. **Channel Time**

   Channel Time is the subset of Session Time that encompasses the duration actively spent on FTM measurements and acknowledgments. This metric provides a focused perspective on the temporal aspects of communication, offering insights into the efficiency of the protocol's engagement with the communication channel. A careful examination of Channel Time allows us to assess resource utilization, particularly how effectively the protocol utilizes available time for productive communication.

4. **Channel Usage**

   Channel Usage, a complementary metric to Channel Time, represents the percentage of Session Time dedicated to FTM measurements. Calculated as Channel Time divided by Session Time, this metric unveils the proportion of the overall session devoted to crucial distance measurements. It is particularly

significant in scenarios with limited bandwidth, shedding light on the protocol's judicious allocation of time for essential communication activities.

5. **Efficiency**

   Efficiency, a synthesized metric, establishes an arbitrary relationship between Error and Channel Time. Calculated as

$$Efficiency = \frac{1}{ChannelUsage \cdot Error} \tag{4.1}$$

   , encapsulates the protocol's ability to balance precision with resource utilization. This metric offers a holistic view, taking into account the trade-offs between error minimization and efficient channel engagement. A higher efficiency score reflects a protocol's adeptness in achieving optimal performance across these key dimensions.

As I move forward in my analysis, a detailed study of these metrics will be performed to understand how different ways of moving around and adjusting FTM settings affect how well the protocol works. The information gathered from this detailed evaluation will help shape the design and improvement of protocols in the future, making location and ranging through FTM more efficient and precise.

Chapter 5

Parameter Performance Analysis

During this chapter, a study over 2 million FTM session simulation data will be provided. These session simulations will be distributed equally amongst the different mobility models and each mobility model will be simulated with over 50 different parameter configurations.

5.1 Burst Duration

The FTM parameter defines the duration of individual bursts within a session, making it dependent solely on time metrics. Throughout this section, we will observe that mobility models with relative distance independence among nodes.

Different values for Burst Duration impact the Session Time metric. Irrespective of the employed mobility model, the correlation between Session Time and Burst Duration remains linear, as illustrated in Figures 5.1 and 5.2.

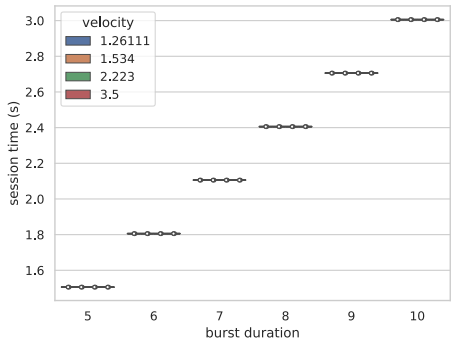

Figure 5.1 – Burst Duration Effect on Session Time for Brownian model

For every mobility mode used during this simulation, burst duration seems not to have any effect on *Channel Time* metric, Figure 5.3. This result was to be expected as

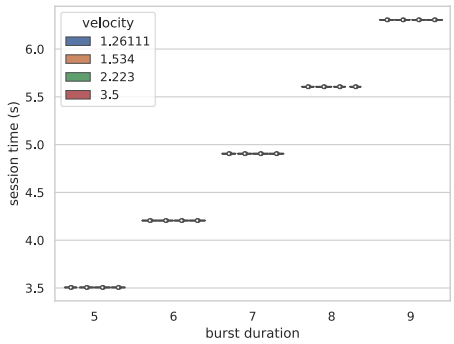

Figure 5.2 – Burst Duration Effect on Session Time for Circle Velocity model

burst duration does not have any influence over how much time is spent performing FTM measurements.

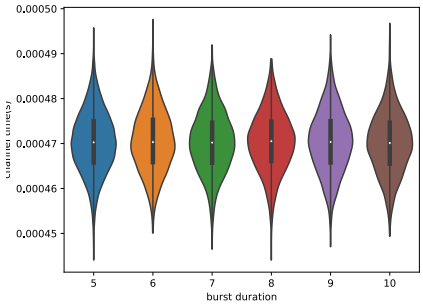

Figure 5.3 – Burst Duration Effect on Channel Time for Fix Position model

On the contrary, the metric called *Channel Usage* offers a relative measurement akin to what Channel Time provided. However, it is influenced by Session Time, which is linearly affected by Burst Duration. Consequently, Channel Usage exhibits an inverse exponential correlation with Burst Duration across all mobility models, as depicted in Figures 5.4 and 5.5. Notably, in the Brownian model, while the mean value behaves as anticipated, varying velocities result in different distributions for Channel Time values, as shown in Figure 5.6.

Moving on from time-related metrics, the error metric is one of the most interesting for any ranging or localization algorithm. Comparing it to burst duration, in all of the models where the distance between nodes remains constant no difference can be easily perceived, Figure 5.7; a small difference is only noticeable when using the Brownian model with high ftm parameter values, the longer the session lasts the longer the node gets to move during transmission time, this effect can be observed in Figure 5.8

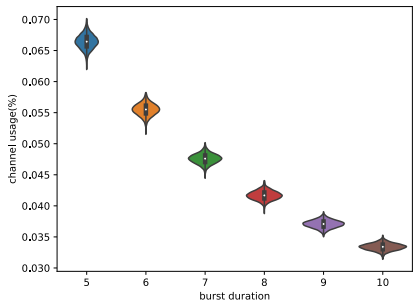

Figure 5.4 – Burst Duration Effect on Channel Usage for Fix Position model

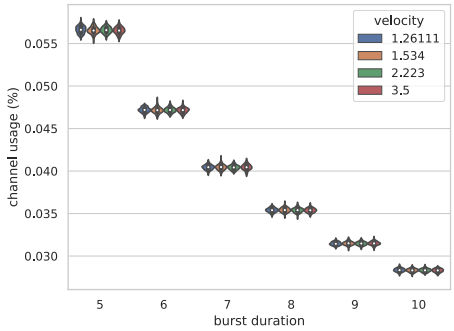

Figure 5.5 – Burst Duration Effect on Channel Usage for Circle Velocity model

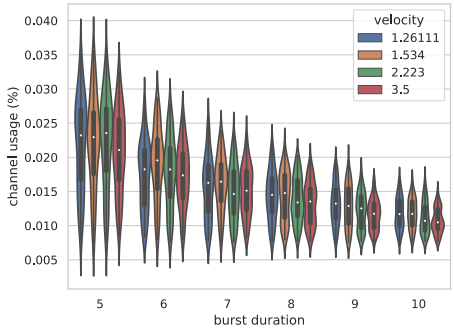

Figure 5.6 – Burst Duration Effect on Channel Usage for Brownian model

Finally, in the majority of models, efficiency tends to improve with higher burst duration values. The efficiency is inversely dependent on the relationship $ChannelTime \cdot Error$, as previously demonstrated; Channel Usage decreases exponentially, and there is no noticeable improvement in error, and in some cases, it may even degrade. Consequently, we may observe a slight enhancement in efficiency in Figure 5.9 or no effect at all.

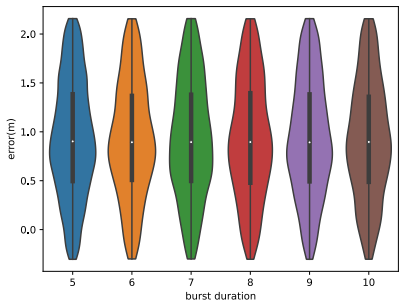

Figure 5.7 – Burst Duration Effect on Error for Circle Mean model

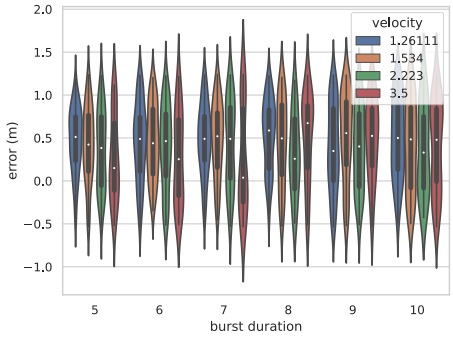

Figure 5.8 – Burst Duration Effect on Error for Brownian model

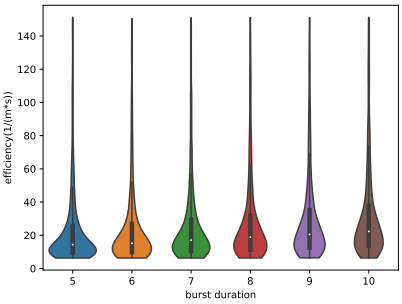

Figure 5.9 – Burst Duration Effect on Efficiency for Brownian model

5.2 Burst Period

The Burst Period parameter determines the frequency at which an FTM burst is transmitted. It does not alter the duration, quantity, or any other factor associated with the FTM session except for the frequency of burst initiation. As depicted in upcoming Figures, the impact of this parameter is generally negligible; in the majority

of cases, minimal to no variation was noted in the data collected for different metrics across diverse mobility models.

