FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

System Development for Geolocation in Harsh Environments

Joana Enes

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Cândido Duarte (PhD)

Co-supervisor: Vasco Correia (MSc)

July 26, 2018

Abstract

With the emergence of the Internet of Things (IoT), new paradigms have risen for the development of wireless sensors networks (WSN) and machine-to-machine (M2M) communication devices. These are usually consisted of a set of distributed devices, equipped with multiple sensors and actuators, capable of being employed in different environment, of varying characteristics.

Among their basic functionalities, ranging applications have become increasingly popular, mainly due to the mass spread of low cost global navigation satellite systems (GNSS) receivers. However, there still exist particular environments, where propagation of these signals is infeasible. Therefore, in recent years, alternative positioning mechanisms have started to be developed.

Under project ROMOVI, whose main goal is the development of low cost autonomous machinery, for performing logistic and monitoring tasks, alongside the Douro river vineyards, a new alternative for geolocation purposes has been pursued. Recently, a new architecture based on the development of a custom orthogonal frequency division multiplexing (OFDM) physical layer was proposed, to be combined with a suitable ranging algorithm, the symmetrical double-sided two-way ranging (SDS-TWR) technique, for providing relative positioning metrics.

This dissertation seeks to continue that work, by performing a study on the SDS-TWR algorithm and its major error sources. As such, the outcome of this report is dual. First of all, a medium access control layer (MAC) frame format is proposed, taking into account the topology of the network at hand and the requirement for the transmission and reception of geolocation information. In second place, a preamble based on Frank-Zadoff-Chu (FZC) sequences was developed, for time and frequency synchronization purposes, due to its correlation properties.

Finally, the proposed preamble and corresponding reception and auto-correlation function was implemented in hardware description language (HDL) and tested in a field programmable gate array (FPGA). The developed blocks have shown to accomplish the required task of synchronization, while working at the required operation and sampling frequency of 56MHz.

Resumo

Com o aparecimento da internet das coisas, novos paradigmas surgiram para o desenvolvimento das redes de sensores sem fios e comunicações máquina para máquina. Estas redes consistem, usualmente, num conjunto de dispositivos distribuidos no espaço, equipados com diversos sensores e atuadores, capazes de monitorizar ambientes com as mais variadas caraterísticas.

De entre as suas funcionalidades mais básicas, as aplicações de localização tem-se tornado cada vez mais populares, principalmente devido à generalização dos recetores de baixo custo de sinais dos sistemas globais de navegação por satélite. Contudo, em alguns ambientes, devido à sua natureza, a propagação destes sinais não é possível. Deste modo, nos últimos anos têm começado a ser desenvolvidos mecanismos de localização alternativos.

No âmbito do projeto ROMOVI, que tem como principal objetivo o desenvolvimento de maquinaria autónoma, de baixo custo, capazes de executar operações de monitorização e logística nas encostas íngremes das vinhas do rio Douro, uma alternativa para fins de geolocalização tem sido procurada. Recentemente, uma nova arquitetura baseada no desenvolvimento de uma camada fisica que utiliza multiplexação por divisão de frequências ortogonais foi proposta, que, combinada com um algoritmo de localização chamado Symmetric Double Sided Two Way Ranging (SDS-TWR), poderá fornecer medidas de posicionamento relativo.

Esta dissertação vem em continuação desse trabalho, com a realização de um estudo sobre o algoritmo SDS-TWR e as suas principais fontes de erro. Deste modo, este relatório apresenta duas vertentes. Na primeira, uma trama para a camada de acesso ao meio é proposta, tendo em conta a topologia da rede em causa e os requisitos para a transmissão e receção de dados de geolocalização. A segunda vertente está enquadrada na camada física, em que foi desenvolvido um preâmbulo baseado nas frequências de Frank-Zaddoff-Chu, devido às suas boas propriedades de autocorrelação, para fins de sincronização em tempo e frequência.

Por fim, o prêambulo proposto e a sua correspondente receção e cálculo da função de autocorrelação foram implementos em linguagem de descrição de hardware e testados em FPGA. Os blocos desenvolvidos demonstraram efetuar a requirida tarefa de sincronização, operando à frequência de amostragem de 56 MHz.

Agradecimentos

Esta dissertação é o culminar de um percurso académico um tanto ou quanto longo, e que foi marcado por momentos e, certamente, pelas pessoas que me acompanharam ao longo desta jornada.

Em primeiro lugar, agradeço às pessoas que tornaram este documento possível, o meu orientador, professor Cândido Duarte, e o meu coorientador, Vasco Correia, pela proposta de tese, que, não sendo propriamente da minha área de especialização, me deu grandes desafios, mas, ao mesmo tempo me fez aprender e desenvolver competências que, de certo, me tornarão mais completa enquanto profissional no mundo da engenharia. Agradeço também a todos os professores que, de alguma forma, contribuiram para a minha formação, ao longo destes anos.

Apesar de ter passado os últimos meses no núcleo de microelectrónica da FEUP, ao qual agradeço por me ter acolhido, não poderia deixar de mencionar o local que foi a minha segunda casa nos últimos dois anos: o netlab. Às pessoas que lá estiveram para me fazer companhia nos dias de trabalho, em especial, ao Pedro e às suas *darkcookies*, agradeço pelo companheirismo e união, que tornou o netlab, não apenas um local de trabalho, mas também um local onde nos pudemos sentir em família.

Agradeço também ao Orfeão Universitário do Porto, esta instituição centenária, que, apesar da sua longevidade, continua a inovar e a estar sempre "um passo à frente". Obrigada pelos concertos, pelas viagens, pelas aventuras, mas, sobretudo, pela oportunidade de descobrir a cultura portuguesa e de me proporcionar uma lufada de ar fresco nos momentos de maior dificuldade.

Por fim, mas não menos importante, agradeço a todos os que tem feito parte da minha vida. À minha família, aos meus amigos, às pessoas que fui conhecendo ao longo do tempo, todos vocês contribuíram para isto: tal como uma casa é feita de tijolos, cada pedacinho do que sou e do que aqui se apresenta, foi construido por vós: é vosso.

Joana Enes

"Não se me dá que vindimem,
Vinhas que eu já vindimei."

from *Canção da Vindima*
(Popular Portuguese Song)

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

ACK	Acknowledgment
ALM	Adaptive Logic Module
AODV	Ad-hoc On-demand Distance Vector
AP	Access Point
AWGN	Additive White Gaussian Noise
BM-SDS-TWR	Burst Mode Symmetric Double Sided Two Way Ranging
BSS	Basic Service Set
CDMA	Code Division Multiple Access
CFO	Carrier Frequency Offset
CIC	Cascaded Integrator-Comb
CSMA/CA	Carrier Sence Multiple Access with Collision Avoidance
CTS	Clear-to-Send
DBTMA	Dual Busy Tone Multiple Access
DCF	Distributed Coordination Function
DIFS	DCF Interframe Space
DS	Distribution System
DSP	Digital Signal Processing
EDCA	Enhanced Distributed Channel Access
FDMA	Frequency Division Multiple Access
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FZC	Frank-Zaddoff-Chu
GCC	Gnu Compiler Collection
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IBSS	Infrastructured BSS
IEEE	Institute of Electrical and Electronics Engineers
HDL	Hardware Description Language
HWMP	Hybrid Wireless Mesh Protocol
MAC	Medium Access Control
MACA	Multiple Access with Collision Avoidance
MBSS	Mesh BSS
MCS	Modulation and Coding Scheme
M2M	Machine to Machine
NAV	Network Allocation Vector
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open System Interconnection

OWR	One-Way Ranging
ppm	*parts per million*
QoS	Quality of Service
REQ	Request
RF	Radio Frequency
RM-AODV	Radio Metric Ad-hoc On-demand Distance Vector
ROM	Read-Only Memory
RREP	Route Reply
RREQ	Route Request
RToF	Roundtrip Time of Flight
RTS	Request-to-Send
SDS-TWR	Symmetric Double Sided Two-Way Ranging
SDS-TWR-MA	Symmetric Double Sided Two-Way Ranging with Multiple Acknowledgment
SFO	Sampling Frequency Offset
SIFS	Short Interframe Space
SNR	Signal-to-Noise Ratio
STA	Station
TDMA	Time Division Multiple Access
ToF	Time of Flight
TTL	Time To Live
TWR	Two-Way Ranging
TXOP	Transmission Opportunity
UART	Universal Asynchronous Receiver/Transmitter
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network

Chapter 1

Introduction

Wireless sensor networks (WSN) consist of spatially distributed, mostly autonomous, devices equipped with several sensors capable of offering remote environmental monitoring. Nowadays, one of its basic requirements is their localization, inside a network. In certain environments with harsh communication conditions, such as the slopes of the Douro vineyards, where global navigation satellite systems (GNSS), like the global positioning system (GPS), cannot provide accurate measurements of the nodes position, new positioning techniques based on radio frequency (RF) transceivers are being attempted.

One possible approach is to calculate the distance between two nodes by measuring a RF signal's time-of-flight (ToF) and multiplying it by the speed of light. Typically, there are two major techniques based on this principle: one-way ranging (OWR) and two-way ranging (TWR). Since OWR requires very precise clock synchronization in the network, a TWR technique is preferred. By increasing the number of nodes and, consequently, the number of packets exchanged, an increase in accuracy is expected. Furthermore, when communication between three or more nodes is feasible, the system allows to perform trilateration and obtain a two-dimensional or three-dimensional estimation of a node's relative position.

1.1 Problem statement

The ROMOVI (Modular and Cooperative Robot for Vineyards) project has as its main goal the development of a low cost and versatile solution for hillside vineyards. An autonomous robot with monitoring and logistic capabilities will allow to mitigate the challenges of the steep slope viticulture, like the sharp inclination of the land, difficult accesses or the high need for manpower, leading, as such, to aiding the wine producers and reducing the related high production costs [1,2].

The lack of reliable communications, due to natural obstacles, makes it impracticable the use of GPS/GNSS signals for precisely locating the robots. As such, a full-custom wireless communication system dedicated to relative positioning purposes is required to enable secure and precise trajectory planning. The purpose of this dissertation is to develop a solution that will enable the

possibility of having robots operating in this kind of environments, aiming to provide better commercial solutions in a sector with high economic potential.

1.2 Proposed Solution

The main objective of this dissertation is the proposal of a medium access control layer (MAC) layer architecture for geolocation purposes, to combine with Symmetric Double Sided Two-Way Ranging (SDS-TWR) algorithm (fig. 1.1). The proposed MAC layer has to meet the general requirements of M2M communications, such as having low overhead, providing mechanisms to avoid and handle possible collisions and being scalable, and the specific requirements for geolocation. This includes implementation on the provided hardware, while being energy efficient. In order to provide a solution for the present problem, it will have to be able to handle two different types of frames: ranging and data frames.

This requires the study of network topologies, such as mesh networks and the mechanisms that meet their requirements, like network formation, multi-hop communication and medium access control. Furthermore, a study on positioning algorithms was performed, with a focus on SDS-TWR, as well as the major error sources which affect its accuracy, such as time or phase offsets, and possible ways to mitigate these effects. In order to employ SDS-TWR algorithm, a new type of frames with geolocation functions is proposed.

Since the transmitter and receiver are not synchronized, mechanisms are required for allowing frame synchronization. In this work, a preamble based on Frank-Zadoff-Chu sequences was studied and developed in hardware description language (HDL), allowing synchronization at the reception of a frame. This mechanism will also be used for providing measurements necessary for SDS-TWR algorithm.

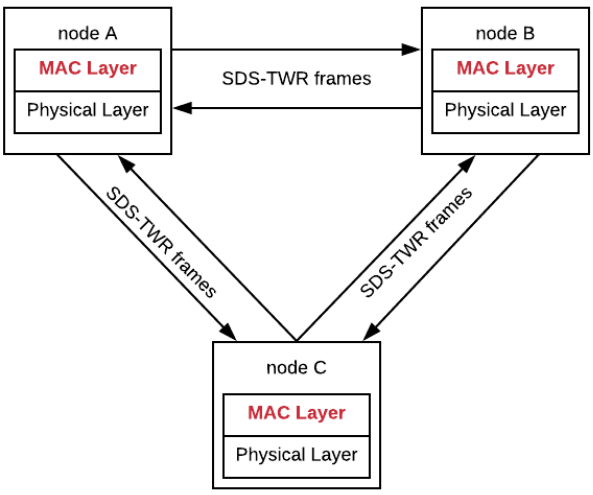


Figure 1.1: Basic analysis of the proposed solution.

1.3 Document Structure

This document is structured as follows:

- **Chapter 2:** an overview of positioning techniques is presented, with a focus on roundtrip time of flight (RToF) and SDS-TWR and its major error sources;
- **Chapter 3:** it is given an overview on mesh networking and MAC layer mechanism for these;
- **Chapter 4:** an architecture is proposed, based on mesh topologies, and a frame format is presented; SDS-TWR and a preamble format are demonstrated through MATLAB^{TM} simulation;
- **Chapter 5:** the implementation of the preamble in hardware is described, along with its verification;
- **Chapter 6:** the conclusions of time work are presented and future work discussed.

Chapter 2

Fundamentals of positioning techniques

In this chapter, an overview of positioning techniques and the main algorithms related to this subject is presented. It is given a focus on RToF algorithms and the SDS-TWR technique and the major error sources present in geolocation techniques are described. In the last section, waveform correlation is briefly presented as a way to determine the arrival of a signal.

2.1 Positioning Techniques

Wireless systems may take advantage of several different positioning techniques, in order to provide location estimation. These are divided in three major categories ([3, 4]):

- **Proximity**: based on sensing techniques, which determine when an object is near a known location. Typically, it requires a dense grid of antennas or anchors, each having a known position and a limited range. Relative location information is given when at least one of the nodes detects the object.

- **Scene Analysis**: uses features (fingerprints) of a scene, collected *a priori*, which describe and identify a specific location. By identifying similarities with observed scene characteristics it is possible to infer an object location.

- **Triangulation**: uses the geometric properties of triangles to estimate the location of an object. It is divided into two subcategories: angulation (angle measurements) and lateration (distance measurements). It requires measurements from multiple reference positions.

For more information on this topic, the author suggests reading [5], where a comparison of some positioning methods is presented, along with some technologies that use them.

This report will be focused mostly on a lateration technique, roundtrip time of flight, which will be presented in the following section.

2.2 Roundtrip Time of Flight

Roundtrip time of flight is a lateration technique based on measuring the time a signal takes to travel from one device to another and back, as seen in fig. 2.1.

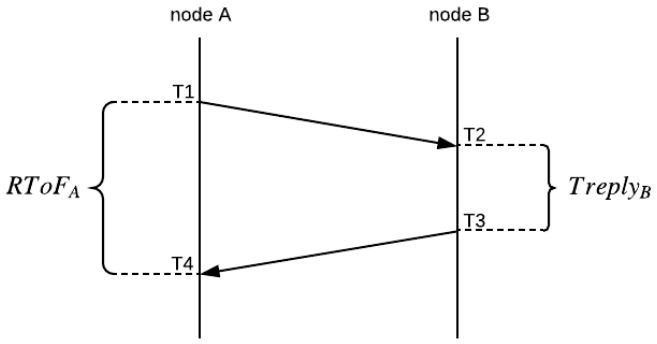


Figure 2.1: RToF algorithm.

Having a device A sending a signal to device B, at instant T1, and receiving an acknowledgment, at instant T4, the time of flight, which is the time the signal takes to travel from A to B, on the device A (ToF_A) perspective is calculated according to (2.1) [6].

$$ToF_A = \frac{T4 - T1 - (T3 - T2)}{2} \tag{2.1}$$

Knowing the propagation velocity (v_p) of the signal medium (speed of light for electromagnetic waves), it is easily inferred the distance (d) between the two devices (2.2). If at least three measures of different reference points are taken, a bi-dimensional position can be determined.

$$d = ToF \cdot v_p \tag{2.2}$$

The major advantage of RToF relatively to OWR algorithms is that it does not require a precise clock synchronization between the two nodes. The devices use timestamps taken by themselves to calculate the propagation time of the signal, so a more moderate clock synchronization is permitted [4]. However, the estimation of the reply time on the opposite device may lead to additional errors.

2.3 Symmetric Double Sided Two Way Ranging

Symmetric double sided two way ranging is an RToF based mechanism which allows nodes on both sides of the link to estimate their relative distances, in the same iteration of the algorithm.

As seen in fig. 2.2, node A starts by sending a ranging request packet (REQ), to which node B responds with an acknowledgment and a new ranging request packet (ACK + REQ), allowing node A to estimate its round-trip time (*TroundA*). Node A then sends an acknowledgment (ACK), which allows node B to apply the same procedure on its side. Considering *Treply* the delay between the

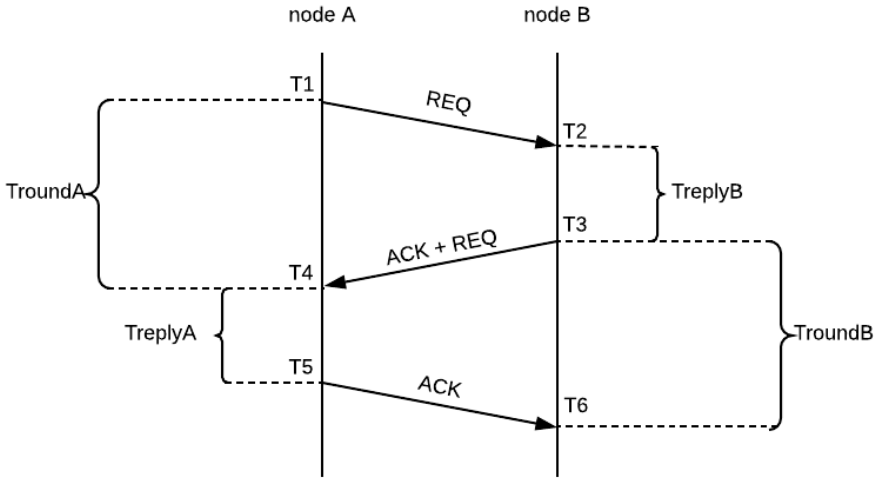


Figure 2.2: SDS-TWR algorithm.

reception of the REQ packet and the transmission of the ACK packet and assuming $TreplyB \simeq TreplyA$, the time of flight between the two nodes, estimated by A, is given by [6]:

$$ToF_A = \frac{1}{2} \cdot (TroundA - TreplyB) \tag{2.3}$$

2.4 SDS-TWR related works

In this section, works which employ the aforementioned RToF ranging technique will be presented. Furthermore, it will be given an analysis of Burst-Mode SDS-TWR (BM-SDS-TWR) algorithm and a comparison between this and other TWR techniques in terms of performance and power consumption will be made.

2.4.1 Burst-mode SDS-TWR

In [7], a new ranging methodology based on SDS-TWR is presented, which claims to mitigate the effect of clock drift and clock resolution by performing consecutive SDS-TWR iterations (see fig. 2.3).

According to this algorithm, if N iterations of SDS-TWR are performed, the error between the estimated time of flight (ToF_{est}) and the real time of flight (ToF) is given by (2.4):

$$ToF_{est} - ToF = \frac{1}{4N} \sum_{i=1}^{N} (2(e_A + e_B)ToF + (e_A - e_B)(T_{reply_{B_i}} - T_{reply_{A_i}})) + \Delta R \tag{2.4}$$

where $e_{A/B}$ represents the clock drift of the device *A/B* and $T_{reply_{A/B_i}}$ is the reply delay of the device *A/B* at iteration i. ΔR is the accomulation of errors resulting from clock resolution and can be

calculated according to (2.5):

$$\Delta R = \frac{1}{4N}\sum_{i=1}^{N}(((\delta_{A_{n+i}} - \delta_{A_i}) - (\delta_{A_{2n+i}} - \delta_{A_{n+i}})) + ((\delta_{B_{n+i}} - \delta_{B_i}) - (\delta_{B_{2n+i}} - \delta_{B_{n+i}}))) \tag{2.5}$$

where δ_i represents the delay between the instant the event i happens and the moment the respective timestamp is taken.

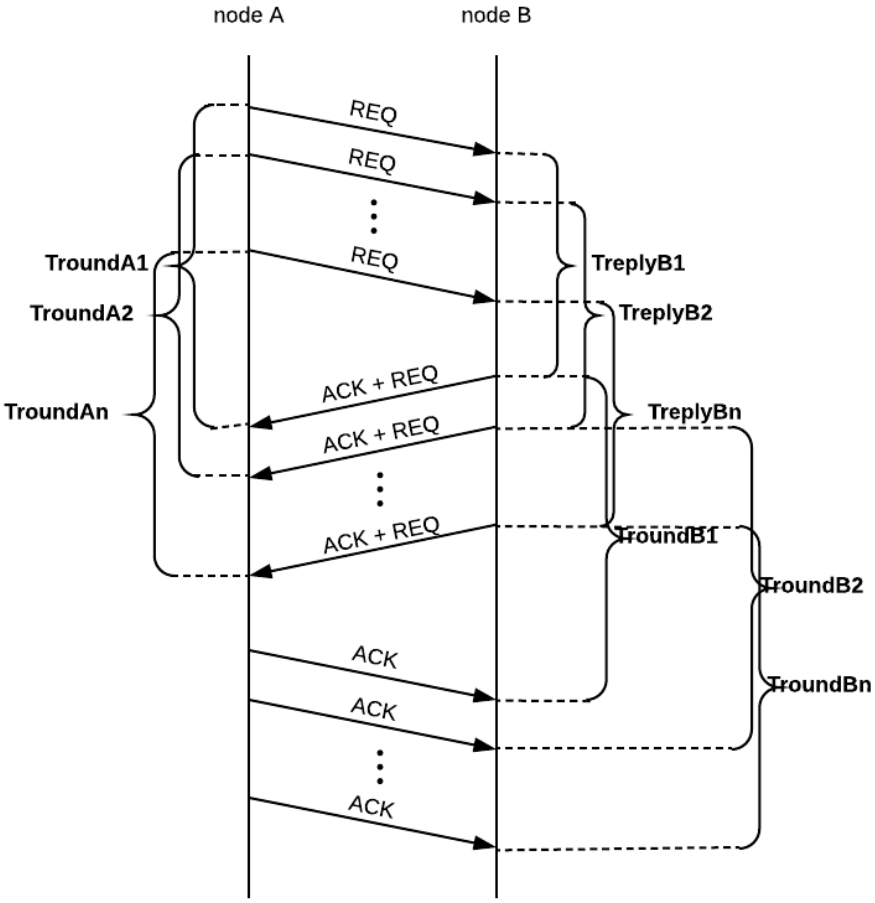


Figure 2.3: Burst mode SDS-TWR algorithm.

Assuming

$$T_{reply_A} = T_{reply_B} >> ToF \tag{2.6}$$

then we can make the following approximation:

$$ToF_{est} - ToF \approx \frac{1}{4N}\sum_{i=1}^{N}((e_A - e_B)(T_{reply_{B_i}} - T_{reply_{A_i}})) + \Delta R \tag{2.7}$$

By analyzing (2.5) and (2.7) it is possible to conclude that both ΔR and the error in estimation decrease as the number of exchanged messages increases. However, this technique increases the

total ranging time (time needed for two nodes to perform the ranging algorithm), which leads to an increase of collision probability of radio signals, higher power consumption in mobile nodes and reduction in the number of range requests each anchor node can handle.

In [8], burst-mode SDS-TWR is tested against other four TWR based methods. Measurements were made in both indoor and outdoor (up to 25 m) environments. The results show that in open field, although the differences between algorithms were minimal, burst-mode SDS-TWR had the best results in terms of accuracy (smallest mean error between estimated and real distances) and precision (smallest standard deviation). However, when the metric is the ratio of number of successful ranging attempts over total ranging attempts, burst-mode SDS-TWR has the worst results, which indicates that this is the least robust method. It is also the slowest method, due to the higher number of messages exchanged during the process. A summary of the results can be found in table 2.1.

Table 2.1: Comparison between TWR algorithms [8].

	TWR	**SDS-TWR**	**BM6-SDS-TWR**
Run Time r_t (ms)	0.9	2.5	17
Outdoor			
Mean Error μ (m)	0.00	0.00	0.00
Max Error μ_∞ (m)	4.85	4.78	2.73
Standard Deviation σ (m)	0.09	0.07	0.07
95 Percentile p_{95} (m)	0.08	0.07	0.06
Robustness η (%)	99.8	99.8	99.2
Indoor			
Mean Error μ (m)	0.05	0.05	0.06
Max Error μ_∞ (m)	4.74	2.36	0.48
Standard Deviation σ (m)	0.07	0.06	0.06
95 Percentile p_{95} (m)	0.14	0.14	0.13
Robustness η (%)	95.0	96.3	89.0

2.4.2 SDS-TWR with Multiple Acknowledgment

SDS-TWR with Multiple Acknowledgment (SDS-TWR-MA) is a method, proposed in [9], in order to reduce ranging time, while also decreasing clock drift. In this algorithm, node B answers with several acknowledgment packets to a single ranging request. Considering n acknowledgment packets and e_A and e_B the clock drifts, the ranging error, $ToF_{est} - ToF$, is given by:

$$ToF_{est} - ToF = \frac{1}{4n}(e_A - e_B)\sum_{i=1}^{n}(t_{replyB_i} - t_{replyA_i}) \tag{2.8}$$

It was concluded that, compared to SDS-TWR executing n iterations (in order to obtain the same number of measurements as SDS-TWR-MA), this algorithm can reduce the ranging packets by 33% (when $n = 1$ or more), with the difference getting even higher as n increases. Therefore, the major advantage of this method is that it enables less power consumption in the mobile nodes while

keeping the ranging procedure accurate. In table 2.2, it is presented a comparison of power consumption between BM-SDS-TWR and SDS-TWR-MA performed with k iterations and n times. It is assumed that mobile nodes remain in Tx mode while they transmit the REQ and ACK frames and in Rx mode while they wait for the ACK+REQ messages. As for the anchors, as they do not know when they will be contacted by the mobile, they must remain in Rx mode until then. P_{Tx} and P_{Rx} represents the consumed power associated to each mode and T_d and T_a represent the duration of the transmission of each REQ and ACK frame, respectively.

Table 2.2: Energy consumption comparison between BM-SDS-TWR and SDS-TWR-MA [10].

Mobile Consumption	
SDS-TWR-MA	$n(P_{Tx}(2T_a+T_d)+(k+1).P_{Rx}.T_d)$
BM-SDS-TWR	$k.n(P_{Tx}(T_d+T_a)+P_{Rx}T_d)$
Anchor Consumption	
SDS-TWR-MA	$n(T_d.P_{Tx}(k+1)+P_{Rx}(T_d+2T_a+\frac{(n-1)((k+2)T_d+2T_a)}{2})$
BM-SDS-TWR	$n.k(P_{Tx}.T_d+P_{Rx}(T_d+T_a+\frac{(2T_d+T_a)(n-1)}{2}))$

2.4.3 SDS-TWR multi way ranging

In order to provide a 2-D localization, a mobile node requires measurements relative to more than one fixed position. In [11], an adaptation of SDS-TWR is proposed, called *Sequential SDS-TWR*, and it is based on having a mobile node performing SDS-TWR requests sequentially to several anchors. Then, it combines the relative distances with the *Ring Localization Algorithm (iRingLA)*, obtaining the estimated position. This algorithm was tested in an indoor environment with a prototype and a simulator. 3 anchors were positioned in two different scenarios ($6.89 \times 8.28m^2$ and $6.68 \times 5.16m^2$ rooms), where 60 samples were collected for each of the positions occupied by the mobile node. The results show a mean localization error around 1 m and less than $2m$, with a 90% percentile.

2.5 Most common geolocation error sources

Geolocation techniques might be affected by several errors, leading to not so accurate measurements and compromising the performance of systems. This section presents the most common errors sources that affect geolocation algorithms, such as digital domain clock drifts, carrier and sampling frequency offset and sampling time offset.

2.5.1 Digital domain clock drifts

In digital systems, timing is usually provided by a local clock driven by crystal oscillators. These type of oscillators rely on the mechanical resonance of a vibrating quartz element to create a periodic electrical signal and can have independent frequency variations caused by physical

factors, such as temperature, manufacturing limitations or aging. This effect leads to unwanted clock drifts, which cause a decrease in ranging accuracy [12].

In TWR algorithms, the effect of clock drift can be seen in the estimation of *Treply*. If there is a delay of N clock cycles between the request and the response, *Treply* will correspond to (2.9), where f_{clk} is the frequency of the clock. If the clock frequency varies in some *parts per million* (*ppm*), the effective delay time seen by the other devices will increase or decrease, according to the variation in frequency.

$$Treply = N \cdot \frac{1}{f_{clk}} \tag{2.9}$$

If we represent clock drift as δ and τ as the time of reply under ideal conditions, the estimated reply time under clock drift effect ($Treply_{est}$) is given by [13]:

$$Treply_{est} = \frac{\tau}{1+\delta} \tag{2.10}$$

The error in distance estimation derives from the fact that each node estimates the delay of the other nodes according to their own delay. Because clock drifts are independent from each other, this will introduce errors in estimation.

2.5.2 Carrier frequency offset

Carrier frequency offset (CFO) is an effect related to the difference between transmitter and receiver carrier frequencies [14]. These are generated by frequency synthesizers using precise crystal oscillators, whose precision is given in *ppm*. Because the receiver's frequency might have a small error in relation to the transmission frequency, a frequency error might be introduced in the process of down-conversion of the received signal to baseband, as illustrated in fig. 2.4 [15]. Assuming Δ_{CFO} as the difference between the transmitted carrier frequency and the one generated on the receiver, after conversion the baseband signal will be phase shifted by $\Delta_{CFO}t$, i.e., a cumulative phase shift with time. This requires the system to have dedicated hardware, in order to compensate the phase shift.

2.5.3 Sampling frequency offset

Sampling frequency offset (SFO) results, once again, from an error in frequency generation. It is defined as the difference in sampling frequency on the receiver relatively to the sampling frequency on the transmitter, as described in (2.11), where T_s and F_s correspond to the sampling period and frequency, respectively, on the transmitter and δ represents the normalized relative sampling clock offset, which is equal to $\Delta T/T_s$, usually specified in *ppm*.

$$SFO = \frac{1}{T_s} - \frac{1}{T_s + \Delta T} \approx F_s \delta \tag{2.11}$$

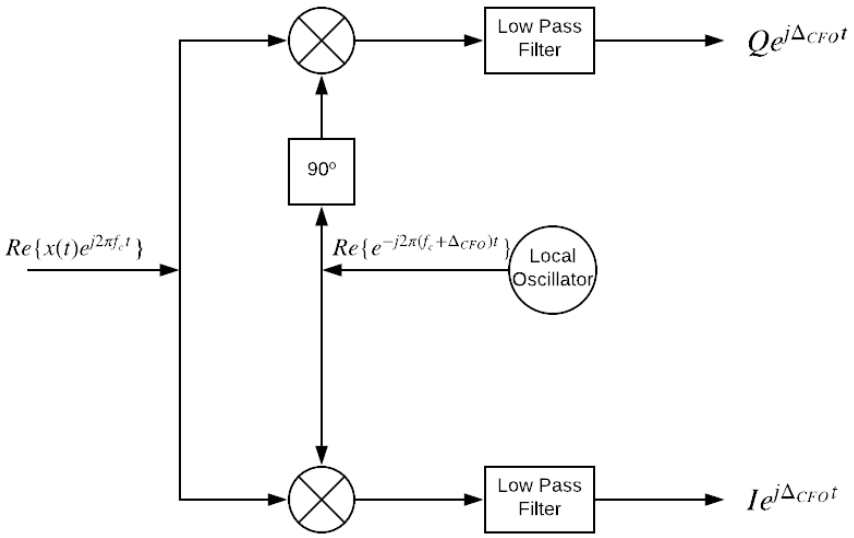


Figure 2.4: Down conversion on reception with CFO.

Sampling frequency offset affects the precision of the sampling clock. Under this effect, the signal $x(t)$ will be sampled, on the receiver, at a different frequency that it was transmitted, according to (2.12), which causes a degradation of the signal [14].

$$x[nT_s] \longrightarrow x[n(T_s + \Delta T)] \tag{2.12}$$

2.5.4 Sampling time offset

Sampling time offset occurs when the sampling clocks of the transmitter and the receiver are not synchronized. The result is that the signal is not sampled at the same instants on the receiver as it was on the transmitter, which leads to a misalignment of the received samples, as seen in fig.2.5.

This effect can be written as:

$$x[nT_s] \longrightarrow x(nT_s + \Delta t) \tag{2.13}$$

This means the transmitted samples are not exactly the same as the ones seen at the received signal. The resulting signal can be seen as the original one, but with a phase shift offset:

$$x(t) = Ae^{j2\pi f_o t} \longrightarrow x(t) = Ae^{j2\pi(f_o t + \Delta f)} \tag{2.14}$$

2.6 Waveform correlation for time-estimation techniques

Time estimation techniques usually rely on matched filters or energy detector receivers [16]. By determining the instant at which the maximum of the correlation at the output of the matched

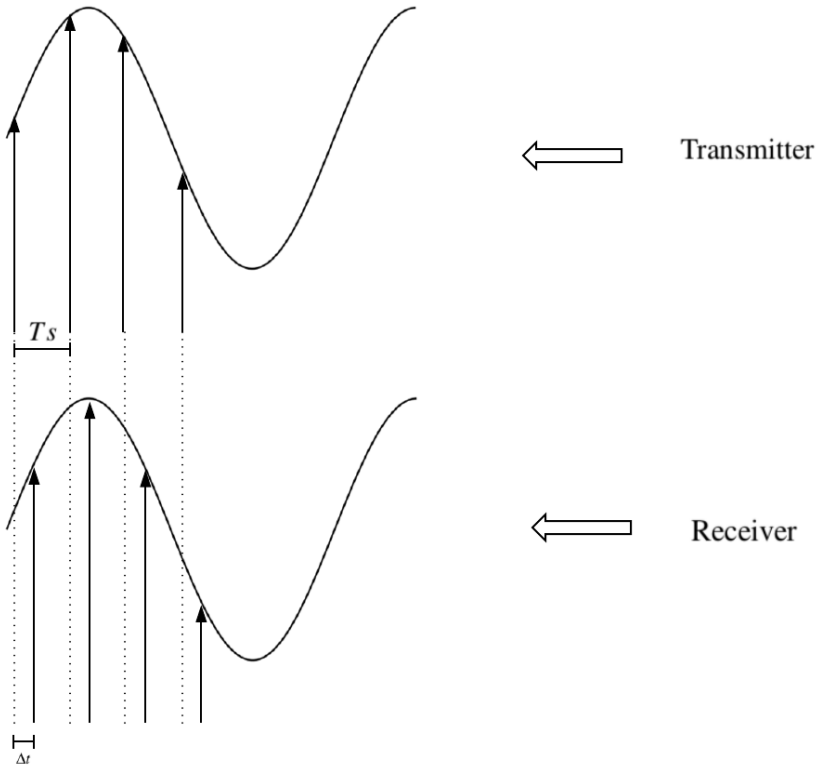


Figure 2.5: Sampling time offset on the reception.

filter or the maximum energy sample occurs, it is possible to identify when the transmitted signal arrived (time of arrival).

A common way to determine this time instant, if the transmitted signal is known, is to calculate the cross-correlation of the received signal ($x_r[n]$) with a reference one ($x_t[n]$) with N_s samples, according to (2.15):

$$corr(x_t[k], x_r[k]) = \sum_{i=0}^{Ns-1} x_t[k] \cdot x_r^*[k-i] \tag{2.15}$$

where $corr(x_t[k], x_r[k])$ is the cross-correlation between x_t and x_r at instant k and $*$ denotes the conjugated symbol. However, most of the times, the receiver doesn't have *a priori* information about the sent signal [17]. An alternative is to calculate the correlation of the received waveform with a delayed copy of itself, by Ns samples, as seen in fig. 2.6.

The result of this procedure is a peak delayed by Ns samples, which matches the delay introduced on the received waveform. It is possible to observe an example of this procedure in fig. 2.7: the correlation is calculated between a sinusoid and the same sinusoid delayed by 200 samples (fig. 2.7a).The result obtained can be seen in fig. 2.7b. The peak of the autocorrelation is located on sample 200, as expected.