Given the aforementioned reasons, Session Time remains entirely unaffected by Burst Period values. This effect is evident in Figure 5.10.

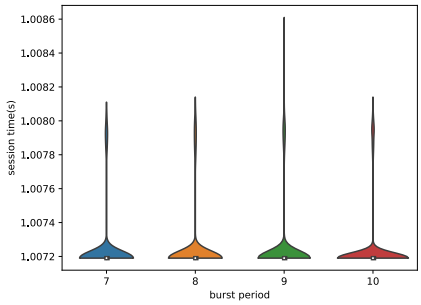

Figure 5.10 – Burst Period Effect on Session Time for Fix Position model

Likewise, Channel Time and Channel Usage exhibit complete independence from the Burst Period. Figures 5.11 and 5.12 illustrate that the session duration and the FTM transmission duration do not demonstrate any distinct behavior based on the Burst Period.

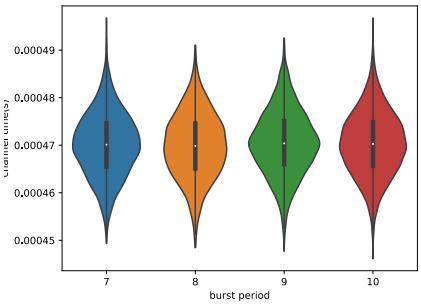

Figure 5.11 – Burst Period Effect on Channel Time for Fix Position model

Similar to the other metrics discussed in this chapter, it is challenging to assert any significant difference in the error distribution based on Burst Period. As evident in Figure 5.13, while the distributions are not precisely identical, claiming any substantial difference is not straightforward either.

5.3 Ftms per Burst

Let's another crucial FTM parameter: "FTM per Burst." This parameter's influence on metrics like error, channel time, session time, and channel usage varies across

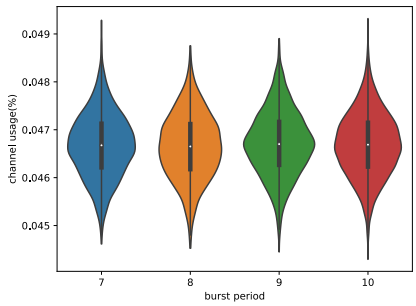

Figure 5.12 – Burst Period Effect on Channel Usage for Fix Position model

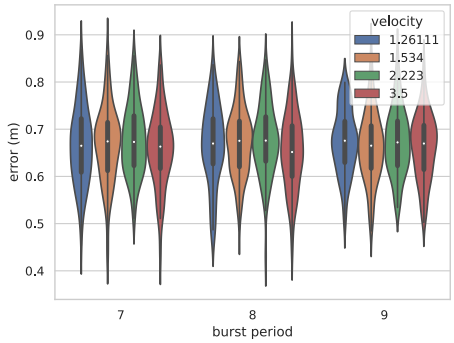

Figure 5.13 – Burst Period Effect on Error for Circle Velocity model

distinct mobility models—specifically, Circle Mean, Circle Velocity, Fix Position, and Brownian.

In terms of its association with Channel Time, we observe a consistent linear increase for every model as FTM per Burst rises, as depicted in Figure 5.14. This aligns with the expected behavior, indicating that higher FTM per Burst values lead to a proportional increase in Channel Time.

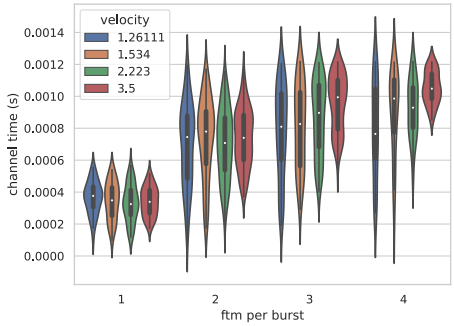

Figure 5.14 – Ftm per Burst Effect on Channel Time for Brownian model

Likewise, the connection between FTM per Burst and Session Time shows a linear trend across all models, suggesting a direct impact on the duration of FTM sessions. This linear correlation is evident in Figure 5.15, reinforcing the idea that elevated FTM per Burst values result in longer session durations.

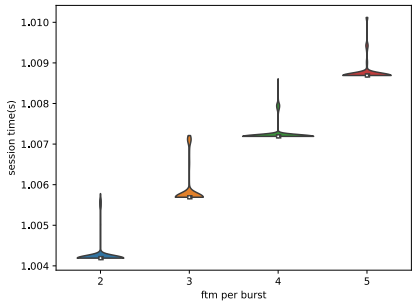

Figure 5.15 – Ftm per Burst Effect on Session Time for Fix Position model

Channel Usage appears to increase linearly across every mobility model even though some increase in deviation can be perceived in the Brownian model when the moving speed of the node changes. This effect is described in Figures 5.16 and 5.17.

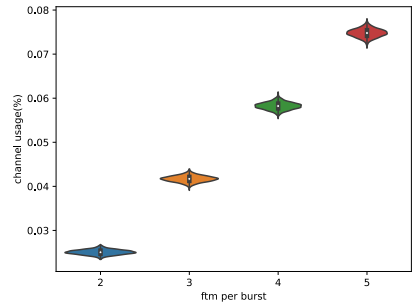

Figure 5.16 – Ftm per Burst Effect on Channel Usage for Fix Position model

An exploration of the dependency on error reveals intriguing insights. In models where the distance between nodes remains constant, such as Circle Mean, Circle Velocity, and Fix Position, an increase in FTM per Burst enhances accuracy and reduces deviation in the error metric, as demonstrated in Figure 5.18. However, this pattern deviates from the Brownian model. The theoretical improvement of the measurement by adding extra ftm per burst to each one of the bursts is compensated by the error that the mobile nodes introduce, Figure 5.19

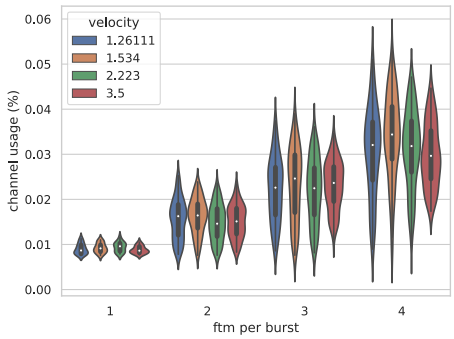

Figure 5.17 – Ftm per Burst Effect on Channel Usage for Brownian model

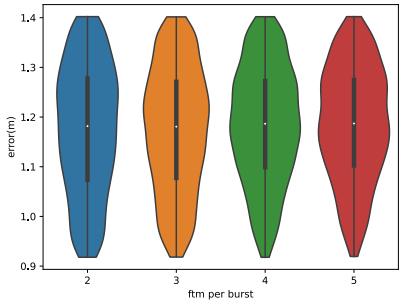

Figure 5.18 – Ftm per Burst Effect on Error for Fix Position model

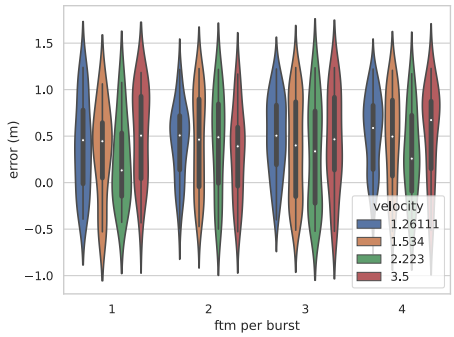

Figure 5.19 – Ftm per Burst Effect on Error for Brownian model

5.4 Burst Exponent

Wrapping up this chapter, we will explore the analysis of the FTM parameter referred to as the "burst exponent." This parameter is of significant importance as it directly influences the number of measurements taken in each session and operates exponentially (the number of bursts sent in a session is always a power of 2).