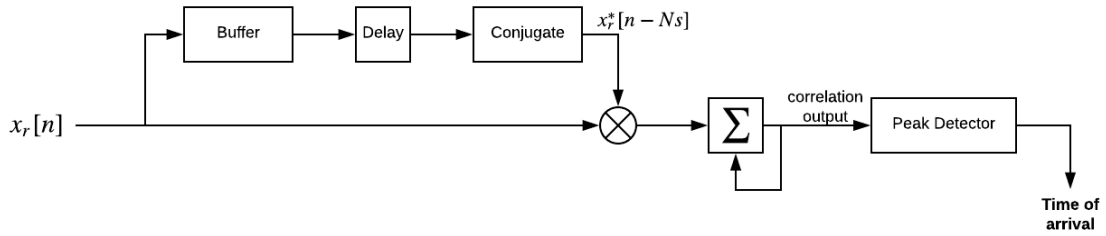
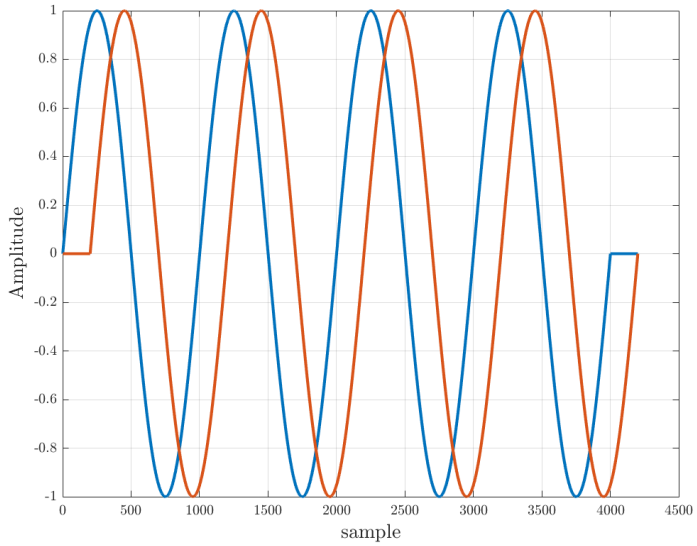
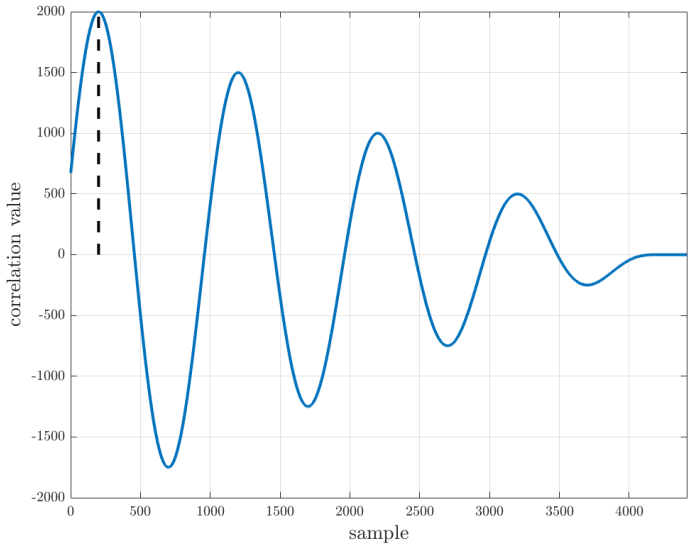


Figure 2.6: Autocorrelation of $x(t)$ with a delayed copy of itself and time of arrival determination.



(a)



(b)

Figure 2.7: Sinusoidal signal and its 200 samples delayed version (a) and corresponding autocorrelation function (b).

Chapter 3

Network Architecture

In order to enable communication among multiple nodes, it is required the definition of a network architecture. Therefore, a research on wireless mesh networks (WMN) was made, with a focus on the MAC layer and Institute of Electrical and Electronics Engineers (IEEE) standards for MAC. In this chapter, it is given an overview of mesh networking and its topologies. Then, a focus on the MAC layer with a presentation and comparison of several MAC mechanisms for M2M communication is given. Finally, it is presented the wireless network standard IEEE 802.11 and some of its MAC mechanisms, and its amendment for mesh networking, IEEE 802.11s.

3.1 Wireless Mesh Networks

Wireless Mesh Networks consist on dynamically self-organized and self-configured decentralized networks, where nodes automatically establish and maintain connectivity among themselves (forming what is known as an ad-hoc network) [18]. This brings many advantages to WMNs, such as low costs, easy network maintenance, robustness and reliable service coverage [19].

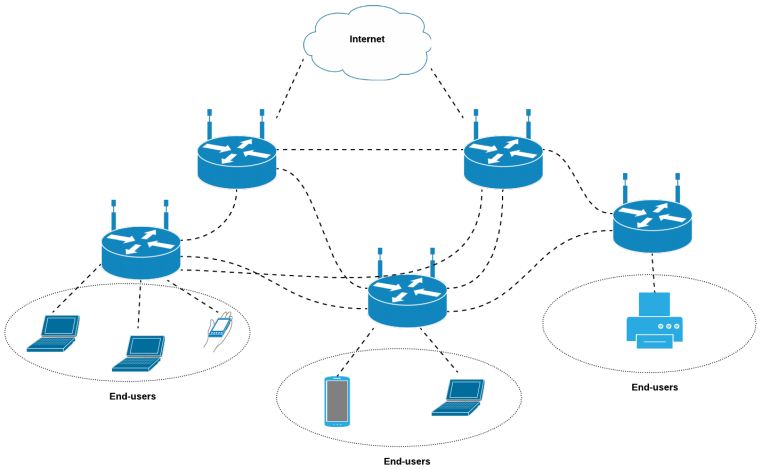
WMNs consist of two types of nodes: mesh routers and mesh clients. Although mesh routers are usually built on similar hardware platforms as conventional wireless routers, they might contain additional routing functions to support mesh networks, such as multiple wireless interfaces and enhanced MAC protocols for better scalability. As for mesh clients, they should have, at least, the necessary functions for mesh networking and be able to work as routers in WMNs, except for gateway and bridge functions.

3.1.1 WMN Architecture

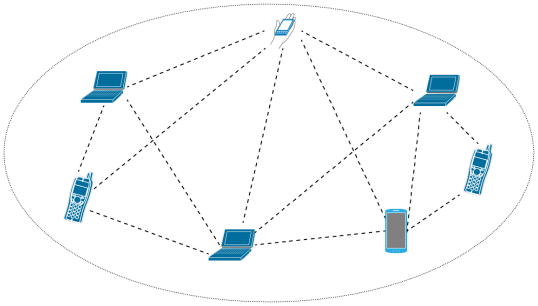
Wireless Mesh Networks architectures can be classified into 3 types:

- **Infrastructure/Backbone WMNs:** characterized by mesh routers forming an infrastructure of self-configuring, self-healing links among themselves, providing a gateway for conventional clients (see fig. 3.1a).

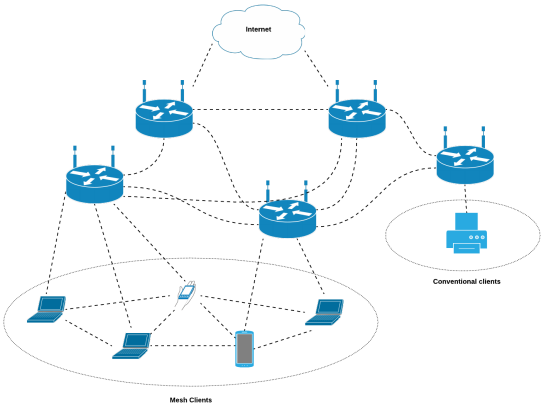
- **Client WMNs:** Constituted by client nodes connected among each other, forming a peer-to-peer network, performing routing and self-configuration functions while providing end-user applications (see fig. 3.1b).
- **Hybrid WMNs:** combines infrastructure and client meshing features. While mesh routers provide an infrastructure and a gateway to other networks, mesh clients can either access the network through the routers, or communicate directly with other mesh clients (see fig. 3.1c).



(a) Infrastructure/Backbone WMN.



(b) Client WMN.



(c) Hybrid WMN.

Figure 3.1: Wireless Mesh Networks architectures.

3.2 Medium Access Control Layer

The medium access control layer is responsible for coordinating access by nodes to a physical shared medium. When a MAC protocol is designed for wireless mesh networks, there are certain aspects to consider, such as:

- **Multi-hop communication:** in WMN, frames can go through multiple hops and, due to problems like the hidden node (which will be explained later), MAC needs to have routing capabilities;
- **Distributed and cooperative communication:** as there is no centralized controller, it is MAC's function to ensure that every nodes cooperate in transmission;
- **Network self-organization:** network management must be a part of the MAC protocol. It performs functions like network formation and nodes association/disassociation. It is also important for multi-hop communication;
- **Mobility of nodes:** In WMN, nodes are often mobile, which may change the network topology. Thus, it is important that nodes exchange network topology information which must be considered for routing update.

These requirements added some specific functions to be performed at the MAC layer, in comparison with general wireless MAC protocols. Therefore, new MAC mechanisms were proposed.

3.2.1 MAC mechanisms for M2M communication

In order to support all the M2M communications requirements, MAC protocols should provide high throughput and reliability, be scalable, energy efficient, have low latency and use low cost hardware [20].

Considering the mechanisms used in order to handle collisions, MAC protocols can be classified into three different categories:

- **Contention-free protocols:** resources are preallocated exclusively for each node:
  - Time division multiple access (TDMA)
  - Frequency division multiple access (FDMA)
  - Code division multiple access (CDMA)
- **Contention-based protocols:** nodes compete for channel access, requiring mechanisms to reduce and recover from collisions:
  - ALOHA
  - slotted-ALOHA
  - Carrier sense multiple access with collision avoidance (CSMA/CA)
  - Multiple access with collision avoidance (MACA)
  - Dual busy tone multiple access (DBTMA)
  - ...
- **Hybrid protocols:** combine features of contention-free and contention-based protocols. Usually switch between random access at low loads and scheduled access at high loads:

- Hybrid MAC (HyMAC)
- Z-MAC [21]
- Multimode hybrid MAC (MH-MAC) [22]
- ...

In table 3.1 is presented a summary of the advantages and disadvantages of each of the categories.

Table 3.1: Advantages and disadvantages of MAC protocols categories.

Category	Advantages	Disadvantages
Contention-free	• Better channel utilization at high loads	• Lower efficiency at low loads • Limited flexibility and scalability • Strict hardware requirements • High overhead and energy consumption
Contention-based	• High flexibility • Better performance at low loads • Simple setup and implementation	• Lack of scalability • Higher number of collisions at high loads
Hybrid	• Good performance at both low and high loads	• Poor scalability • High overhead

As seen, general wireless protocols don't fit well into M2M communication. Therefore, several protocols made specifically for M2M communication have been proposed in recent years. A summary of the performance of some of these protocols, in terms of the requirements presented above, is given on table 3.2.

Table 3.2: Comparison between several specific MAC protocols for M2M communications [20].

Protocol	Throughput	Scalability	Energy efficiency	Latency	Cost	Burst handling
DPCF-M	Moderate	Moderate	Moderate	Moderate	Low	Yes
CSMA-TDMA hybrid	Moderate	Moderate	Moderate	Low	Low	No
Contention-FDMA hybrid	Moderate	Moderate	Low	High	Low	Yes
ATL-SMACA	Low	Low	Low	High	Low	No
CERA	High	Moderate	Moderate	Low	High	Yes
IEEE 802.11ah	High	High	Moderate	Moderate	Low	No
FASA	Low	Low	Low	High	Low	No
M2M LTE [23]	Moderate	Low	Moderate	Moderate	High	No
M2M LTE [24]	High	Moderate	Moderate	Moderate	High	Yes
Cognitive	High	Moderate	Moderate	Moderate	High	Yes
Cognitive polling	Moderate	Low	High	High	High	No

3.2.2 IEEE 802.11

IEEE 802.11 is a standard for wireless networks specified in [25], which specifies a medium access control layer and several specifications for the physical layer. The purpose of IEEE 802.11 is to describe an architecture for wireless local area networks (WLAN) and to define norms for communication between stations within this networks.

This research focused on MAC mechanisms standardized in IEEE 802.11, some of which will be presented next: carrier sensing, with the use of the request-to-send(RTS)/clear-to-send(CTS) algorithm along with network allocation vector (NAV); and medium access control, provided by the carrier sense multiple access with collision avoidance (CSMA/CA) protocol.

3.2.2.1 Carrier Sensing

Carrier sensing means to determine if the medium is available. There are two types of carrier sensing: physical carrier sensing and virtual carrier sensing. Physical carrier sensing is provided by the physical layer and depends on the medium and modulation used. However, it has some disadvantages, like the cost to build physical carrier sensing hardware for RF-based media and problems like the hidden node [26]. A hidden node scenario can be observed in fig. 3.2, where node C is a hidden node to node A, and vice-versa. If no access control mechanism is used, since node A and node C are not aware of the existence of each other, they might transmit simultaneously to node B, which results in frame collisions and loss of information.

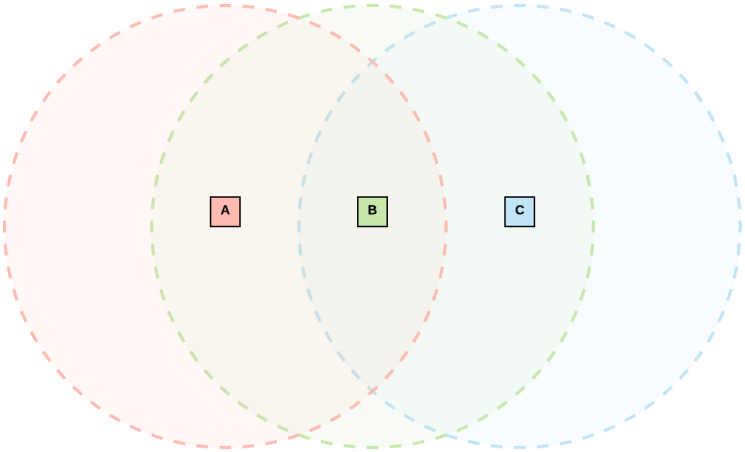


Figure 3.2: Hidden node scenario.

In order to mitigate this problem, virtual carrier is proposed in IEEE 802.11, with the optional RTS/CTS handshake and the NAV. RTS/CTS is a mechanism which aims to provide control over the shared medium and reduce frame collisions caused by the hidden node problem. It is based on the exchange of short packets preceding the actual data transmission. It begins with the transmitter sending a RTS packet, at which the receiver replies with a CTS packet. At the reception of CTS, the transmitter knows the medium is available and starts transmitting the data packet. To provide reliability, after receiving the data, the receiver sends an acknowledgment (ACK) packet.

Both RTS and CTS packets contain the duration of data transmission, which is used by the other nodes who hear some of the packets that they should back off from accessing the medium while the transmission occurs. This duration field can be used by nodes to set the NAV. The NAV is a timer used by the virtual carrier sensing function to indicate if the medium is busy or idle. When receiving a RTS or a CTS packet, NAV is set to a certain duration. If NAV is higher then zero, virtual carrier sensing function indicates medium is busy, and the NAV starts to being decremented. When reaching zero, the virtual carrier sensing function indicates medium is idle.This process can be seen on fig. 3.3 [27]. Furthermore, we can find that nodes have to wait a certain amount of time between frames. This is called interframe spacing, and in high priority transmissions, like RTS/CTS, it is used the short interframe space (SIFS). The other time depicted is known as distributed coordination function (DCF) interframe space (DIFS), which is defined as the minimum medium idle time before a node is allowed to transmit. This means that when the NAV reaches zero and the medium is considered idle, every node has to wait at least one DIFS before starting transmitting.

3.2.2.2 Carrier Sense Multiple Access With Collision Avoidance

CSMA/CA is a medium access protocol used in IEEE 802.11, which can be used alongside the aforementioned RTS/CTS mechanism. It is based on the idea that nodes should wait random periods of time before transmitting. According to CSMA/CA, when a node has a frame to transmit,

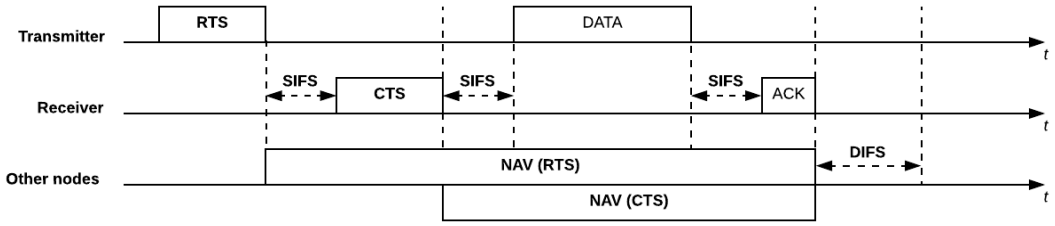


Figure 3.3: RTS/CTS mechanism and NAV.

it should sense the medium. If the medium is idle for at least one DIFS, the transmission can start immediately; if not, the node should wait for the medium to be idle for one DIFS and then set a random backoff timer, which is decremented for each time slot the medium is idle. If the medium gets busy before the timer reaches zero, the timer is frozen until the next instant the medium remains idle for one DIFS. When the timer reaches zero, the node may start to transmit [28]. An example illustration of this protocol can be found in fig. 3.4, where there are 3 nodes which want to transmit data. The access to the medium is coordinated by CSMA/CA.

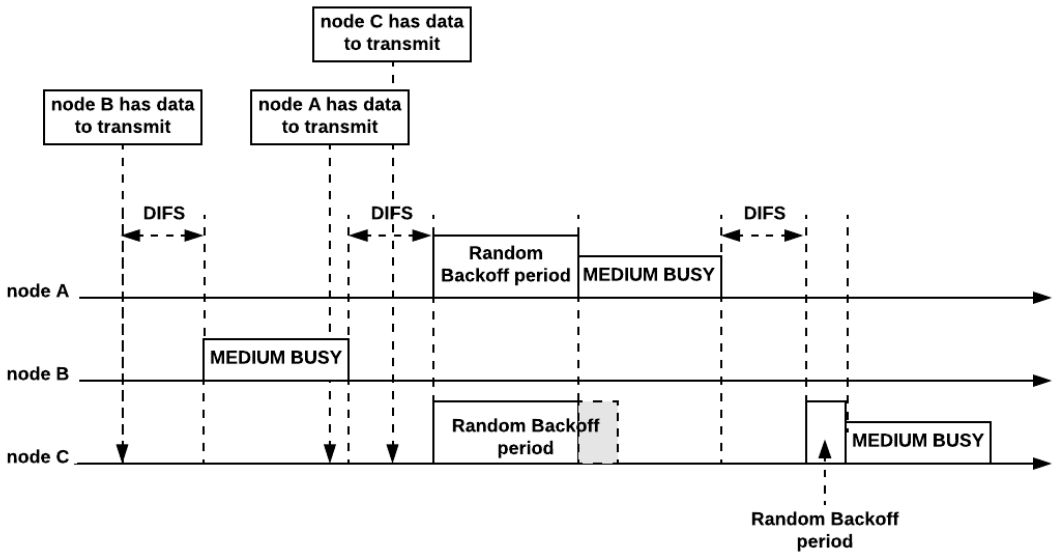


Figure 3.4: CSMA/CA protocol.

3.2.3 IEEE 802.11s

IEEE 802.11s appeared as an attempt to develop a standard for wireless mesh networking. It is an extension to the IEEE 802.11 MAC that provides routing capabilities at the MAC layer. IEEE 802.11 standard was initially developed for wireless networks interconnected through wired links, with routing functions being performed in the third layer, the network layer, of the OSI model. However, since wireless links are not as reliable as wired links, in wireless mesh networks,

multihop communication should be provided by a routing protocol which considers the nature of these links. IEEE 802.11 does not provide specifications about the interfaces, the network layer needs to obtain link metrics (criteria used to characterize the performance, quality and eligibility of a link [29]) from the MAC layer [30]. Therefore, 802.11s emerged as a solution for multihop communication in wireless mesh networks.

3.2.3.1 Network structure

802.11 most basic entity is a station. Two stations connected form the most basic network, called basic service set (BSS). A station which provides the association of other stations to the network is called an access point (AP). A BSS where an AP is present is called an infrastructured BSS (IBSS). BSS are interconnected by a distribution system (DS). In 802.11s, a wireless LAN formed by autonomous stations establishing peer-to-peer connections and exchanging messages is called a mesh BSS (MBSS).

Mesh stations, which belong to a MBSS, can perform mesh functionalities, provided by MAC enhancements, such as the formation of the MBSS, path selection and forwarding. Mesh stations cannot communicate directly with non-mesh stations. However, MBSS can access the DS through mesh gates, thus, allowing MBSS to communicate with conventional 802.11 networks. There is also the possibility to integrate 802.11 networks with non-802.11, like Ethernet (802.3), using a logical component called portal [29], which runs a routing protocol at the network layer, needed for path selection from the mesh network to the outside, and vice-versa (see fig. 3.5).

3.2.3.2 Mesh formation

Mesh stations can use both passive scanning, which is the observation of beacon frames transmitted by other mesh stations, or active scanning, which is the transmission of probe frames. Both beacon and probe frames contain a mesh ID (the identifier of the mesh network) and parameters supported by the mesh stations which transmitted the frame. This helps mesh stations to find a network which uses the same parameters as them, like the path selection protocol and metric. Once they find a matching peer, the association starts.

In order to establish a connection with a peer, mesh stations use the mesh peer link management protocol, which is used to establish, maintain and close mesh peering between two stations, while supporting security mechanisms. When the security flag is disabled, a mesh peer link is established after [29]:

1. both mesh stations have sent and received a mesh peering open frame;

2. both mesh stations have sent and received a corresponding mesh peering confirm frame.

After a mesh link is established, the airtime cost (c_a) for this link is calculated, which is used as a metric for path selection. By default, c_a is calculated by the expression given in 3.1 [31]. It considers the channel access overhead O_{ca}, which depend on the modulation and coding scheme

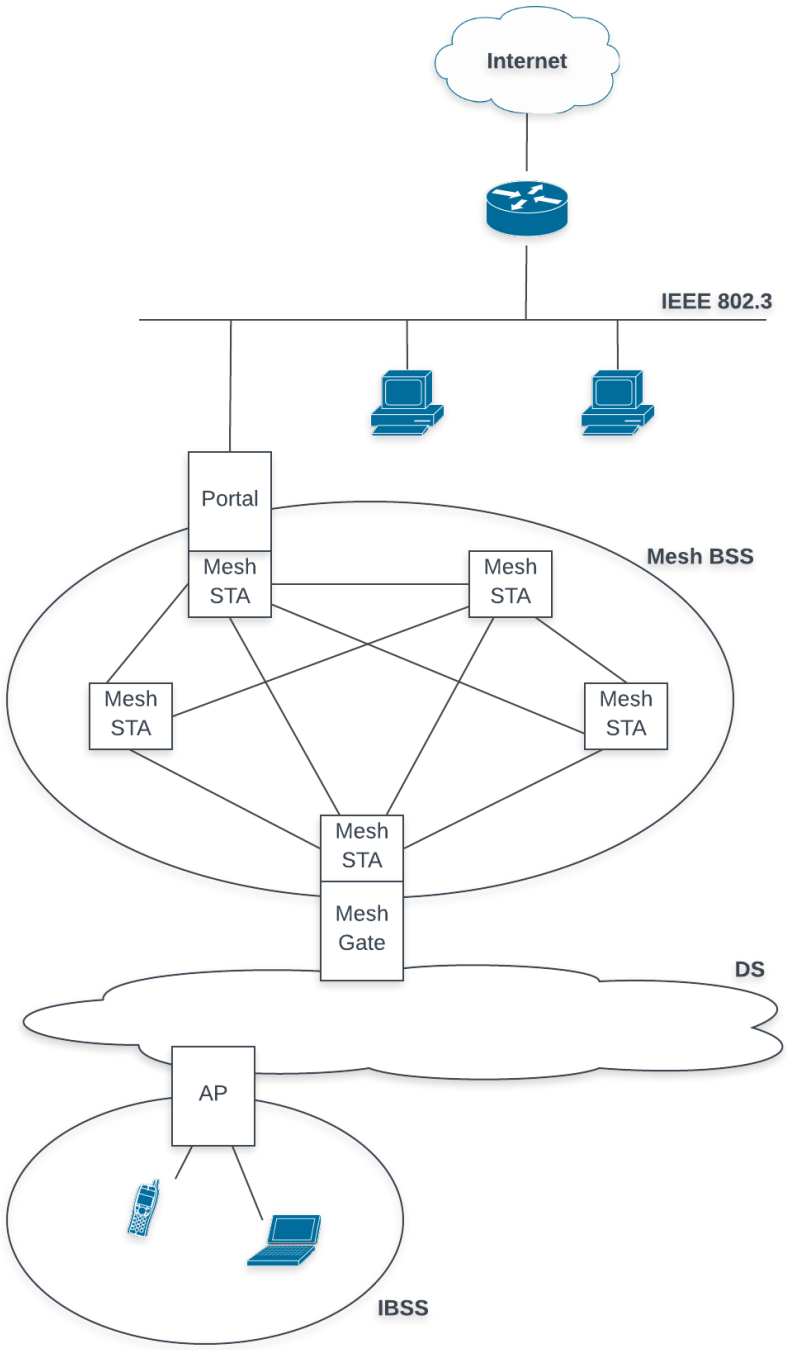


Figure 3.5: 802.11s network architecture.

(MCS), the protocol overhead (O_p), the number of bits in a test frame (B_t), the MCS bit rate (r) and the frame error rate (e_f) for the test frame.

$$c_a = (O_{ca} + O_p + \frac{B_t}{r}) \times \frac{1}{1 - e_f} \tag{3.1}$$

3.2.3.3 Medium Access Control.

In 802.11s, medium access control relies on enhanced distributed channel access (EDCA), which is a contention-based protocol specified in 802.11e [32]. EDCA provides categorization of traffic, with different priority levels, thus providing quality of service (QoS). Mesh stations with higher priority levels gain access to the medium by obtaining a transmission opportunity (TXOP). During the TXOP, the station may transmit multiples frames. The reception of each frame is confirmed by an ACK frame sent by the receiver [33].

3.2.3.4 Path Selection

In a mesh network, the path selection protocol and metric has to be common to all mesh stations. 802.11s uses, by default, the hybrid wireless mesh protocol (HWMP) and airtime metric (which was presented in 3.2.3.2). HWMP combines two modes: on-demand routing and proactive tree-based routing.

On-demand routing is based on radio-metric ad hoc on-demand distance vector (RM-AODV), which is a version of ad hoc on-demand distance vector (AODV) [34] that works at the data link layer, by using a radio aware counting metric. AODV is a reactive protocol, which means that it only searches for a path when it is needed to send data between two mesh stations. Despite having a higher latency at the sending of the first frame, it has the advantages of always using the most recent link state information and introducing reduced routing overhead at low traffic, since there are no periodic route maintenance messages. This makes it suitable for networks with highly mobile nodes and low traffic loads [35].

An example of AODV route discovery in an ad hoc network can be seen in fig. 3.6: when a mesh station has a message to deliver, it first checks the routing table whether there is a path to the destination. If not, it sends its neighbors a route request message (RREQ). If a neighbor has a path to the destination or is itself the destination, it will reply with a route reply message (RREP). If not, it will also send a RREQ message to its neighbors. Thereby, the request for a route spreads among the network, until reaching nodes which have a path to the destination or are the destination themselves.

In order to use the proactive tree-based routing, it is required a mesh station to be configured as a root mesh station. This station has the function to periodically broadcast routing messages to either establish or maintain routes to all mesh stations in the network. This creates a tree topology with the root mesh station as the root of the tree. This mechanism uses the same distance vector methodology as RM-AODV and also reuses its routing control messages.

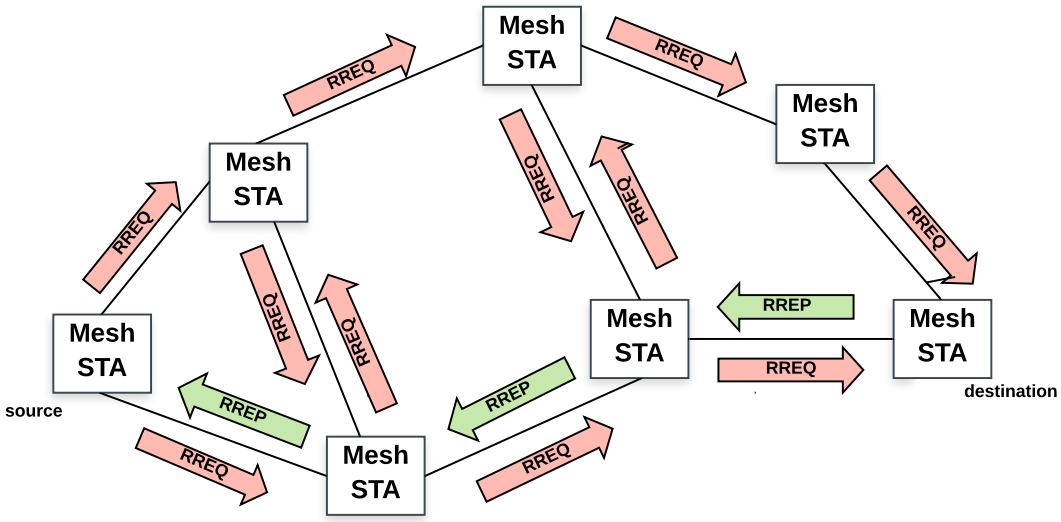


Figure 3.6: AODV route discovery.

Chapter 4

Proposed architecture

This chapter is divided in two sections. On the first, a network architecture is proposed, based on the research made. As is can be seen on fig. 4.1, it is proposed a network topology, considering that the major function of the system is to enable node positioning. According to the chosen topology, some MAC mechanisms are proposed, based on IEEE wireless networks standards. Considering these and SDS-TWR algorithm, a frame format was defined.

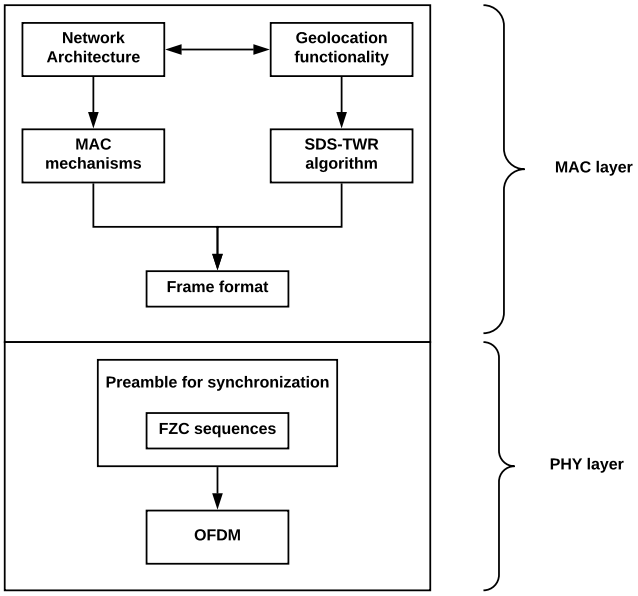


Figure 4.1: Proposed architecture.

In the second section of this chapter, a focus on the geolocation functionality of the system is given. First, it is presented a study of SDS-TWR algorithm and how the errors mentioned in chapter 2 can affect this measures. Finally, it is presented a technique for nodes synchronization, at the reception of a frame, through the introduction of a preamble based on Frank-Zaddoff-Chu (FZC) sequences. A study on these sequences was made, as well as the detection of the preamble for different link conditions.

4.1 Network architecture

The geolocation system to be developed has to be employed in a self-configured decentralized wireless network. Therefore, a preliminary architecture is proposed, based on a client mesh network, where all nodes can form peer-to-peer links among each other, making them able to perform the SDS-TWR algorithm among their peers. The network formation and link establishment should follow the approach presented on IEEE 802.11s standard, employing mesh peer link management protocol to manage link peering between nodes.

For the medium access control, since no QoS is required for now (there is no need for traffic prioritization or resource reservation), a simple contention-based protocol shall be used. It is proposed the use of CSMA/CA, as it has been a reliable protocol, often used in wireless networks and standardized by IEEE 802.11. Because nodes ranging and topology of the land are propitious to originating hidden nodes, this protocol should be used alongside RTS/CTS mechanism.

For now, it was not defined a path selection protocol, which would be necessary for multihop communication. Since the main focus of this dissertation is the geolocation functionality, it doesn't make sense to use multihop, as nodes only perform SDS-TWR with their neighbors. However, for data transmission purposes, a multihop protocol should be considered in future developments.

4.1.1 Frame format definition

Considering the proposed architecture, a preliminary MAC layer frame format was defined. This frame format is based on the one presented in IEEE 802.11s, with the addition of geolocation functions. The general frame format is presented in fig. 4.2.

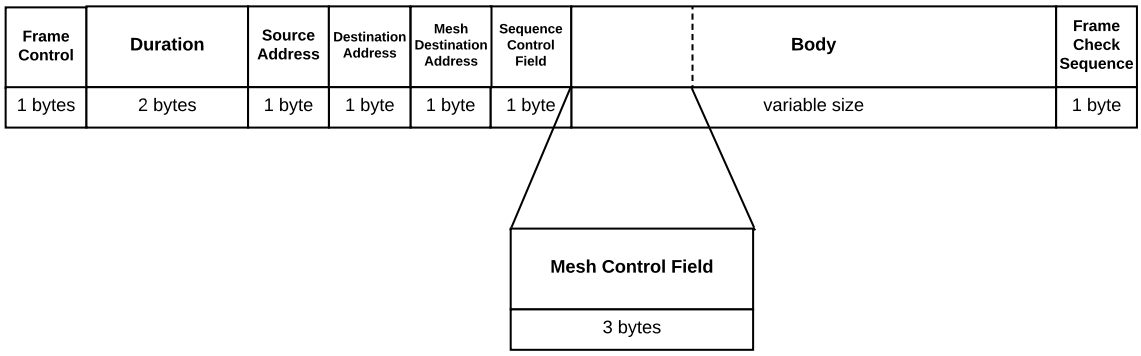


Figure 4.2: Generic Frame Format.

4.1.1.1 Frame control field

Frame control (see fig. 4.3) has a total length of 1 byte and contains the following fields:

- **Type + subtype** (2 + 4 bits): indicate the function of the frame. There are four types of frames:
  - management: frames dedicated to link establishment and maintenance;

  - control: frames used by the RTS/CTS mechanism;
  - geolocation: frames necessary to perform the SDS-TWR algorithm (the introduced type);
  - data;

- **Retry field** (1 bit): indicates if the frame is a retransmission of a previous frame. Helps in the detection of duplicated frames;
- **Mesh control field present** (1 bit): indicates if the mesh control field is present in the frame;

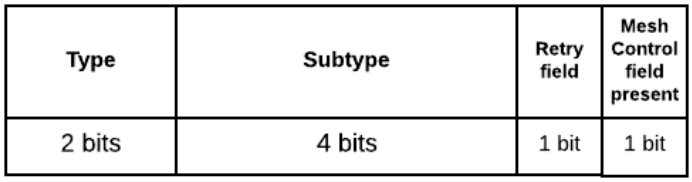


Figure 4.3: Frame control field.

The types and some subtypes have already been defined are can be found in table 4.1.

Table 4.1: Frame Control.

Frame Control			
Bits 0-1	**Type**	**Bits 2-5**	**SubType**
00	Management	0000	Association Request
		0001	Association Response
		0010	Reassociation Request
		0011	Reassociation Response
		0100	Probe Request
		0101	Probe Response
		1000	Beacon
01	Control	0000	RTS
		0001	CTS
		0010	ACK
10	Geolocation	0000	REQ
		0001	ACK + REQ
		0010	ACK
11	Data	0000	NULL Data
		0001	Data

4.1.1.2 Duration Field

The duration field has a length of 16 bits, which represents an unsigned integer between 0 and 65535. If contains the duration in microseconds that the medium is expected to remain busy due to an ongoing transmission. This field is monitored by all stations that receive the frame in order to update the NAV.

4.1.1.3 Address fields

There are three address fields:

- **Source address**: represents the node address that is the source of transmission.
- **Destination address**: represents the node address that is the final recipient of the frame.
- **Mesh destination address**: in an eventual multihop communication for data transmission, this field indicates the next hop of the transmission, which is the mesh station that will receive the frame and posteriorly retransmit it to the network.

Because this specific protocol is expected to be employed in a network with a low number of nodes, the addresses are not defined by a MAC address, but by an identifier of the each node, which is expected to be 1 byte in length, for a maximum of 256 nodes.

4.1.1.4 Sequence Control Field

This field is used to help with the process of discarding duplicated frames. At each frame sequence number is attributed, controlled by a counter, which is incremented at every new frame processed by the MAC layer. When a frame is retransmitted, the sequence number does not change. Therefore, when a station receives two frames with the same sequence number, it probably means one of them was retransmitted after a failed acknowledgment.