On one hand, the data we collected suggests that this parameter exhibits an exponential relationship with Session Time, as illustrated in Figure 5.20. This aligns with the initial hypothesis, where an exponential increase in the number of bursts leads to a corresponding exponential increase in session time, considering each burst has the same duration.

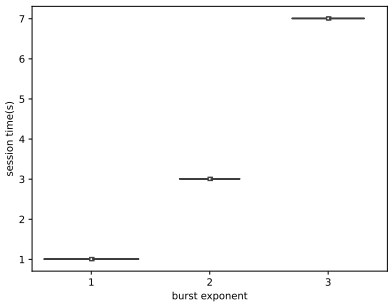

Figure 5.20 – Burst Exponent Effect on Session Time for Fix Position model

Consequently, it seems reasonable to argue that since the Session Time is increasing exponentially, so should the Channel Time. Indeed, the channel Time also has an exponential relationship with Burst Exponent, this relationship is exemplified in Figure 5.21

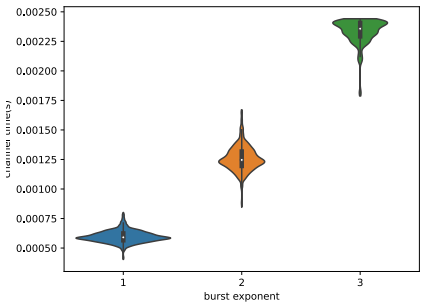

Figure 5.21 – Burst Exponent Effect on Session Time for Fix Position model

Conversely, predicting the behavior of Channel Usage is not as straightforward. While it might be logical to assume that Channel Usage should remain relatively constant, this would only hold if the bursts were triggered consecutively without any delay. However, this is typically not the scenario, as each burst is spaced apart by the Burst Period minus the Burst Duration. Consequently, the idle time introduced with each burst adversely affects our Channel Usage metric, as depicted in Figure 5.22.

The dependency on error is particularly intriguing. In models such as Circle Mean, Circle Velocity, and Fix Position, where the distance between nodes remains constant,

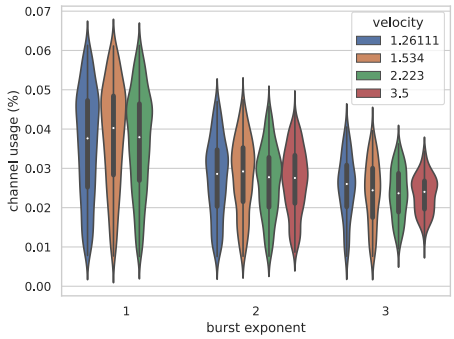

Figure 5.22 – Burst Exponent Effect on Channel Usage for Brownian model

an increase in Burst Exponent appears to enhance accuracy and reduce deviation in the error metric, as illustrated in Figure 5.23. However, the same pattern does not hold for the Brownian model. In this case, the variable distance leads to a different outcome, where a higher burst exponent implies longer sessions. Consequently, longer sessions allow nodes to move away (or towards each other) for an extended period, as depicted in Figure 5.24. This results in imprecise measurements, and even the ground truth is not truly representative.

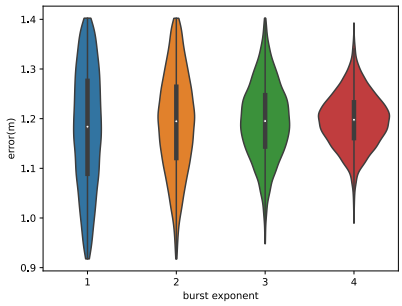

Figure 5.23 – Burst Exponent Effect on Channel Usage for Fix Position model

5.5 Overview

The analysis reveals that the dependency on error is particularly intriguing, as an increase in FTM per Burst enhances accuracy and reduces deviation in the error metric for models where the distance between nodes remains constant, such as Circle Mean, Circle Velocity, and Fix Position. However, this pattern deviates from the Brownian model, where the theoretical improvement of the measurement by adding extra FTM per Burst to each burst is compensated by the error that the mobile nodes introduce. In conclusion, the analysis provides valuable insights into

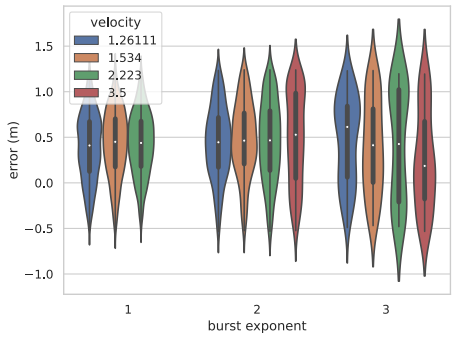

Figure 5.24 – Burst Exponent Effect on Channel Usage for Brownian model

the impact of various FTM parameters on different metrics for different mobility models. It underscores the need for a nuanced understanding of the relationships between these parameters and the resulting metrics, especially in the context of real-world applications. The findings can potentially inform the optimization of FTM parameters for improved performance in practical scenarios. The detailed analysis and visual representations of the data in the provided text offer a comprehensive understanding of the topic.

Chapter 6

Algorithm Analysis

In this section, we will analyze the outcomes derived from the information collected in the preceding chapters. Concluding these findings, various adaptive algorithm approaches will be formulated and their respective performances scrutinized. The first step is finding a comprehensive way of approaching the real case scenarios of the mobility models that we have been working with. Two basic states will be considered: *stationary* and *dynamic*. For the first state, the results drawn in the fixed position mobility model; we will be using the configuration that has been proven to be more effective under a fixed position mobility model. In contrast, during the dynamic state, the distance between nodes will be unpredictable and we will assume that the node movement is random and its future position cannot be predicted based on previous measurements (in other words, Kalman Filtering will not be used as it is considered to be out of the scope of this paper).

6.1 Algorithm

The formulation of a strategy for state estimation and the subsequent algorithmic treatment of state transitions have emerged as nontrivial tasks throughout this thesis development. The adopted strategy to identify a state change involves the introduction of a novel transition state called *transition*. This state serves as an intermediary phase between primary states. When the algorithm is designated the transition state, it makes use of the values assigned by the preceding active state. A visual representation of the state transition is displayed in Figure 6.1. This workaround ensures that the algorithm does not change its parameters as soon as a potential change is perceived, as a result, 2 consecutive state changes are required to change state, otherwise, the algorithm will return to its previous state.

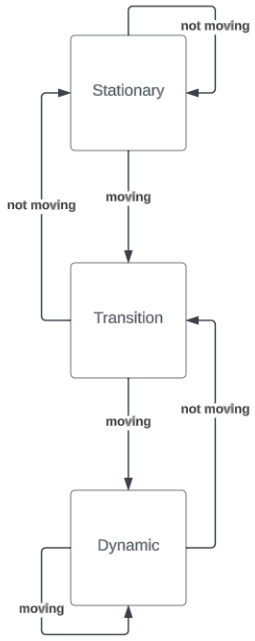

Figure 6.1 – Transition between states

It is a tremendously important matter to find a reliable procedure to detect state changes. The first approach was, for each measurement, to compare the current with the former ranging measurement and decide the mobility state of the node based on a static arbitrary percentage. The challenge with relying on this percentage is that, while it may offer a reasonably accurate representation of a particular mobility model with a defined set of parameters, even the slightest adjustment to these parameters can significantly alter the resulting distribution. This is a shred of concrete evidence that this model lacks elasticity and a more adaptive procedure is needed. The solution to this issue is to calculate this value dynamically, that is, every time the parameters or the state changes, a new value will be calculated (or loaded from memory in case it was already calculated). This value that the algorithm will calculate is the standard deviation of the active mobility model with the current configuration of parameters. Even though our data distributions do not correlate with normal distributions, these values hold 3/4 of all the obtained values. Hence, the chances of performing a wrong state change are $\frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16} = 6,25\%$. A flowchart representing the decision-making process of the algorithm can be seen in Figure 6.2

Given the current state of the algorithm, the standard deviation (which is dependent on the current state and the parameters the algorithm is running), and the previous state, visually in Figure 6.2:

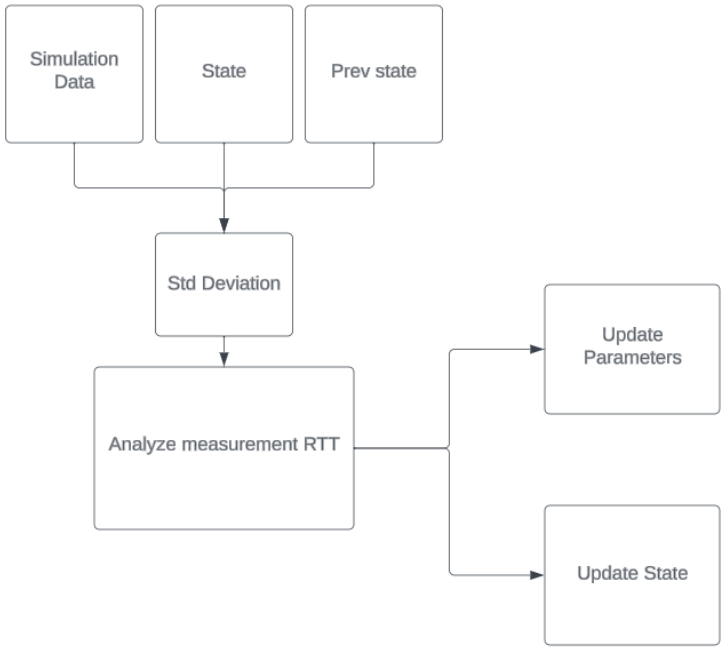

Figure 6.2 – Algorithms Flowchart

```
1 void FtmAdaptiveRanger::Analysis(){
2     bool config_change = false;
3     std::string new_state = state;
4     if (hist_rtt.size() > 2)
5     {
6         double rtt_change_percentage = abs(hist_rtt[0].second - hist_rtt[1].second) / hist_rtt[1].second;
7         if (rtt_change_percentage > std_deviation)
8         { // node IS moving
9             if (!transition)
10                HandleTransitionBrownian(new_state, config_change);
11            else if (state == "fix_position")
12                transition = !transition;
13        }
14
15        else
16        { // node is NOT moving
17            if (!transition && state == "brownian")
18                transition = !transition;
19
20            HandleTransitionFixPosition(new_state, config_change);
21        }
22    }
23
24    state = new_state;
```

```
25     LoadStatisticalVariables();
26 }
```

Every time a measurement is received, method *Analysis()* is executed. Provided that the new measurement's RTT combined with the other parameters fulfill the specified conditions to switch state the future state and parameters change consequently. Regardless of the overall result of the operation, after every measurement is received the method *LoadStatisticalVariables()* is executed which loads the proper standard deviation values for the current parameter set and mode.

```
1 // based on the "parameters" and "state" (filter ), updates the value of the
2 // following global variables:
3 //     -distribution_mean
4 //     -standard_deviation
5 void FtmAdaptiveRanger::LoadStatisticalVariables()
6 {
7
8     std::string map_key = GenerateParameterKey();
9     std::map<std::string, std::map<std::string, double>> *statistical_data =
10        (state == "brownian")
11            ? &brownian_statistical_data
12            : &fix_position_statistical_data; // update the pointer to the current data structure
13
14    if ((*statistical_data).find(map_key) == (*statistical_data).end())
15    { // its the first time this configuration is used, lets calculate the statistical values an store them
16        double mean_error = 0;
17        double mean_distance = 0;
18        int count = 0;
19        bool same_configuration = false;
20        std::vector<std::pair<std::string, std::vector<double>>> *data =
21            (state == "brownian") ? &brownian_data
22                                  : &fix_position_data; // update the pointer to the
23                                                        // current data structure
24
25        for (unsigned i = 0; i < (*data)[3].second.size(); i++)
26        {
27            same_configuration =
28                (*data)[3].second[i] == double(parameters.GetMinDeltaFtm()) &&
29                (*data)[4].second[i] == double(parameters.GetBurstPeriod()) &&
30                (*data)[5].second[i] == double(parameters.GetNumberOfBurstsExponent()) &&
31                (*data)[6].second[i] == double(parameters.GetBurstDuration()) &&
32                (*data)[7].second[i] == double(parameters.GetFtmsPerBurst());
33            if (same_configuration)
34            {
35                mean_error += (*data)[8].second[i];
36                mean_distance += (*data)[1].second[i];
37                count++;
38            }
```

```
39          }
40          mean_error = double(mean_error / count);
41          mean_distance = double(mean_distance / count);
42
43          std_deviation = 0;
44          for (unsigned i = 0; i < (*data)[3].second.size(); i++)
45          {
46              same_configuration =
47                  (*data)[3].second[i] == double(parameters.GetMinDeltaFtm()) &&
48                  (*data)[4].second[i] == double(parameters.GetBurstPeriod()) &&
49                  (*data)[5].second[i] == double(parameters.GetNumberOfBurstsExponent()) &&
50                  (*data)[6].second[i] == double(parameters.GetBurstDuration()) &&
51                  (*data)[7].second[i] == double(parameters.GetFtmsPerBurst());
52              if (same_configuration)
53              {
54                  std_deviation += std::pow((*data)[8].second[i] - mean_error, 2);
55              }
56          }
57          std_deviation = std::sqrt(double(std_deviation / count));
58
59          (*statistical_data)[map_key]["count"] = count;
60          (*statistical_data)[map_key]["mean_distance"] = mean_distance;
61          (*statistical_data)[map_key]["mean_error"] = mean_error;
62          (*statistical_data)[map_key]["mean_error_p"] =
63              double(mean_error / mean_distance);
64          (*statistical_data)[map_key]["std_deviation"] = std_deviation;
65          (*statistical_data)[map_key]["std_deviation_p"] =
66              double(std_deviation / mean_distance);
67      }
68
69      else
70      {
71          std::cout << "I red the value from memory!" << std::endl;
72      }
73
74      std_deviation = (*statistical_data)[map_key]["std_deviation_p"];
75  }
```

6.2 Optimal Values

6.2.1 Fixed Position - Stationary

When the nodes are perceived to be in a stationary position, receiving high-frequency updates is not that vital and the resources can be allocated more accurately. As discussed in Chapter 4, the deviation of the error can be reduced by choosing higher values for *ftm per burst* and *burst exponent*. Also, increasing the burst duration's

value has been proven to increase the efficiency while decreasing the channel usage, so higher values compared to the dynamic configuration can be chosen.

As a result, we will define the following configuration as the target or ideal configuration for the model.

- *Ftm per Burst*: 4
- *Burst Exponent*: 4
- *Burst Duration*: 10
- *Burst Period*: 10

The way of transitioning between the ideal configurations will depend on the version of the algorithm.

6.2.2 Brownian - Dynamic

When the relative movement between nodes is considered to be non-stationary it will fall into this *Dynamic* category and will be handled accordingly based on the data gathered during the simulations for the Brownian model. During this phase, a high frequency of measurements is needed as a quick detection of a switch of states is needed. This is based on the assumption that whenever a node goes into Stationary mode, it tends to remain the same for a long time, for instance, the user of the smartphone is sitting down to work, sleeping, etc. whereas, when the node is not stationary we cannot make any of this assumptions. The sessions must remain short time-wise as the longer they take the bigger the error as seen in previous graphs. Due to the exponential relationship of burst exponent with session time, this value must remain low. On the other hand, if the value of burst exponent remains low, the improvement obtained by configuring the ftm protocol with a relatively high value for ftm per burst definitely overcomes the increase of error produced by the increase of session time.

An optimal configuration for the Dynamic mode will be defined as follows:

- *Ftm per Burst*: 4
- *Burst Exponent*: 1
- *Burst Duration*: 7
- *Burst Period*: 7

6.3 Versions

During this section, different variations of the adaptive algorithm will be presented, each version will implement different features. These versions will be compared to some reference static models to compare the individual effects of each one of the features introduced by each version.

- *Version 0.0* → **Fast Static**
  The first one of our reference versions. This version will run with the previously defined dynamic optimal parameters in Section 6.2.2

- *Version 0.1* → **Slow Static**
  The second one of our reference versions. This version will run with previously defined static optimal parameters in Section 6.2.1

- *Version 1.0* → **Static Switch**
  Version 1.0 is our first dynamic configuration. This version implements no smoothness, the configuration alternates between the two optimal configurations in version 6.3 and version-0.1. It also introduces an intermediate state, the state transition diagram can be seen in Figure 6.2.

- *Version 1.1* → **Smooth BE Switch**
  This is a refined version of the 1.0 version. It introduces smoothness for the burst exponent parameter (burst exponent value is increased from 1 to 4 once every two iterations in fixed position state), this way, the effect of an incorrect switch of states is reduced. Once the stationary state is detected, the burst exponent increases the burst exponent once for each two measurements that do not change the state from static to dynamic, the *Burst Exponent* increases until a maximum value of 4.