It was defined the length of 1 byte for this field as it is not expected much traffic nor a big network, for the ROMOVI project. The sequence number can vary from 0 to 255, and it is assumed that a frame will be received before the counter turnaround.

4.1.2 Mesh Control Field

In order to enable multihop communication, it has been foreseen the existence of a mesh control field, whose structure can be found in fig. 4.4.

This field contains information useful for routing. It contains the following subfields:

- **Mesh time to live (TTL)** (1 byte): Represents an unsigned integer which corresponds to the remaining number of hops the frame can still be forwarded.
- **Mesh sequence number** (1 byte): Similarly to the Sequence control field, can help avoiding the transmission of duplicated frames. Each frame has a distinct mesh sequence number,

Mesh TTL	Mesh Sequence Number	Mesh Source Address
1byte	1byte	1byte

Figure 4.4: Mesh control field.

given from a counter. In a mesh network, a mesh station can receive multiple copies of the same frame from different neighbors. This field facilitates the filtering of the duplicated frames, helping the mesh station to avoid forwarding copies of the same frame.

- **Mesh source address** (1 byte): Carries the node address that forwarded the frame.

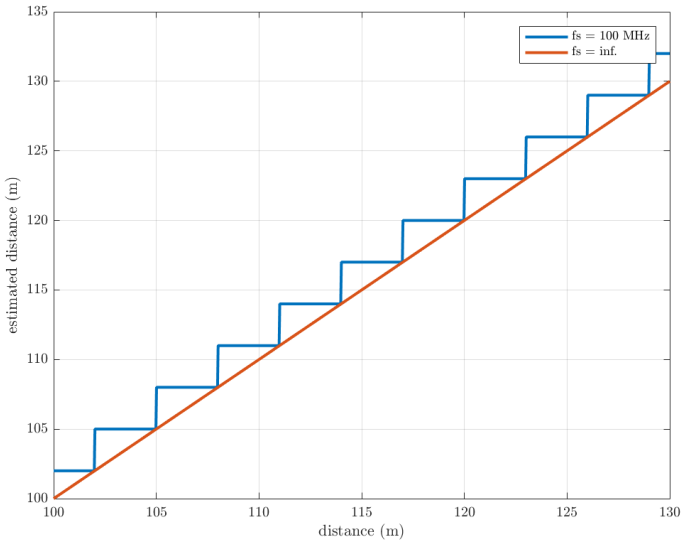
4.2 Geolocation system

In this section, it is made an approach of the system from the point of view of its geolocation functionality. First, it is presented a study on the SDS-TWR algorithm, which was performed for a better understanding of the errors that might affect its measures. Later, on a physical layer perspective, a preamble was developed and simulated, necessary for frame synchronization in the reception.

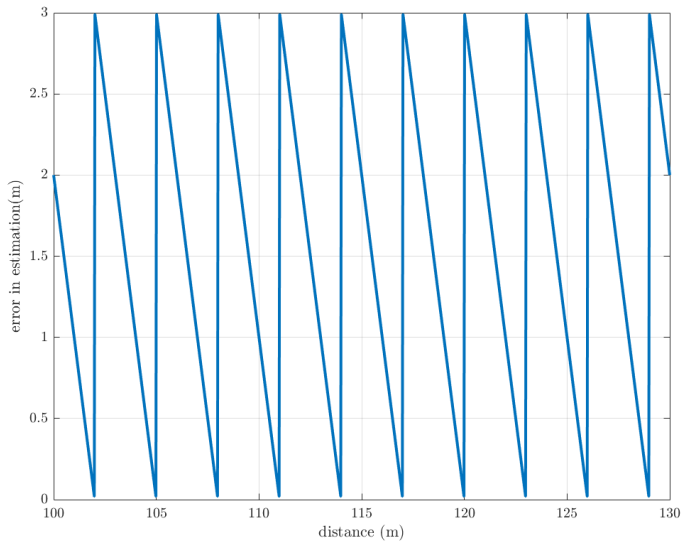
4.2.1 SDS-TWR simulation

The main goal of the system to be developed is to provide nodes positioning. For that, it should employ SDS-TWR algorithm, which was presented in chapter 2. By using this algorithm, two nodes can determine their relative distance to each other, by measuring the time a signal takes to travel from one node to another and back.

This algorithm was tested through MATLAB^{TM} simulation. The process began with a script which simulates two nodes at distances between 100 and 130 m, under ideal conditions (i.e.: not considering the errors mentioned in section 2.5). The measures resulting from the algorithm were compared with the real distance. The overall results can be found in fig. 4.5.



(a)



(b)

Figure 4.5: Distance estimated by node A in ideal conditions (a) and error in estimation (b).

In the figure, it can be seen a distance estimation with a sampling frequency of 100 MHz and the ideal case, which would be an infinite sampling frequency. From this simulation, we can conclude that there is an error in estimation associated to the resolution of the sampling frequency (fs), and it increases with the decrease of fs, varying from 0 to $\frac{c}{fs}$. For example, using a sampling frequency of 100 MHz results in a maximum error of 3 m. This error is zero at distances multiple of $\frac{c}{fs}$ and increases linearly with distance.

4.2.1.1 Clock drift

To the previous simulation, it was added one of the errors studied in chapter 2: clock drift. This error derives from small frequency variations in local clocks of the nodes, leading to an error in the estimation of the time of reply ($Treply$) of nodes.

To test this effect, there were introduced deviations in the clock frequencies of the two nodes, varying the difference between clock drift on node A (δ_A) and on node B (δ_B) between 0 and 40 ppm. The time instants illustrated on fig. 2.2 were calculated as follows:

$$\begin{cases} T1 = 0 \\ T2 = ToF \\ T3 = T2 + Treply_B \\ T4 = T3 + ToF \\ T5 = T4 + Treply_A \\ T6 = T5 + ToF \end{cases} \tag{4.1}$$

The time of flight estimated by node A was calculated according to (4.2), where $Treply_A$ and $Treply_B$ are given by (2.10), $fclk_A$ is the clock frequency on the node A (affected by clock drift) and $fclk$ is the ideal clock frequency. Note that $Treply_A$ and $Treply_B$ are estimated according to the opposite node's $Treply$.

$$ToF_{A_{est}} = \frac{(T4 - T1) - Treply_A}{2} \cdot \frac{f_{clk_A}}{f_{clk}} = \frac{2 \cdot ToF + Treply_B - Treply_A}{2} \cdot (1 + \delta_A) \tag{4.2}$$

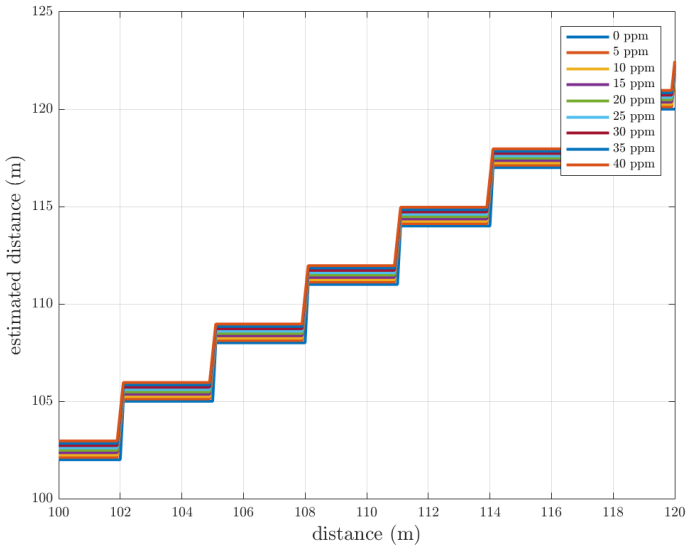
If we ignore the term $(1 + \delta_A)$, we can estimate the error originated by clock drift according to (4.3), where $Treply$ is the expected time of reply of nodes.

$$ToF_{A_{est}} - ToF \approx \frac{Treply_B - Treply_A}{2} \approx \frac{Treply}{2} \cdot \left(\frac{1}{1 + \delta_B} - \frac{1}{1 + \delta_A}\right) \tag{4.3}$$

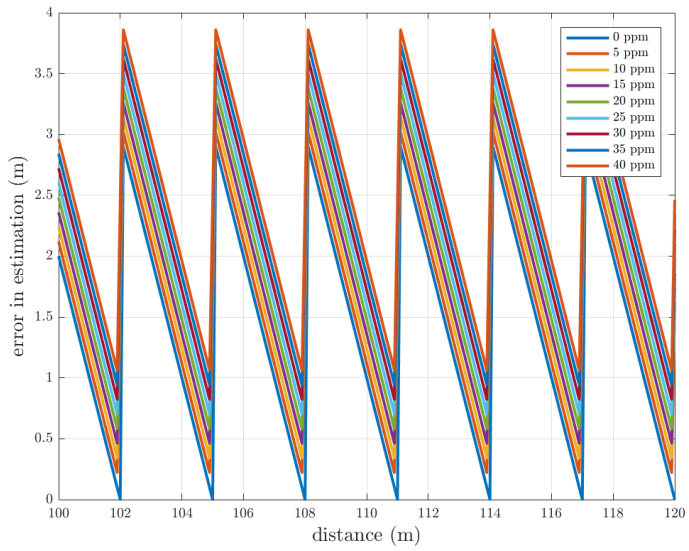
Simplifying the equation:

$$ToF_{A_{est}} - ToF \approx \frac{Treply}{2} \cdot (\delta_A - \delta_B) \tag{4.4}$$

For simulation purposes, it was assumed a $Treply$ of 160 ms. In fig. 4.6 can be seen the results of clock drift effect added to the previous simulation.



(a)



(b)

Figure 4.6: distance estimated by node A, under the influence of clock drift (a) and error in estimation (b).

By observing the figure, we can conclude that the distance and error waveforms stay the same as before, but, under the effect of clock drift, an offset is added to the wave. This offset increases proportionally to the clock drift. As it would be expected from (4.4), at a maximum difference between δ_A and δ_B of 40 ppm, there is an error of 1 m.

4.2.2 Preamble generation

One of the requirements of the system is to use OFDM. Therefore, it is needed a mechanism for time synchronization, which will allow the receiver to precisely determine the beginning of a frame. The mechanism adopted was the introduction of a preamble, based on Frank-Zadoff-Chu sequences, at the beginning of a frame [36]. Through a periodic autocorrelation technique, the receiver should be capable of determining the instant at which the frame arrived, which is useful for both data synchronization and providing measures to SDS-TWR algorithm.

4.2.2.1 Frank-Zadoff-Chu sequence

Frank-Zadoff-Chu (FZC) sequence is a complex-valued sequence, part of what is known as CAZAC (constant amplitude zero autocorrelation waveform) sequences, calculated according to:

$$a_k = \begin{cases} e^{j2\pi rk^2/N}, & \text{N even,} \\ e^{j2\pi rk(k+1)/N}, & \text{N odd} \end{cases} \tag{4.5}$$

where N is the sequence length and r is relative prime to N ([37–40]).

One of its most known properties is that its periodic autocorrelation function is zero, except for a lag of an integer multiple of the sequence length, as can be seen in fig. 4.7.

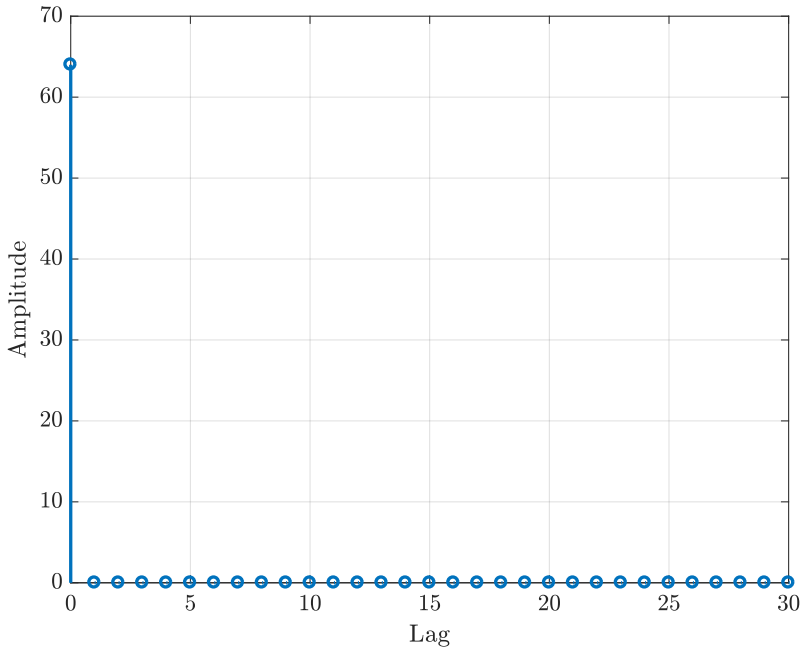


Figure 4.7: Periodic autocorrelation of FZC sequence.

4.2.2.2 Preamble simulation

A preamble (p) based on FZC sequences with $r = 63$ and $N = 64$ (X_{FZC}) was defined. It consists of a sequence composed by X_{FZC} and its conjugated, repeated 3 times, according to (4.6) and (4.7).

$$sequence = [X_{FZC}\ X_{FZC}^{*}\ X_{FZC}\ X_{FZC}^{*}] \tag{4.6}$$

$$p = [sequence\ sequence\ sequence] \tag{4.7}$$

In order to verify the performance of this preamble under different conditions of the link, a simulation was made, in MATLAB^{TM}.

It was calculated its autocorrelation, through a sliding window technique, of size N, by applying the expression in (4.8).

$$autocorr(p[k]) = \sum_{i=k}^{k+N/2} p[i] \cdot p^{*}[N+1-(i-k)] \tag{4.8}$$

Under ideal conditions, the plot of autocorrelation assuming $N = 64$ is illustrated in fig. 4.8. It is observed that peaks are repeated from 64 to 64 samples, at it should be expected.

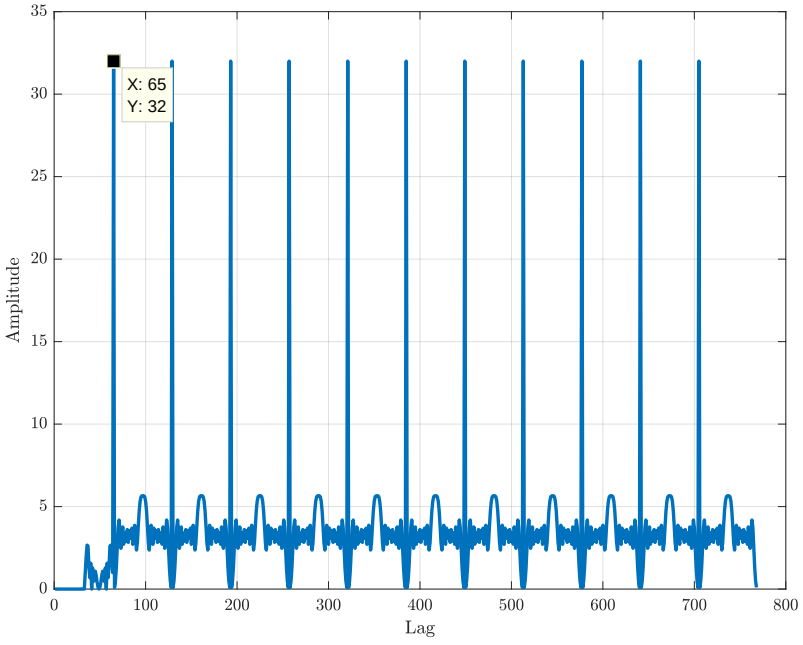
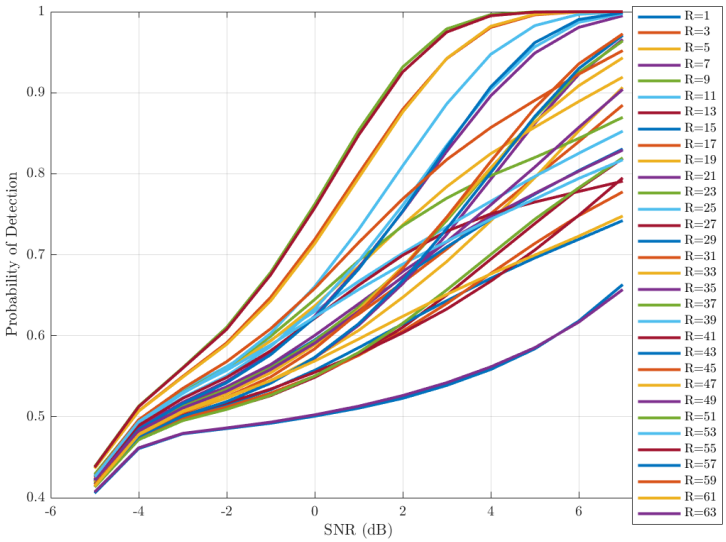


Figure 4.8: Autocorrelation of the preamble under ideal conditions.

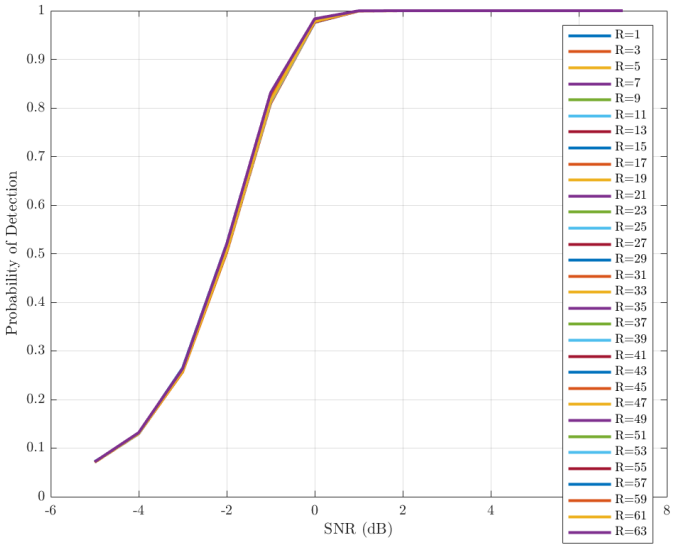
On the next simulation, the computation of the autocorrelation function was analyzed, with variations of the signal-to-noise ratio (SNR) . For that, the preamble was affected by additive white Gaussian noise (AWGN) and it was measured the probability of the peaks of the autocorrelation

matching those from ideal conditions. It was also tested autocorrelation against cross-correlation with a copy of the preamble itself.

In fig. 4.9 can be seen the results of probability of detection using cross-correlation (fig. 4.9a) and autocorrelation (fig. 4.9b) for SNR between -5 and 7 dB, for every possible value of r (which has to be relative prime to N), which gives every possible FZC sequences of length 64. Each simulation was repeated 100000 times for each value of r and SNR.



(a)



(b)

Figure 4.9: Probability of cross-correlation (a) and autocorrelation (b) peak detection under AWGN.

As it would be expected, it is possible to see that probability of detection decreases with the

degradation of the signal. It is also evident that cross-correlation behaves differently for different values of r. On the contrary, in the graphic of autocorrelation the lines almost overlap, which indicates this technique has a much uniform behavior for different FZC sequences. In fig. 4.10 it is compared the probability of detection using the FZC sequences that offer best results for cross-correlation ($r = 9$) and autocorrelation ($r = 63$).

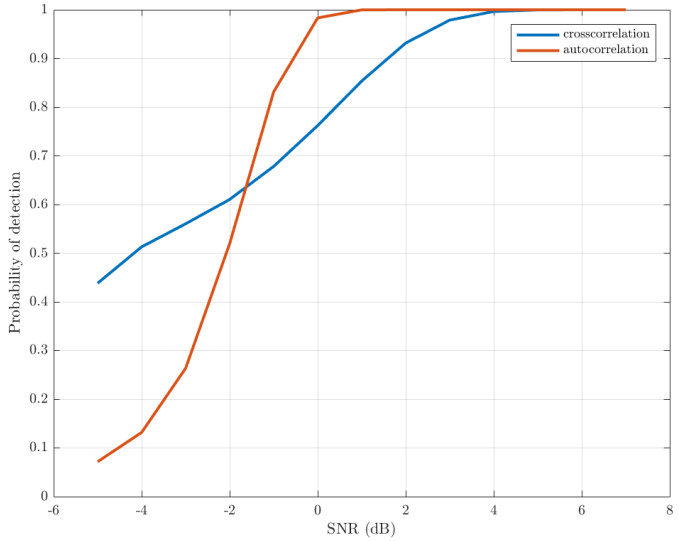


Figure 4.10: Performance comparison of cross-correlation and autocorrelation under AWGN.

Despite having lower results for lower values of SNR, autocorrelation shows much better results within the range of SNR values that would be expectable in real circumstances. For values greater than 0 dB, autocorrelation originates a probability of detection of almost 100%, which means peaks will be detected at the correct instant almost every time. For cross-correlation, this only happens near 4 dB. Therefore, it is easily concluded that autocorrelation offers more advantages than cross-correlation.

In the next simulation, SDS-TWR was simulated with the introduction of the preamble. T_{reply} is now calculated according to 4.9:

$$T_{reply} = T_{detection} + T_{send} \tag{4.9}$$

where $T_{detection}$ is the time interval from the beginning of the reception of the preamble to the detection of the last autocorrelation peak ($T_{detection}$) and T_{send} is the time needed for completely transmitting the preamble. For example, in ideal conditions, the last autocorrelation peak would be at sample 705. At a sampling frequency of 8 MHz, $T_{detection}$ would be $705/8 \times 10^6 = 88.125\ \mu s$. If clock drift is not considered, T_{send} would be always the same, which is $768 \cdot 8 \times 10^6 = 96 \mu s$, for a sampling frequency of 8 MHz. If some error is added in the calculation of $T_{detection}$ or T_{send}, an error is also introduced in the final estimation of distance.

As already observed in the last simulation, $T_{detection}$ is mainly affected by SNR, which affects the detection of autocorrelation peaks. In table 5.1 can be found the average distance estimation error and respective variance when AWGN is added to the preamble, for SNR values ranging from -5 and 5 dB.

Table 4.2: Average estimation error and variance according to SNR.

SNR (dB)	**AVG (m)**	**VAR (m)**
-5	423.8379	0.9038×10^5
-4	438.7421	0.9884×10^5
-3	438.3819	1.1488×10^5
-2	361.3193	1.3521×10^5
-1	153.9287	0.9165×10^5
0	18.5574	0.1303×10^5
1	0.2754	177.3560
2	0.0000	0.0000
3	0.0000	0.0000
4	0.0000	0.0000
5	0.0000	0.0000

This results match the results from the previous simulation (fig. 4.10). For SNR lower than 0 dB the average error in higher than 100 m, and increases with the degradation of the signal. As SNR gets higher, the error decreases and reaches 0 at 2 dB, which is the SNR value where the probability of detection is almost 100%.

Finally, in order to verify the influence of clock drift in the whole system, as well as carrier frequency offset in the performance of the preamble, a final MATLAB^{TM} simulation was performed, of which the results can be found in fig. 4.11.

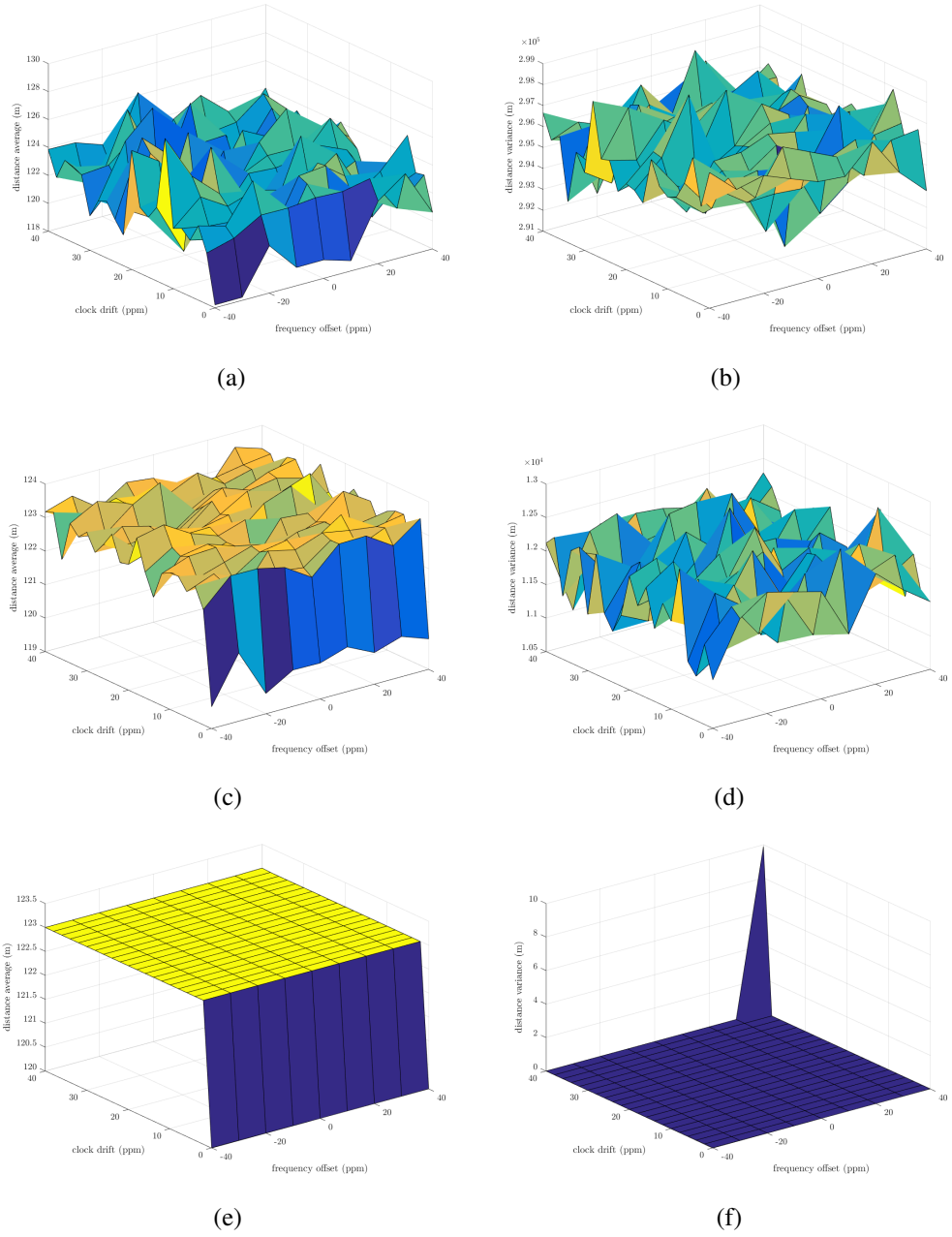


Figure 4.11: Distance estimation average and variance under clock drift and carrier frequency offsets for -5 dB SNR (a) and (b), 2 dB SNR (c) and (d) and 5 dB SNR (e) and (f).

By looking at the figure, it can be concluded that the preamble seems immune to carrier frequency offset. As to clock drift, combined with the error introduced by sampling frequency resolution, generates an average error of about 3 m for a real fixed distance of 120 m. As already mentioned, SNR affects the measures at lower values, which can be seen with the increase of the variance.

Chapter 5

Hardware Implementation

In this chapter, it is described the implementation of the preamble transmission and reception, in hardware.

Since this dissertation comes in continuance of a previous work, where a physical layer architecture based on orthogonal frequency division multiplexing (OFDM) was defined, some requirements have to be considered, such as a sampling frequency of 56 MHz. However, because digital to analog converters receive phase and quadrature components of the signal in an interleaves mode (i.e.: they are received in alternated sampling edges), it is required a main clock frequency of 112 MHz, which is 2 times the sampling frequency. Therefore, to control this system, it is used a main clock of 112 MHz with 3 enable signals, which are synchronized with the clock and are activated at each 14 (*i_enable_preamble*), 2 (*i_enable_filter*) and 1 (*i_enable*) clock cycles, as can be seen in fig. 5.1.

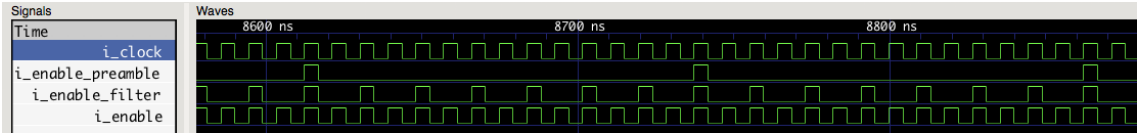


Figure 5.1: Control signals in the system.

In fig. 5.2 and fig. 5.3 are represented general schematics of the transmission and reception block diagram, respectively. A more detailed view of each block is presented in the following sections.

5.1 Transmission

The module which generates the preamble, in the transmitter, was implemented through a read-only memory (ROM) where the real and imaginary parts of the preamble samples were stored. This module is controlled by a main clock at 112 MHz and is enabled by a signal which is active each 14 clock cycles, generating a 16 bit sample (real and imaginary parts) at a sample rate of $112/14 = 8\ MHz$. The block diagram of this module can be seen in fig. 5.4.

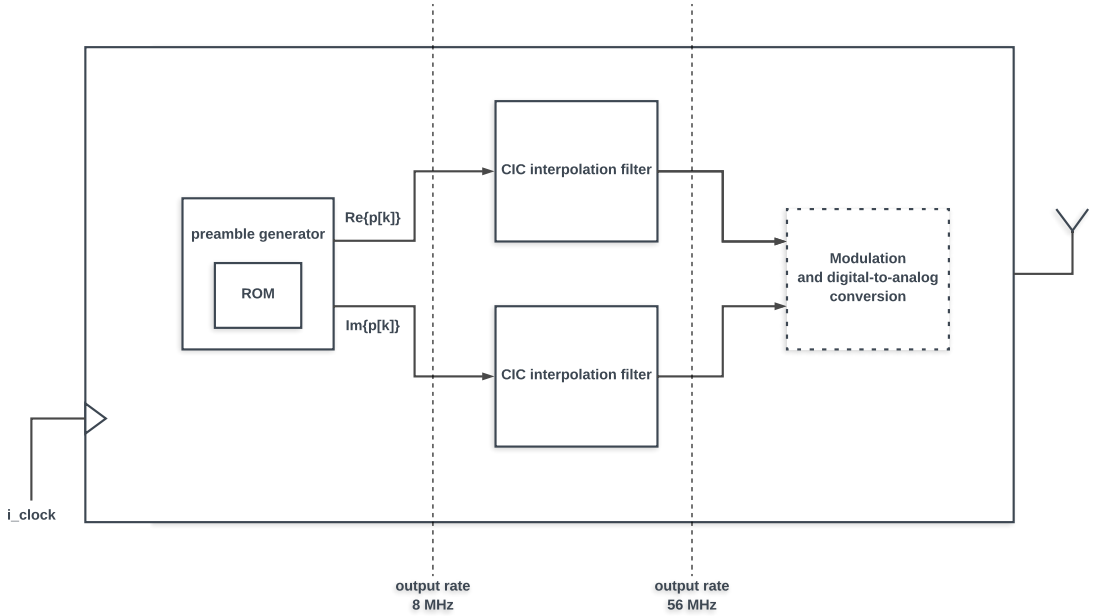


Figure 5.2: Transmission block diagram.

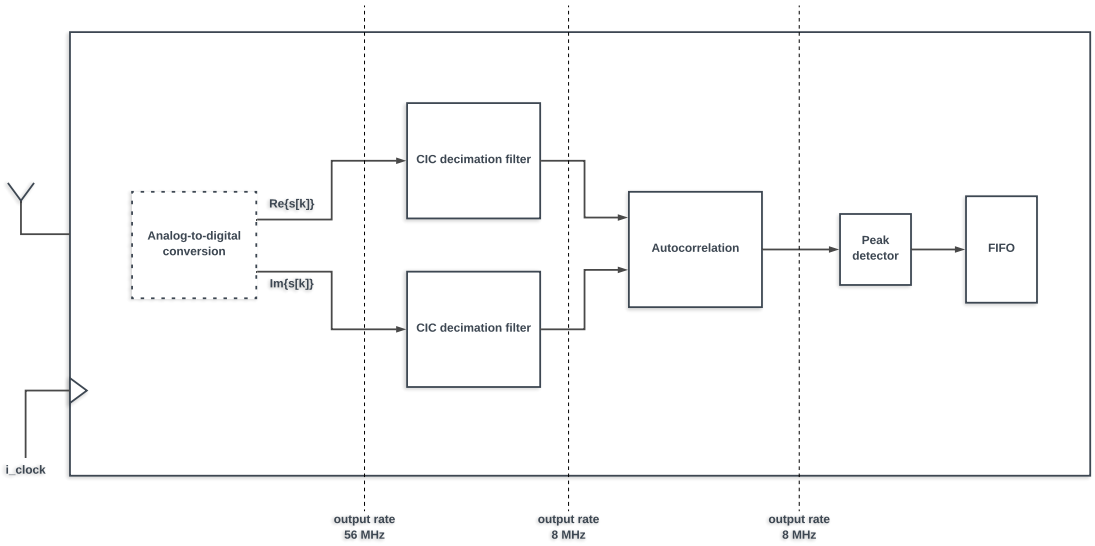


Figure 5.3: Reception block diagram.

The samples then go through a cascaded integrator-comb (CIC) interpolation filter, described in [41]. CIC filters have the advantage of requiring no multipliers and using limited storage, which leads to more economical hardware implementations. They consist in two section: the integrator section, which operates at the higher sampling frequency (in this case, 56 MHz), and the comb (derivation) section, which operates at the lower sampling frequency (in this case, 8 MHz).

The interpolation of the signal comes with the need to reduce digital conversion distortion. However, if we simply oversample (introduce zeros between samples) the signal, spectral replicas will appear among the entire sampling bandwidth. To reduce the signal to its original bandwidth,

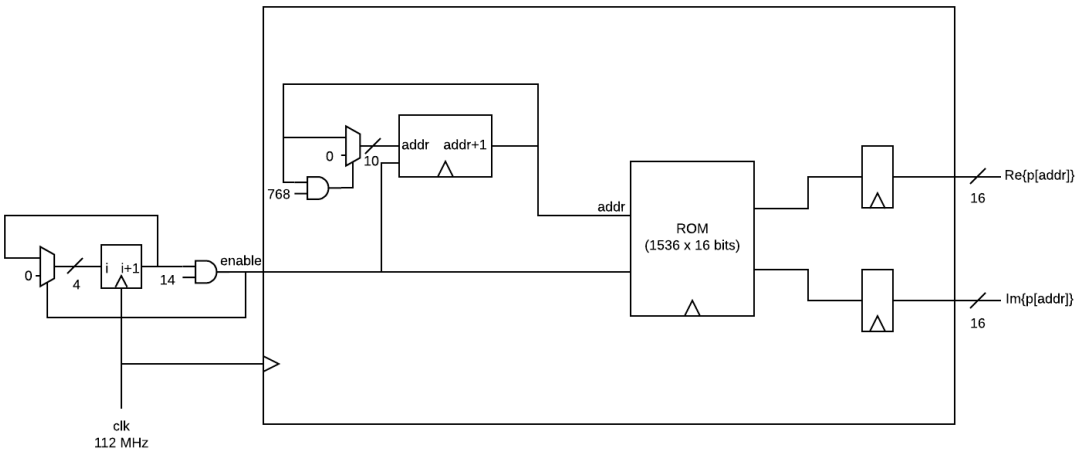


Figure 5.4: Preamble generation block diagram.

a low pass interpolation filter should be used [42]. This process can be seen in fig. 5.5, where the power spectral density is represented before and after filtering.

By comparing fig. 5.5a and 5.5b, it is possible to see that, after upsampling, the signal goes from being contained in a bandwidth of 8 MHz to be spread over a bandwidth of 56 MHz. However, this comes with the addition of replicas to the spectrum, the same as the interpolation factor (7, in this case). These replicas can be filtered by applying a low pass interpolation filter, making the signal returning to be contained in a 8 MHz bandwidth, as seen in fig. 5.5c.

For concept proof, it was designed a filter with a pipeline structure with 10 stages of comb and integration. The two sections are connected by an upsampler stage, as seen in Fig. 5.6. It was chosen an interpolation factor is $R = 7$ for an output rate of 7 times the input rate, $8MHz \times 7 = 56MHz$, as intended.