- *Version 1.2* → **Full Smooth**
  The last version of the algorithm, this version aims to find out whether any difference will be perceived by also introducing smoothness to burst duration and burst period. These two parameters' smoothness works exactly as the burst exponent smoothness in Version 6.3. The maximum value for *Burst Duration* and *Burst Period* is 10.

6.3.1 Simulation Conditions

The conditions provided to the simulation try to emulate a real-case scenario as well as possible, so far, the only mobility model that could get somehow close to a real-scenario situation was the Brownian model. Nevertheless, during this study we will be introducing one last mobility model, the reason for this change

is the seeking of a more representative mobility model. Therefore, the ns3 model *RandomDirection2dMobilityModel*, discussed in Section 3.4.3, will be used for this purpose. This mobility model will be ideal for studying the performance of our algorithm versions as it will combine both situations where our configurations can be considered ideal and the only perceived difference should be relying on the different versions' features.

6.4 Comparation

The error can be reduced overall by increasing the sampling frequency, on the other hand, this perceived increase in accuracy does not come for free, channel time has to be invested to achieve this upgrade. During this chapter, the results gathered for different metrics on the algorithm versions will be analyzed, each one of the versions in Section 6.3 will run for an hour for every stop time, the analyzed stop times are: 20s, 40s, 80s, and 200s. Also, issues such as the perceived error (how much error am I perceiving for how long based on the measurement received) will be addressed in the following sections. Among all the versions, the static versions with high sampling frequency seem to have a satisfactory performance error-wise. Even though the adaptive versions have more extreme error values, they perform most of the time better error-wise and channel usage-wise.

6.5 Error Analysis

6.5.1 Methodology

Even though the error for each measurement can be somehow calculated in the simulation by comparing the expected distance with the obtained distance via RTT, a more precise study of the accuracy of the algorithm requires the introduction of a new metric specially tailored to provide more accurate insight on the error. This new metric will not only consider the difference between perceived and obtained distance but will consider for how long is this difference perceived. This metric will be referred to as *Error Area*.

It can be visually seen as the area between the two curves in Figure 6.3. it is important to note that the distances displayed in $g(x)$ are the ground truth for the measurements, in other words, the middle point between the initial point when starting the session and the final point of the session. As a consequence, g(x) will always behave linearly between 2 consecutive points t_x and t_{x+1}

The longer the sessions, the longer the perceived error will last. On top of that, while the next measurement is received the initiating node may be moving,

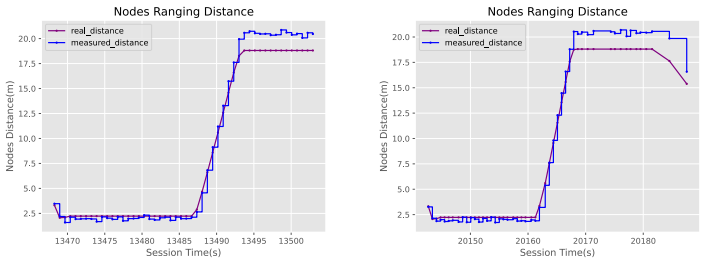

Figure 6.3 – Nodes perceived distance vs current distance for versions 0.0 and 1.1

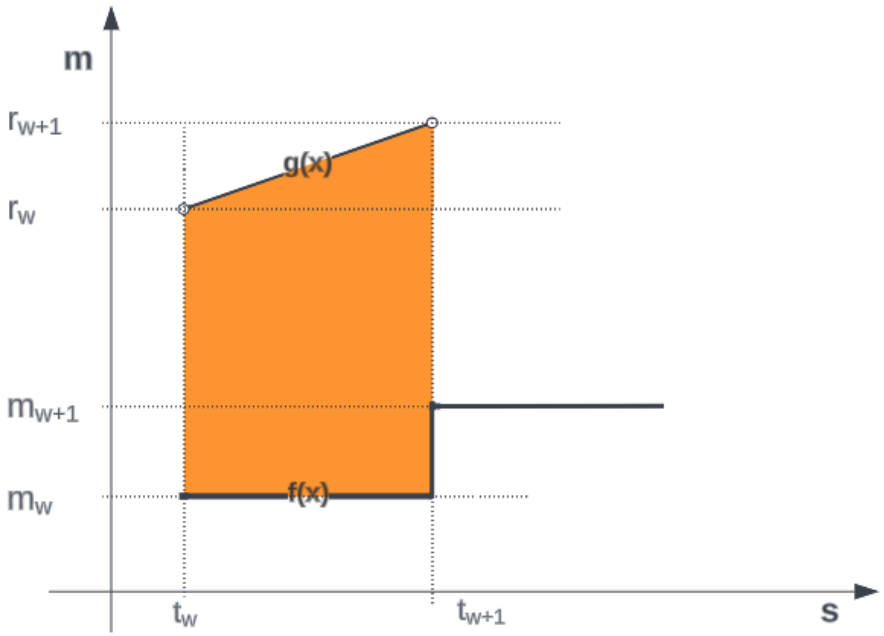

Figure 6.4 – Error area calculation

the perceived error will depend on the state of the initiating node and its relative movement to the receiving node. In this section, we try to study in more depth the perceived error for different versions of the algorithm.

The calculation of the area between the two curves will be performed by finding out the area between both curves with an integral. In Figure 6.4 the area between the two curves is displayed in orange color.

For a simulation of size n measurements, and for every two measurements $w, w+1 : 0 < w < n-1$, knowing that (r_w, t_w) is a point of g(x), the real distance between nodes $g(x)$ can be expressed as:

$$y - r_w = \frac{r_{w+1} - r_w}{t_{w+1} - t_w} \cdot (x - t_w) \tag{6.1}$$

$$g(x) = \frac{\Delta r}{\Delta t} \cdot x + \frac{t_{w+1} r_w - t_w r_{w+1}}{\Delta t} \tag{6.2}$$

Consequently, the measured distance $f(x)$ for a given point (m_w, t_w) during the interval of time $[t_w, t_{w+1})$

$$f(x) = m_w \tag{6.3}$$

The area between the two curves for every pair of consecutive measurements $w, w+1 : 0 < w < n-1$ can be expressed as:

$$\int_{t_w}^{t_{w+1}} g(x) - f(x)\, dx \tag{6.4}$$

6.5.2 Results

The resulting distributions from the mathematical procedure explained above will throw more light on the actual error perceived, in Figure 6.5 a graphical representation of these distributions is displayed. As expected, our adaptive algorithm performance will be placed somewhere between versions 0.0 and 0.1, whose characteristics are explained in section 6.3. In Figure 6.5 a graph containing the perceived error distributions for each version can be seen, the KDE plot version 0.0 with its dynamic-mode parameters reflects that this version is achieving a good performance using a high sampling frequency, the error obtained is only perceived for a brief period and the variations on the state of the node do not have a big impact on the measured error. On the other hand, version 0.1 with its fixed-position-looking parameters is not performing accurately overall, the long waiting time between consecutive measurements makes the results highly inaccurate.

Exciting results are seen for the dynamic versions, in most cases a decent performance is achieved, but for the remaining part of the cases, an increase in accuracy can be seen opposite to the static versions, which is explained by the adaptiveness of our parameter configuration. By including smoothness in other parameters, the distributions seem to be improving, especially from version 1.0 to version 1.1, the upgrade from version 1.1 to 1.2 is harder to judge, but at least it seems like there is a higher frequency of measurements that have an error close to 0. As mentioned before, in case the previous assumption that *when the algorithm's state switches from 'dynamic' to 'static' it tends to stay in that state for a prolonged period* was true, in Figure 6.5 we can see how these adaptive versions suffer an increase in performance for longer wait times in the mobility model.

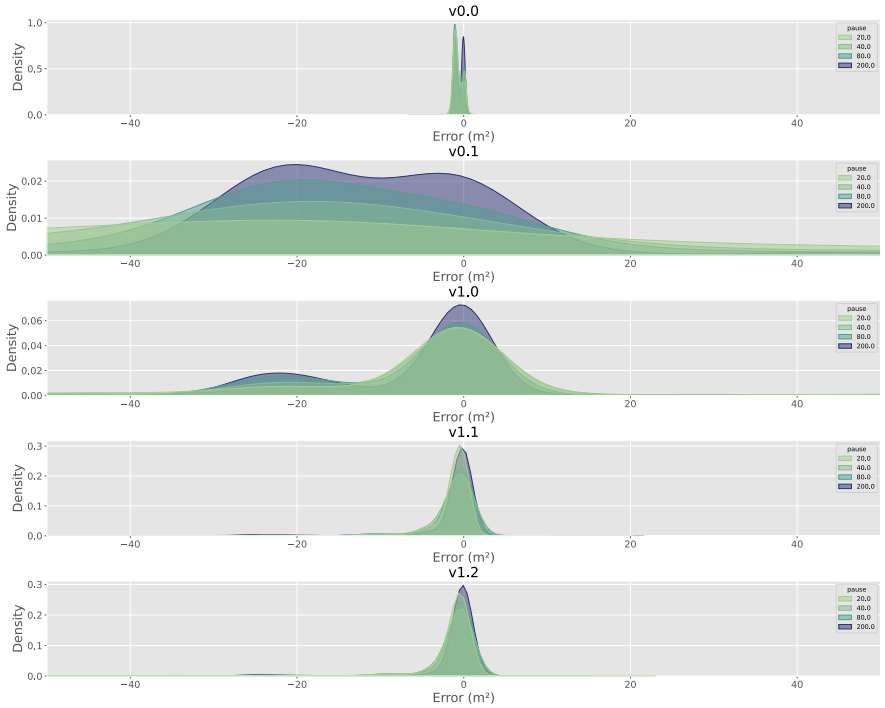

Figure 6.5 – Error perceived for each version (version 0.0 on top, version 1.2 on bottom)

6.6 Channel Time Analysis

6.6.1 Methodology

During the last section, we have been able to analyze the performance of different versions error-wise. Overall, the static version with dynamic parameters (version 0) has proved to be hard to improve error-wise due to the nature of the mobility model and the high frequency of measurements of this configuration. The adaptive versions have improved the accuracy of a percentage of the measurements while introducing more highly deviated outliers, nevertheless, some noticeable differences can be seen between adaptive versions. During this section, the channel usage will be analyzed. One of the priorities of our adaptive algorithms is to be responsible with the Channel Usage finding the balance between the error measured and the resources invested to get that precision. The analysis of the channel time that will follow will be based on the study of the metric 3, by observing the distribution of the values of this metric among different models some conclusions will be drawn and consequently, the chance for a new study between the invested resources and obtained results will be possible.

6.6.2 Results

Looking at Figures 6.6 it can be seen that to achieve the best performance in error (version 0), the utilization of the channel has to be the highest. The reason for this is that when performing short sessions with low parameters, most of the session time is used for transmitting the ftm frames, the sessions are then scheduled one after the other. When the algorithm runs with this specific set of parameters, having a low session time is a priority, consequently, burst duration and burst period will have the lower possible value (but big enough to complete the session). Using this approach, no overhead is introduced by using multiple bursts for one session or by increasing parameters such as burst duration that increase the session time length without an increase of ftm frames transmission.

The adaptive versions presented in this paper use the channel in a far more responsible way than the static versions. As discussed in section 6.5, when analyzing these adaptive versions we see a progressive increase in performance when the static phases of the mobility model get longer. It is no different for the channel usage, in Figure 6.6 versions' 0.0 to 1.2 channel usage are displayed from top to bottom. All the adaptive algorithms experience a considerable upgrade in channel usage as the

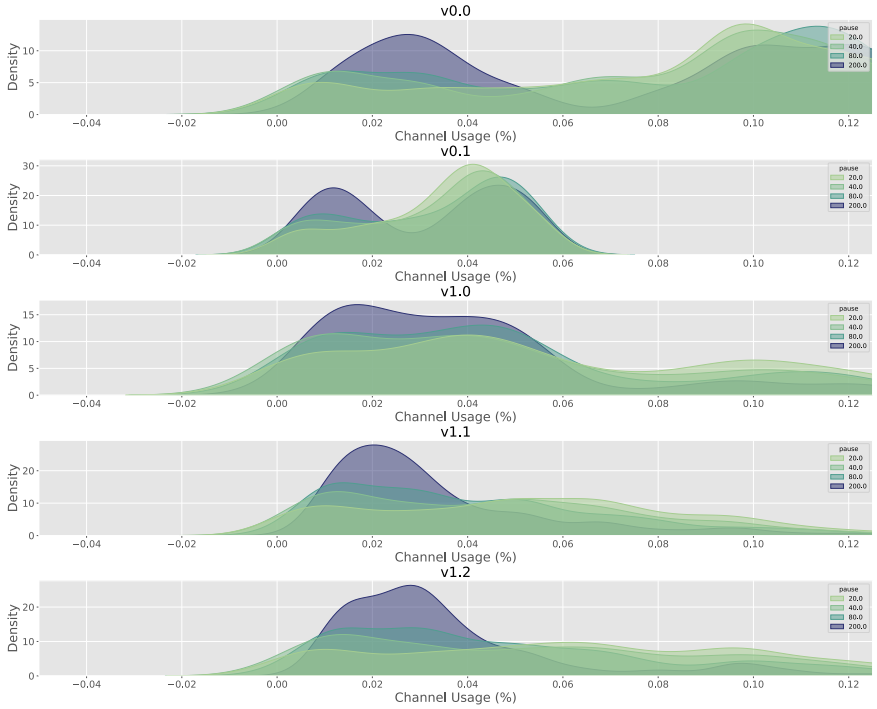

Figure 6.6 – Channel Usage for each version (version 0.0 on top, version 1.2 on bottom)

pauses become longer, which indicates that in real-case indoor location scenarios where devices stay static for long periods while working, sleeping, or watching a movie... these adaptive algorithms could experience a performance boost.

The performance of the adaptive versions illustrates how introducing smoothness for different ftm parameters helps the overall channel time performance of the algorithm and it tends to increase the mean channel time as well as reduce the deviancy. It is relevant to point out that not much of an upgrade was witnessed by introducing smoothness in burst duration and burst period in version 1.2. This is an expected result as the burst duration and burst period introduce overhead to the bursts reducing the channel usage. The progressive increase of these parameters worsens the channel usage of the transition measurements (measurements where the smoothness is still settling down) between states.

Chapter 7

Conclusion

To summarize, this study has explored the effect of different ways of adapting the ftm parameters. The objective was to prove that some balance could be found in between high-accuracy (but also high-frequency sampling) strategies and channel-efficient strategies.

The observed effect of different ftm parameters and different mobility models on the metrics described in Section 4.2 constitute the foundation on which the strategies described in Section 6.3 are based. The effect of smoothness in transition for different parameters was still a mystery, by defining a ground truth of two static versions a comprehensive analysis of the perks and the cons of the smoothness of different ftm parameters was achieved. The results of those studies led to the conclusion that to increase the overall efficiency of the measurements resources such as channel time have to be invested. The nature of the technology and its use cases require that some balance is achieved between high-precision measurements and channel-efficient and non-blocking. The versions presented in this article have proved to fill those requirements, finding a middle ground between both and defining a path for further research in which more complex strategies can be designed and more accurate ways of detecting the mobility mode can be explored. This way, the best traits of each mobility model can be achieved and the smooth versions will not only surpass the static versions in balance but in overall performance.

List of Abbreviations

List of Figures

List of Tables

Bibliography

[1] P. Suresh, J. V. Daniel, V. Parthasarathy, and R. H. Aswathy, "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment," in *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, Chennai, India: American Association for the Advancement of Science (AAAS), Nov. 2014, pp. 1–8. DOI: 10.1126/ICSEMR.2014.7043637.

[2] A. Zubow, C. Laskos, and F. Dressler, "FTM-ns3: WiFi Fine Time Measurements for NS3," in *2022 17th Wireless On-Demand Network Systems and Services Conference (WONS)*, 2022, pp. 1–7. DOI: 10.23919/WONS54113.2022.9764460.

[3] D. Schepers, M. Singh, and A. Ranganathan, "Here, There, and Everywhere: Security Analysis of Wi-Fi Fine Timing Measurement," in *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec '21, Abu Dhabi, United Arab Emirates: Association for Computing Machinery, 2021, pp. 78–89. DOI: 10.1145/3448300.3467828.