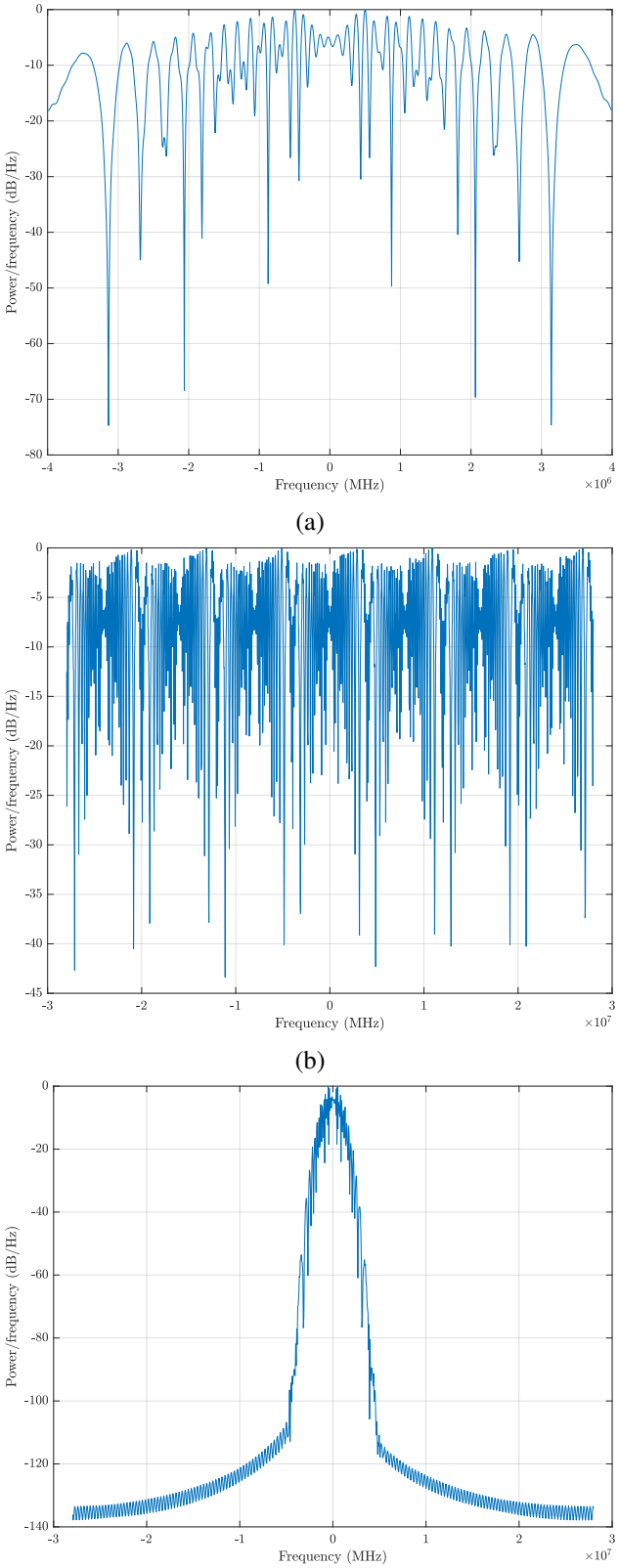


Figure 5.5: Normalized power spectral density of preamble in baseband (a), upsampled by a factor of 7 (b), after filtering (c).

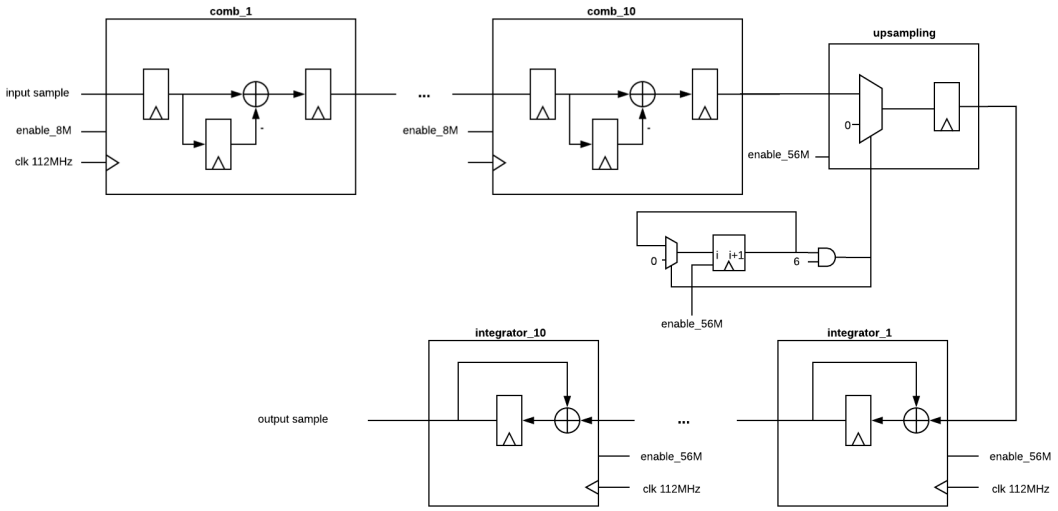


Figure 5.6: Block diagram of the designed CIC interpolation filter.

5.2 Reception

At reception, the opposite process has to be performed: the signal must be decimated, by a decimation CIC filter. The designed filter has the same structure as the interpolation filter, as seen in fig. 5.7. The received signal at 56 MHz is decimated by 7, which leads to the original sample rate of 8 MHz.

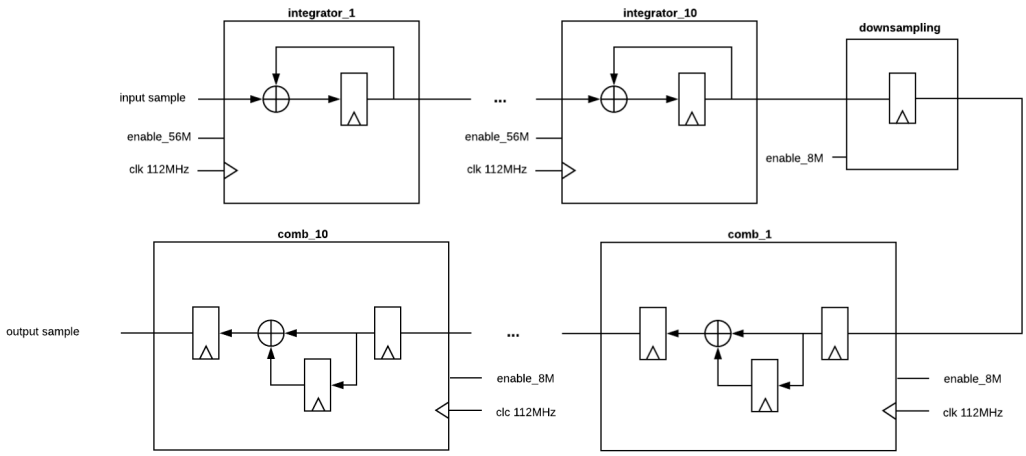


Figure 5.7: Block diagram of the designed CIC decimation filter.

The filtered signal is then stored into a 65 sample size buffer and the autocorrelation function is performed, according to the technique described in chapter 4:

$$autocorr(s_{received}) = \sum_{i=0}^{31} s[i] \cdot s^*[64-i] = \sum_{i=0}^{31} (A_i + jB_i) \tag{5.1}$$

$$A_i = (\Re\{s[i]\} \cdot \Re\{s[64-i]\} + \Im\{s[i]\} \cdot \Im\{s[64-i]\}) \tag{5.2}$$

$$B_i = (\Im\{s[i]\} \cdot \Re\{s[64-i]\} - \Re\{s[i]\} \cdot \Im\{s[64-i]\}) \tag{5.3}$$

where $s_{received}$ is the signal resulting from the last 65 received samples, and A_i and B_i denote the real and imaginary components of the inner product, respectively.

The autocorrelation module receives one sample at each 14 clock cycles corresponding to the real and imaginary parts. In order to reduce the number of used digital signal processing (DSP) blocks, which exist in a limited number in many boards, as well to simplify the used logic in the circuit, the expression in (5.1) is implemented by using only 10 multiplier blocks. A_i (5.2) and B_i (5.3) are calculated alternately.

In Fig. 5.8, it is represented a diagram of the implementation of one iteration, for either A_i or B_i, depending on the value of *selectRI*, which switches at each 7 clock cycles. This design is repeated 5 times. Each multiplier is capable of performing 7 multiplications, in a total of 64 *per* received sample, according to the selected real or imaginary operation, determined by *selectRI*, as aforementioned. For this, each multiplier has two multiplexers, which determine which samples will be processed at each clock cycle, controlled by a 3 bits signal, *select06*. The values are then stored into registers and summed by a pipelined structure composed by sum blocks.

In order to keep the values in a predefined scale, the adopted strategy was to divide the absolute value of the autocorrelation result by the norm of the last 32 received samples, which represents the energy of the received signal, according to the expression in (5.4). Because the square root is a complex operation to avoid, in order to remove it, both operands were squared.

$$\frac{\|autocorr(s_{received}[k])\|}{norm(s_{received}[k])} = \frac{\sqrt{A_k^2 + B_k^2}}{\sqrt{\sum_{i=0}^{31} \|s[k-i]\|^2}} = \frac{\left(\sqrt{A_k^2 + B_k^2}\right)^2}{\left(\sqrt{\sum_{i=0}^{31} \|s[k-i]\|^2}\right)^2} \tag{5.4}$$

This operation required the use of an Intel® FPGA Integer Arithmetic IP core, the LPM-Divide [43].

LPM-Divide is a megafunction provided in the Intel Quartus® Prime software, which is optimized for Altera FPGA devices, with improved area and performance when compared to generic basic logic elements. This block has parameterizable input data widths for the numerator and denominator values, supports both signed and unsigned data representation and allows for pipelining with a parameterizable output latency. It also provides control inputs, such as an asynchronous clear and clock enable. The used configuration has a pipeline length of 21 stages, in order to meet the maximum frequency requirements, and uses the following signals:

- **numerator:** 32 bits unsigned input. Corresponds to the non-normalized autocorrelation value;

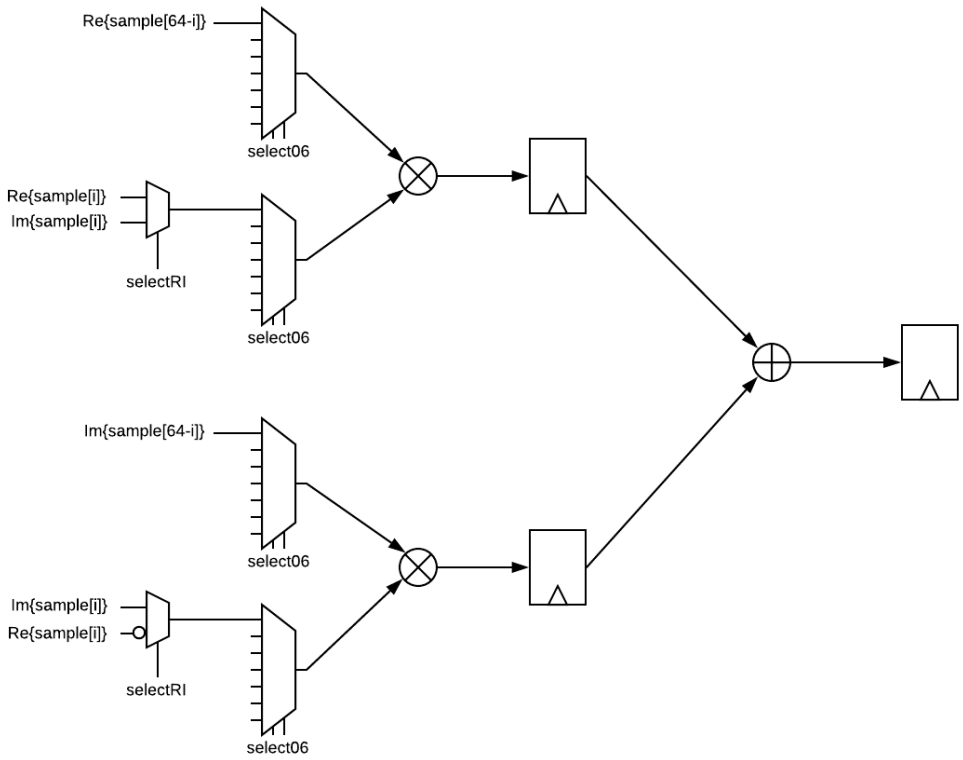


Figure 5.8: Block diagram of an autocorrelation module section implementing one operation over a sample.

- **denominator:** 16 bits unsigned input. Corresponds to the norm of the 32 of the last 32 received samples;
- **clock:** input. Receives the system's main clock;
- **quotient:** 32 bits unsigned output. Corresponds to the quotient of the operation;
- **remain:** 16 bits unsigned output. Corresponds to the remain of the operation (not used).

The quotient of the divider is stored as the autocorrelation result, and sent to another module, the peak detector. This module analyzes the values and determines the maximum value within a window of 64 samples. For eliminating "false detections", it was set a threshold, which corresponds to half of the diagram of amplitude of an autocorrelation peak, in ideal conditions. The peak detector was implemented according to the fig. 5.9.

Because the reception of one full preamble implies the reception of 11 peaks, it was added a new register to count the number of received peaks at each $64 \cdot 11$ received values, as it can be seen in fig. 5.10.

The output values of the peak detector are finally stored in a buffer. This buffer was implemented as a FIFO with parameterizable size.

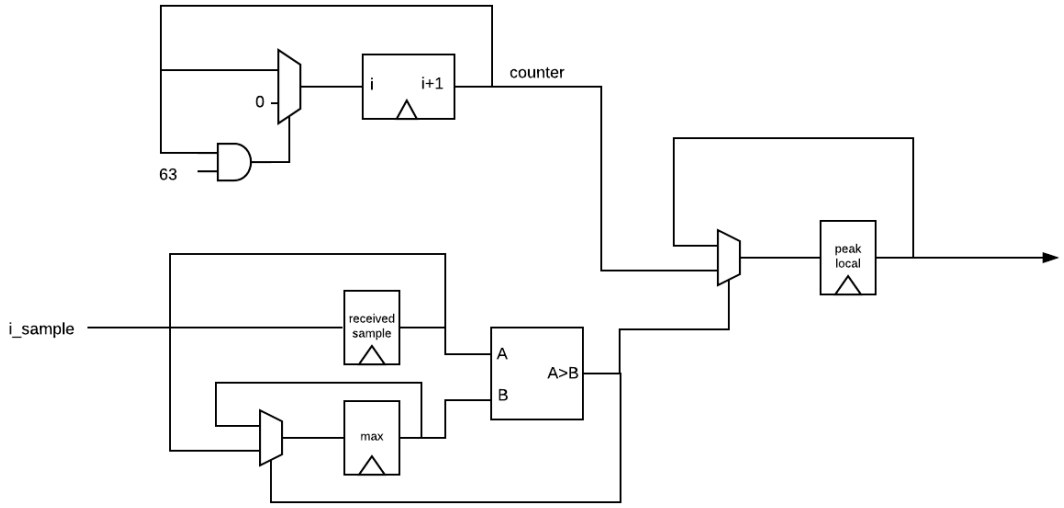


Figure 5.9: Block diagram of the peak detector section which determines the peak each 64 samples.

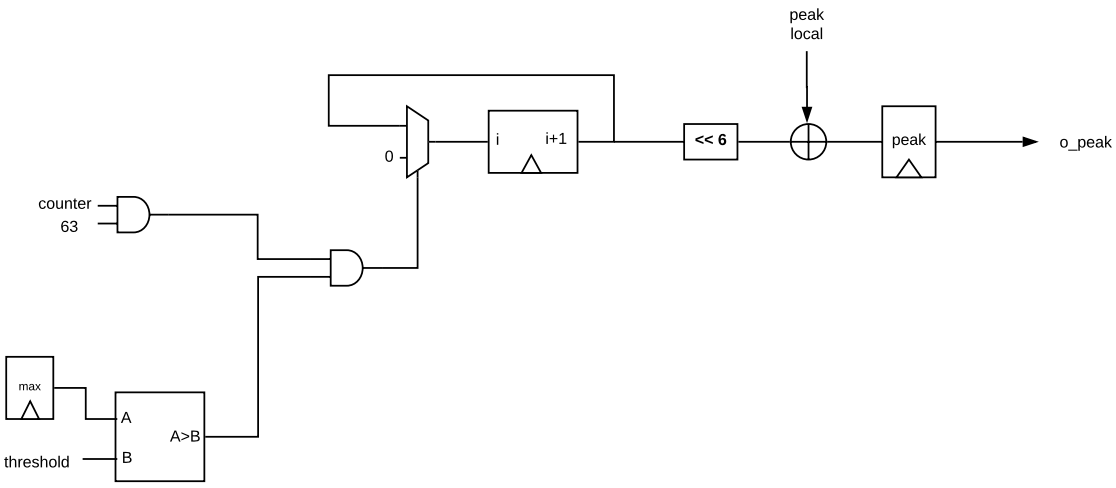


Figure 5.10: Block diagram of the peak detector section which calculates the peak in the entire sequence.

5.3 Functional verification

In order to verify the logic behavior of the developed modules, a testbench was made. It was enabled the transmission of the preamble, and the waveforms at the output of both the interpolation and decimation filters and autocorrelation block were analyzed and compared. Furthermore, the peaks index were tested to verify their occurrence each 64 samples. To verify the testbench, a comparison with the results generated in MATLAB^{TM} was made, being presented in the following figures.

- **Interpolation CIC filter**

  In fig. 5.11 can be seen the output of the interpolation CIC filter, for both real (*o_sample_real*) and imaginary (*o_sample_imag*) components. When compared to waves generated through MATLAB^{TM} simulation (fig. 5.12), we can conclude the waves are very similar, which validates the transmission module behavior.