[4] V. Barral Vales, O. C. Fernández, T. Domínguez-Bolaño, C. J. Escudero, and J. A. García-Naya, "Fine Time Measurement for the Internet of Things: A Practical Approach Using ESP32," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 305–18 318, 2022. DOI: 10.1109/JIOT.2022.3158701.

[5] S. H. Mohan and R. C. Sofia, "Fine Time Measurement based Time Synchronization for Multi-AP Wireless Industrial Environments," in *2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2023, pp. 399–404. DOI: 10.1109/WiMob58348.2023.10187788.

[6] Y. Yu, R. Chen, L. Chen, S. Xu, W. Li, Y. Wu, and H. Zhou, "Precise 3-D Indoor Localization Based on Wi-Fi FTM and Built-In Sensors," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 753–11 765, 2020. DOI: 10.1109/JIOT.2020.2999626.

[7] C. Yu, W. Xiangming, L. Xinqi, and Z. Wei, "Research on the modulation and coding scheme in LTE TDD wireless network," in *2009 International Conference on Industrial Mechatronics and Automation*, 2009, pp. 468–471. DOI: 10.1109/ICIMA.2009.5156664.

[8] F. Keceli, I. Inan, and E. Ayanoglu, "TCP ACK Congestion Control and Filtering for Fairness Provision in the Uplink of IEEE 802.11 Infrastructure Basic Service Set," in *2007 IEEE International Conference on Communications*, 2007, pp. 4512–4517. DOI: 10.1109/ICC.2007.745.

[9] S. Lindroos, A. Hakkala, and S. Virtanen, "A systematic methodology for continuous WLAN abundance and security analysis," *Computer Networks*, vol. 197, p. 108 359, 2021. DOI: https://doi.org/10.1016/j.comnet.2021.108359.

[10] D. Canedo and A. Romariz, Data Analysis of Wireless Networks Using Classification Techniques, Jul. 2019.

[11] D. Saliba, R. Imad, and S. Houcke, "Wifi channel selection based on load criteria," in *2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2017, pp. 332–336. DOI: 10.1109/WPMC.2017.8301833.

[12] J. A. Adebusola, A. A. Ariyo, O. A. Elisha, A. M. Olubunmi, and O. O. Julius, "An Overview of 5G Technology," in *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, 2020, pp. 1–4. DOI: 10.1109/ICMCECS47690.2020.240853.

[13] Nidhi, A. Mihovska, and R. Prasad, "Overview of 5G New Radio and Carrier Aggregation: 5G and Beyond Networks," in *2020 23rd International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2020, pp. 1–6. DOI: 10.1109/WPMC50192.2020.9309496.

[14] M. Iyer and T. Shanmuganantham, "LNA design for WLAN applications," in *2017 IEEE International Conference on Circuits and Systems (ICCS)*, 2017, pp. 319–323. DOI: 10.1109/ICCS1.2017.8326013.

[15] S.-Y. Yun, Y. Yi, J. Shin, and D. Y. Eun, "Optimal CSMA: A survey," in *2012 IEEE International Conference on Communication Systems (ICCS)*, 2012, pp. 199–204. DOI: 10.1109/ICCS.2012.6406138.

[16] H. Mousavi, I. S. Amiri, M. Mostafavi, and C. Choon, "LTE physical layer: Performance analysis and evaluation," *Applied Computing and Informatics*, vol. 15, no. 1, pp. 34–44, 2019. DOI: https://doi.org/10.1016/j.aci.2017.09.008.

[17] N. Shankar, D. Dash, H. Madi, and G. Gopalakrishnan, "WiGig and IEEE 802.11ad - For multi-gigabyte-per-second WPAN and WLAN," Nov. 2012.

[18] H. Chung and S. Wilson, "Multimode modulation and coding of QAM," *IEEE Transactions on Communications*, vol. 41, no. 1, pp. 1–6, 1993. DOI: 10.1109/26.212356.

[19] W. Webb and L. Hanzo, "Modern Quadrature Amplitude Modulation: Principles and applications for fixed and wireless channels," Jan. 1994.

[20] Y. Guo, S. Zhang, and D. Xiao, "Overview of Wi-Fi Technology," in *Proceedings of the 2012 International Conference on Computer Application and System Modeling (ICCASM 2012)*, Atlantis Press, 2012/08, pp. 1293–1296. DOI: 10.2991/iccasm.2012.330.

[21] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, and B. Walke, "The IEEE 802.11 universe," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, 2010. DOI: 10.1109/MCOM.2010.5394032.

[22] A. Athanasopoulos, E. Topalis, C. Antonopoulos, and P. S. Koubias, "Evaluation Analysis of the Performance of IEEE 802.11b and IEEE 802.11g Standards," May 2006, pp. 141–141. DOI: 10.1109/ICNICONSMCL.2006.92.

[23] D. Vassis, G. Kormentzas, A. Rouskas, and I. Maglogiannis, "The IEEE 802.11g standard for high data rate WLANs," *IEEE Network*, vol. 19, no. 3, pp. 21–26, 2005. DOI: 10.1109/MNET.2005.1453395.

[24] E. Perahia, "IEEE 802.11n Development: History, Process, and Technology," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 48–55, 2008. DOI: 10.1109/MCOM.2008.4557042.

[25] E. H. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtilä, "IEEE 802.11ac: Enhancements for very high throughput WLANs," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011, pp. 849–853. DOI: 10.1109/PIMRC.2011.6140087.

[26] E. Khorov, A. Kiryanov, and A. Lyakhov, "IEEE 802.11ax: How to Build High Efficiency WLANs," in *2015 International Conference on Engineering and Telecommunication (EnT)*, 2015, pp. 14–19. DOI: 10.1109/EnT.2015.23.

[27] J. Christie, R. Fuller, J. Nichols, A. Chen, R. Hayward, K. Gromov, and T. Pfafman, "Development and deployment of GPS wireless devices for E911 and location based services," in *2002 IEEE Position Location and Navigation Symposium (IEEE Cat. No.02CH37284)*, 2002, pp. 60–65. DOI: 10.1109/PLANS.2002.998889.

[28] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019. DOI: 10.1109/COMST.2019.2911558.

[29] W. Wang, X. Liu, M. Li, Z. Wang, and C. Wang, "Optimizing Node Localization in Wireless Sensor Networks Based on Received Signal Strength Indicator," *IEEE Access*, vol. 7, pp. 73 880–73 889, 2019. DOI: 10.1109/ACCESS.2019.2920279.

[30] P. Kumar, L. Reddy, and S. Varma, "Distance measurement and error estimation scheme for RSSI based localization in Wireless Sensor Networks," in *2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN)*, 2009, pp. 1–4. DOI: 10.1109/WCSN.2009.5434802.

[31] K. Heurtefeux and F. Valois, "Is RSSI a Good Choice for Localization in Wireless Sensor Network?" In *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, 2012, pp. 732–739. DOI: 10.1109/AINA.2012.19.

[32] A. Awad, T. Frunzke, and F. Dressler, "Adaptive Distance Estimation and Localization in WSN using RSSI Measures," in *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007)*, 2007, pp. 471–478. DOI: 10.1109/DSD.2007.4341511.

[33] A. T. Parameswaran, M. I. Husain, and S. J. Upadhyaya, "Is RSSI a Reliable Parameter in Sensor Localization Algorithms – An Experimental Study," 2009.

[34] S.-H. Fang, T.-N. Lin, and P.-C. Lin, "Location Fingerprinting In A Decorrelated Space," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 685–691, 2008. DOI: 10.1109/TKDE.2007.190731.

[35] T.-N. Lin and P.-C. Lin, "Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks," in *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 2, 2005, 1569–1574 vol.2. DOI: 10.1109/WIRLES.2005.1549647.

[36] S. Bi, J. Lyu, Z. Ding, and R. Zhang, "Engineering Radio Maps for Wireless Resource Management," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 133–141, 2019. DOI: 10.1109/MWC.2019.1800146.

[37] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *IEEE INFOCOM 2004*, vol. 2, 2004, 1012–1022 vol.2. DOI: 10.1109/INFCOM.2004.1356988.

[38] D. Fan, P. Yu, P. Du, W. Li, and X. Cao, "A Novel Probabilistic Model Based Fingerprint Recognition Algorithm," *Procedia Engineering*, vol. 29, pp. 201–206, 2012, 2012 International Workshop on Information and Electronics Engineering. DOI: https://doi.org/10.1016/j.proeng.2011.12.695.

[39] S. Wielandt and L. D. Strycker, "Indoor Multipath Assisted Angle of Arrival Localization," *Sensors*, vol. 17, no. 11, 2017. DOI: 10.3390/s17112522.

[40] R. Peng and M. L. Sichitiu, "Angle of Arrival Localization for Wireless Sensor Networks," in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 1, 2006, pp. 374–382. DOI: 10.1109/SAHCN.2006.288442.

[41] A. Bensky, "Chapter 14 - Technologies and applications," in *Short-range Wireless Communication (Third Edition)*, A. Bensky, Ed., Third Edition, Newnes, 2019, pp. 387–430. DOI: https://doi.org/10.1016/B978-0-12-815405-2.00014-2.

[42] M. Scherhäufl, M. Pichler, D. Müller, A. Ziroff, and A. Stelzer, "Phase-of-arrival-based localization of passive UHF RFID tags," in *2013 IEEE MTT-S International Microwave Symposium Digest (MTT)*, 2013, pp. 1–3. DOI: 10.1109/MWSYM.2013.6697558.

[43] O. Friedewald and M. Lange, "Localization methods by using phase of arrival in systems with passive RFID Tag's," in *Smart SysTech 2016; European Conference on Smart Objects, Systems and Technologies*, 2016, pp. 1–6.

[44] P. V. Nikitin, R. Martinez, S. Ramamurthy, H. Leland, G. Spiess, and K. V. S. Rao, "Phase based spatial identification of UHF RFID tags," in *2010 IEEE International Conference on RFID (IEEE RFID 2010)*, 2010, pp. 102–109. DOI: 10.1109/RFID.2010.5467253.

[45] S. He and X. Dong, "High-Accuracy Localization Platform Using Asynchronous Time Difference of Arrival Technology," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 7, pp. 1728–1742, 2017. DOI: 10.1109/TIM.2017.2666278.

[46] J. I. Kim, J. G. Lee, and C. G. Park, "A mitigation of line-of-sight by TDOA error modeling in wireless communication system.," in *2008 International Conference on Control, Automation and Systems*, 2008, pp. 1601–1605. DOI: 10.1109/ICCAS.2008.4694487.

[47] F. Liu, H. Chen, L. Zhang, and L. Xie, "Time-Difference-of-Arrival-Based Localization Methods of Underwater Mobile Nodes Using Multiple Surface Beacons," *IEEE Access*, vol. 9, pp. 31 712–31 725, 2021. DOI: 10.1109/ACCESS.2021.3060565.

[48] C. Gentner and T. Jost, "Indoor positioning using time difference of arrival between multipath components," in *International Conference on Indoor Positioning and Indoor Navigation*, 2013, pp. 1–10. DOI: 10.1109/IPIN.2013.6817908.

[49] J. J. Pérez-Solano, S. Ezpeleta, and J. M. Claver, "Indoor localization using time difference of arrival with UWB signals and unsynchronized devices," *Ad Hoc Networks*, vol. 99, p. 102 067, 2020. DOI: https://doi.org/10.1016/j.adhoc.2019.102067.

[50] X. Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila, "Comparison of indoor geolocation methods in DSSS and OFDM wireless LAN systems," in *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No.00CH37152)*, vol. 6, 2000, 3015–3020 vol.6. DOI: 10.1109/VETECF.2000.886867.

[51] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo, "A ranging system with IEEE 802.11 data frames," in *2007 IEEE Radio and Wireless Symposium*, 2007, pp. 133–136. DOI: 10.1109/RWS.2007.351785.

[52] S. A. Golden and S. S. Bateman, "Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1185–1198, 2007. DOI: 10.1109/TMC.2007.1002.

[53] "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, 2021. DOI: 10.1109/IEEESTD.2021.9363693.

[54] "IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016. DOI: 10.1109/IEEESTD.2016.7786995.

[55] S. Lanzisera, D. Zats, and K. S. J. Pister, "Radio Frequency Time-of-Flight Distance Measurement for Low-Cost Wireless Sensor Localization," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 837–845, 2011. DOI: 10.1109/JSEN.2010.2072496.

[56] G. Turin, "An introduction to matched filters," *IRE Transactions on Information Theory*, vol. 6, no. 3, pp. 311–329, 1960. DOI: 10.1109/TIT.1960.1057571.

[57] V. Hinostroza, "Frequency selective fading and frequency correlation analysis on indoor fading radio channels," in *2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No.04CH37514)*, vol. 1, 2004, 166–170 Vol.1. DOI: 10.1109/VETECS.2004.1387935.

[58] Y. Schröder, D. Heidorn, and L. Wolf, "Investigation of Multipath Effects on Phase-based Ranging," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pp. 1–8. DOI: 10.1109/IPIN.2019.8911817.

[59] Q. Spencer, B. Jeffs, M. Jensen, and A. Swindlehurst, "Modeling the statistical time and angle of arrival characteristics of an indoor multipath channel," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 347–360, 2000. DOI: 10.1109/49.840194.

[60] N. Jathe, M. Lütjen, and M. Freitag, "Indoor Positioning in Car Parks by using Wi-Fi Round-Trip-Time to support Finished Vehicle Logistics on Port Terminals," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 857–862, 2019, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019. DOI: https://doi.org/10.1016/j.ifacol.2019.11.237.

[61] G. F. Riley and T. R. Henderson, "The ns-3 Network Simulator," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34. DOI: 10.1007/978-3-642-12331-3_2.

[62] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, and Y. Liu, "A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 891–917, 2017. DOI: 10.1109/COMST.2016.2630047.

[63] Y. Xue, H. S. Lee, M. Yang, P. Kumarawadu, H. H. Ghenniwa, and W. Shen, "Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks," in *2007 Canadian Conference on Electrical and Computer Engineering*, 2007, pp. 1372–1375. DOI: 10.1109/CCECE.2007.345.

[64] Mobility - Model Library.

[65] J. Humpherys, P. Redd, and J. West, "A Fresh Look at the Kalman Filter," *SIAM Review*, vol. 54, no. 4, pp. 801–823, 2012. DOI: 10.1137/100799666. eprint: https://doi.org/10.1137/100799666.

[66] S. Yamaguchi and T. Tanaka, "GPS Standard Positioning using Kalman filter," in *2006 SICE-ICASE International Joint Conference*, 2006, pp. 1351–1354. DOI: 10.1109/SICE.2006.315572.

[67] S. Y. Chen, "Kalman Filter for Robot Vision: A Survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012. DOI: 10.1109/TIE.2011.2162714.

[68] C.-M. Cho and H.-S. Don, "A parallel Kalman algorithm for fast learning of multilayer neural networks," in *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, 1991, 2044–2049 vol.3. DOI: 10.1109/IJCNN.1991.170644.

[69] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," USA, Tech. Rep., 1995.

[70] S. Xu, Y. Wang, and M. Si, "A Two-Step Fusion Method of Wi-Fi FTM for Indoor Positioning," *Sensors*, vol. 22, no. 9, 2022. DOI: 10.3390/s22093593.

[71] H. Cao, Y. Wang, J. Bi, Y. Zhang, G. Yao, Y. Feng, and M. Si, "LOS compensation and trusted NLOS recognition assisted WiFi RTT indoor positioning algorithm," *Expert Systems with Applications*, vol. 243, p. 122 867, 2024. DOI: https://doi.org/10.1016/j.eswa.2023.122867.

[72] H. Chen, J. Yang, Z. Hao, M. Ga, X. Han, X. Zhang, and Z. Chen, "Research on indoor positioning method based on LoRa-improved fingerprint localization algorithm," *Scientific Reports*, vol. 13, Aug. 2023. DOI: 10.1038/s41598-023-41250-x.

[73] J. Yang, S. Deng, M. Lin, and L. Xu, "An Adaptive Calibration Algorithm Based on RSSI and LDPLM for Indoor Ranging and Positioning," *Sensors*, vol. 22, p. 5689, Jul. 2022. DOI: 10.3390/s22155689.

[74] K. Ibwe, S. Pande, A. Abdalla, and G. Mruma, "Indoor positioning using circle expansion-based adaptive trilateration algorithm," *Journal of Electrical Systems and Information Technology*, vol. 2023, p. 10, Feb. 2023. DOI: 10.1186/s43067-023-00075-4.