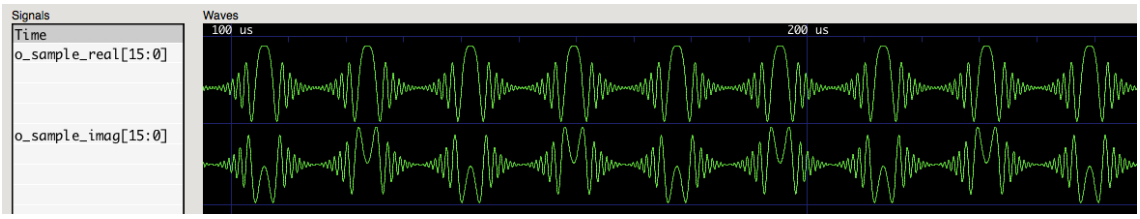
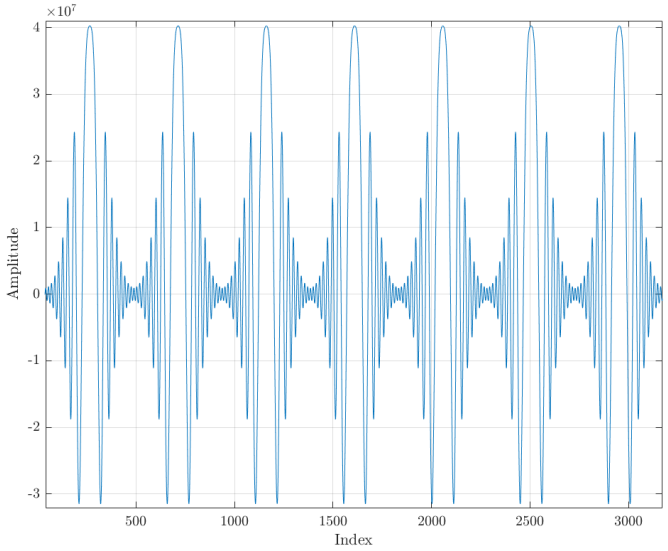


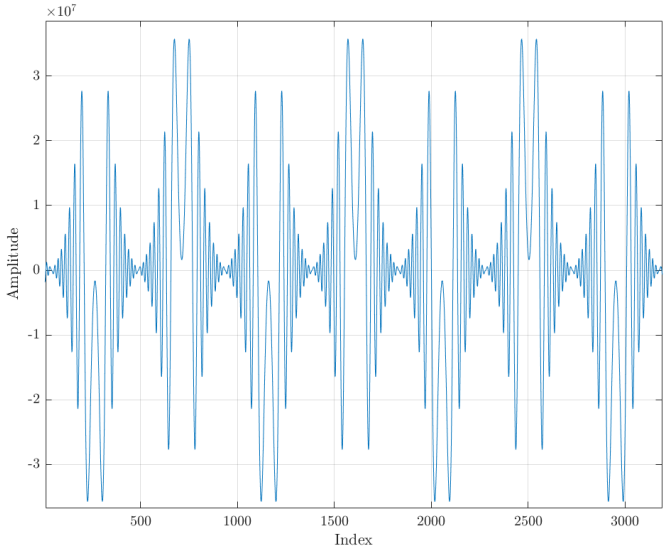
Figure 5.11: Interpolation waveforms generated from circuit simulation.

- **Decimation filter**

  In fig. 5.13 it can be seen the output of the decimation filter, at the reception. Real part is illustrated as *r_real* and imaginary part is illustrated as *r_imag*. The waveforms at the output of the circuit appear to be similar to those from MATLAB^{TM} simulation, which are represented in fig. 5.14. However, it can be noted a small distortion of the signal. This distortion can result from the concatenation of the values at the output of the interpolation filter, which derives from the need to have a 16 bit output sample.



(a)



(b)

Figure 5.12: Interpolation waveforms generated from MATLAB™ simulation: real (a) and imaginary (b) components.

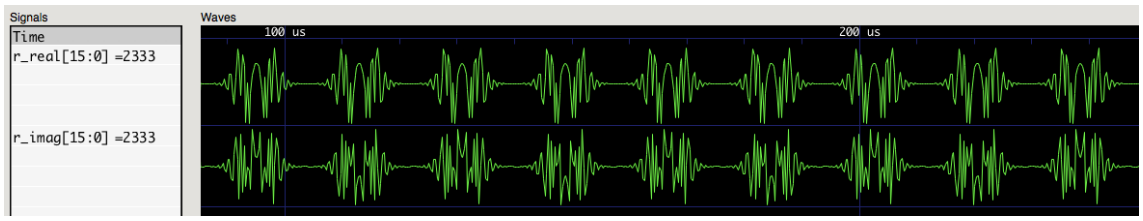
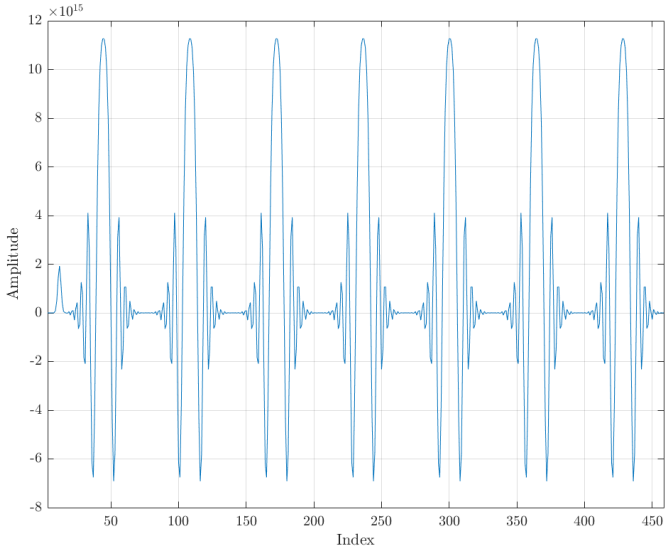
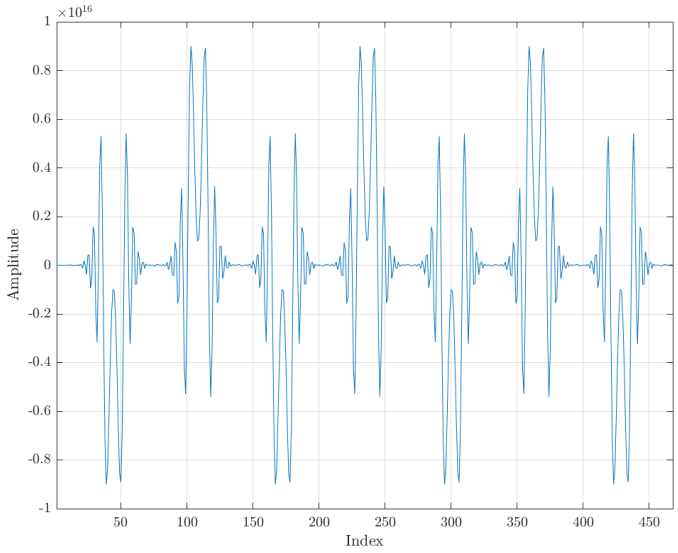


Figure 5.13: Decimation waveforms generated from circuit simulation.



(a)



(b)

Figure 5.14: Interpolation waveforms generated from MATLAB^{TM} simulation: real (a) and imaginary (b) components

- **Autocorrelation and peak detector**

  The autocorrelation functional verification can be made along with the peak detector. An example of a typical output of both blocks can be seen in fig. 5.15. By observing the wave represented by *o_correlation*, which is the output of the autocorrelation block, it can be seen that the peaks are well defined and equally spaced. The indexes at which they occur are indicated by the output of the peak detector, *peak*. Each value is outputted for each 64 received autocorrelation values. In this figure, peaks are detected at 85, 149, 213, 277

samples, and so on. Therefore, it can be concluded that peaks appear separated from 64 samples, and both autocorrelation and peak detector modules are working as expected. It should be noted that there is an error code, which is outputted when the peak does not detect any value above a threshold, corresponding to 65535, or 0xFFFF, as seen in the figure.

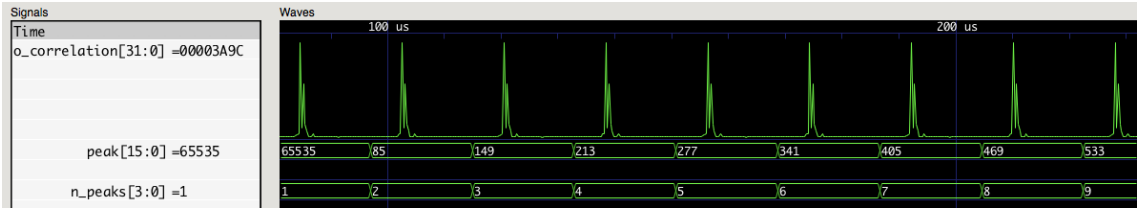


Figure 5.15: Autocorrelation waveform and detected peak indexes generated from circuit simulation.

5.4 Synthesis results

Besides functional verification, a resource usage and maximum operating frequency estimation was performed, for each implemented block, and is presented in table 5.1.

Table 5.1: Resource usage and maximum operating frequency estimation report.

Module	ALMs	Total Registers	DSPs	Fmax
Preamble generator	787	58	0	115.92 MHz
Interpolation filter	607	1598	0	175.07 MHz
Decimation filter	648	1696	0	190.88 MHz
Autocorrelation	2328	3880	23	64.33 MHz *
Peak detector	60	97	0	109.06 MHz *
FIFO	492	918	0	93.14 MHz *
top-level	5764 (20%)	11410	23 (15%)	63.39 MHz *

As expected, the block which consumes the most resources is the autocorrelation, as it is the one that uses the most complex logic.

It appears to exist a problem at reception, regarding the maximum operating frequency. There are three blocks which can't reach the 112 MHz required. However, since the maximum frequency of the system is higher than the required sampling frequency, this part of the system doesn't need to work at 112 MHz, only 56 MHz, which is the rate at which new samples are received. Thus, it would be possible to adapt the autocorrelation block to operate at this frequency, by duplicating the number of DSPs and, thus, performing the calculus of A_i (5.2) and B_i (5.3) in parallel.

5.5 Test in FPGA

In order to test the developed system in hardware, an FPGA Development Kit was used, the Terasic Cyclone V GX Starter Kit [44], which embeds the Cyclone V 5CGXFC5C6F27C7N [45], from Intel Altera. It was added a RS-232 Uart from the Intel® IP Catalog within the Quartus II software [46]. A top level module was implemented, which implements the finite state machine found in fig. 5.16:

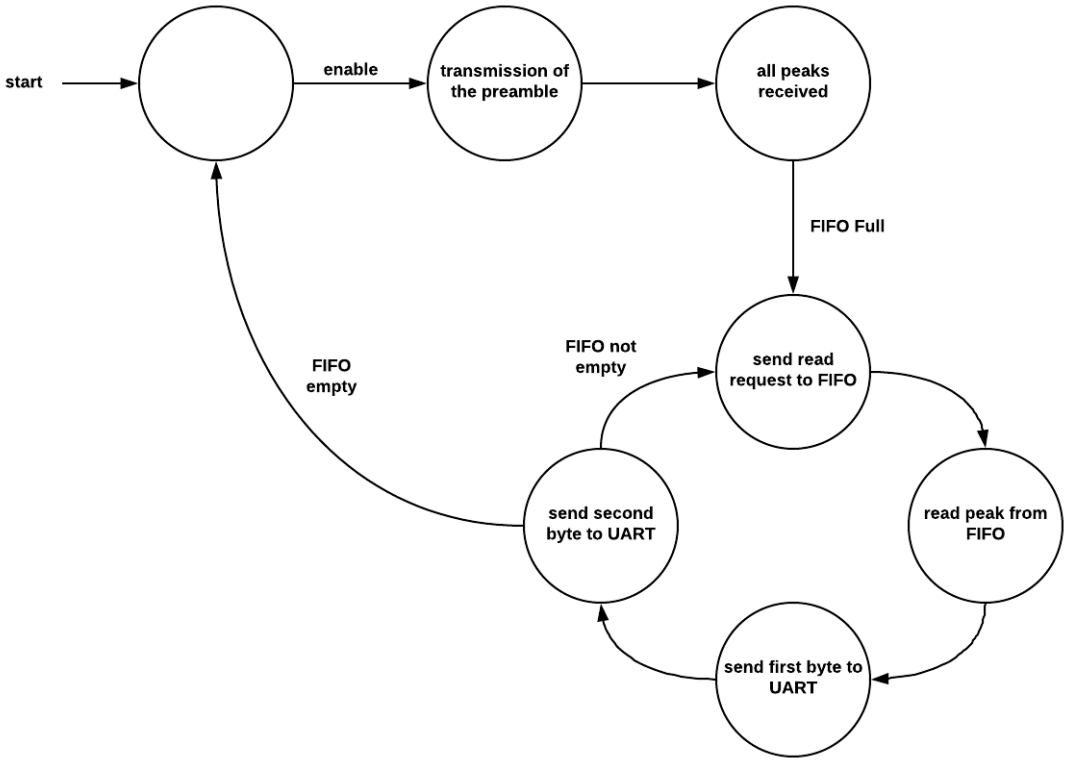


Figure 5.16: Finite state machine implementing the test algorithm.

1. In the initial state, the system is in an waiting loop until the enable signal is received;
2. When the enable is activated, by one of the switches of the board, the preamble starts being transmitted. 5 frames are sent.
3. The samples are received by the reception block, are decimated and autocorrelation in performed. Peaks are detected and stored in the FIFO.
4. When the FIFO is full, the process of reading the FIFO starts:
   (a) A read request is sent to the FIFO.
   (b) After 1 clock cycle, the sample to be read is ready.
   (c) One clock cycle later, the less significant byte of the sample is sent to the UART.

(d) In the clock cycle, the most significant byte of the sample is sent to the UART. If there are still samples to be read in the FIFO, the system goes to state 4.(a). If the FIFO is empty, the system returns to the initial state.

On the receiving side of the UART, a script was written in MATLAB^{TM}, which receives each byte and calculates the 16 bit samples at each 2 bytes.

By performing this test, it is possible to compare the indexes of the autocorrelation peaks generated in hardware with those from simulation.

Chapter 6

Conclusions

This dissertation intended to propose a geolocation architecture for harsh environments. By using SDS-TWR, accurate ranging measures are possible, enabling the possibility of relative positioning. Combined with a mesh architecture, which is, by definition, decentralized and self-configured, nodes in a network may establish links among each other and perform the ranging algorithm with their neighbors.

Therefore, the main contributions of this work were:

- the study of SDS-TWR algorithm and its major error sources, from which was concluded its measures are affected mainly by clock drift and sampling resolution;
- the proposal of a MAC layer based on a mesh architecture and the definition of a new frame format with geolocation functions, with the addition of a new frame type, making it suitable for a posterior development of SDS-TWR algorithm;
- the development of a preamble based in FZC sequences and implementation in hardware of a system for its transmission and reception, for demonstrating the defined preamble is suitable for nodes synchronization.

6.1 Future Work

This dissertation is part of a complex system, which combines subjects from various areas. As such, there are still several tasks to be completed.

First, at the physical layer, an optimization of the developed system for preamble transmission/ reception must be made, in order to increase the maximum operation frequency. This passes through optimizing the autocorrelation function block design or the algorithm itself. The CIC filters should also be optimized, for an increase of their performance.

The development of a MAC layer employing the proposed mechanisms and the defined frame format and corresponding integration with the physical layer should be made. It should also be made a study on path selection protocols, for posteriorly enabling data transmission among all network.

Finally, the SDS-TWR algorithm should be implemented for the system to be completed and ready to be tested in a real scenario.

Bibliography

[1] F. N. dos Santos, H. Sobreira, D. Campos, R. Morais, A. P. Moreira, and O. Contente, "Towards a Reliable Monitoring Robot for Steep Slope Vineyards Monitoring," in *Journal of Intelligent and Robotic Systems*, February 2016, pp. 1–16.

[2] J. Mendes, F. Santos, I. Castelão, T. Ramalho, C. Duarte, F. Almeira, and C. Carlos, "Romovi - robô modular e cooperativo para vinhas de encosta," *Agrotec*, October 2017.

[3] J. Hightower and G. Borriello, "Location sensing techniques," IEEE Computer, Tech. Rep., 2001.

[4] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov 2007.

[5] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and system," *Journal of Computer Networks and Communications*, vol. 2013, p. 12, 2013. [Online]. Available: http://dx.doi.org/10.1155/2013/185138

[6] D. Neirynck, E. Luk, and M. McLaughlin, "An alternative double-sided two-way ranging method," in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, Oct 2016, pp. 1–4.

[7] A. I. Baba and M. M. Atia, "Burst mode symmetric double sided two way ranging," in *2011 IFIP Wireless Days (WD)*, Oct 2011, pp. 1–3.

[8] M. Pelka, D. Amann, M. Cimdins, and H. Hellbriick, "Evaluation of time-based ranging methods: Does the choice matter?" in *2017 14th Workshop on Positioning, Navigation and Communications (WPNC)*, Oct 2017, pp. 1–6.

[9] H. Kim, "Double-sided two-way ranging algorithm to reduce ranging time," *IEEE Communications Letters*, vol. 13, no. 7, pp. 486–488, July 2009.

[10] R. Dalce, A. van den Bossche, and T. Val, "Reducing localisation overhead: A ranging protocol and an enhanced algorithm for uwb-based wsns," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–5.

[11] ——, "Indoor self-localization in a wsn, based on time of flight: Propositions and demonstrator," in *International Conference on Indoor Positioning and Indoor Navigation*, Oct 2013, pp. 1–6.

[12] M. Buevich, N. Rajagopal, and A. Rowe, "Hardware assisted clock synchronization for real-time sensor networks," in *2013 IEEE 34th Real-Time Systems Symposium*, Dec 2013, pp. 268–277.

[13] S. Frattasi and F. Rosa, *Mobile Positioning and Tracking: From Conventional to Cooperative Techniques*, ser. Wiley - IEEE. Wiley, 2017. [Online]. Available: https://books.google.pt/books?id=rvImDwAAQBAJ

[14] L. Smaini, *RF analog impairments modeling for communication systems simulation: application to OFDM-based transceivers. Radio frequency analog impairments modeling for communication systems simulation*. Chichester: J. Wiley & Sons, 2012. [Online]. Available: http://cds.cern.ch/record/1598453

[15] B. Razavi, "Design considerations for direct-conversion receivers," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 6, pp. 428–435, Jun 1997.

[16] M. J. Segura, V. A. Mut, and H. D. Patino, "Wavelet correlation toa estimation with dynamic threshold setting for ir-uwb localization system," in *2009 IEEE Latin-American Conference on Communications*, Sept 2009, pp. 1–6.

[17] Z. Sahinoglu, S. Gezici, and I. Gvenc, *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. New York, NY, USA: Cambridge University Press, 2011.

[18] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, Sept 2005.

[19] I. Akyildiz and X. Wang, *Wireless Mesh Networks*, ser. Advanced Texts in Communications and Networking. Wiley, 2009. [Online]. Available: https://books.google.pt/books?id=utTrj26vay0C

[20] A. Rajandekar and B. Sikdar, "A survey of mac layer issues and protocols for machine-to-machine communications," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 175–186, April 2015.

[21] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-mac: A hybrid mac for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511–524, June 2008.

[22] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto, "A wireless sensor mac protocol for bursty data traffic," in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2007, pp. 1–5.

[23] Y. Chen and W. Wang, "Machine-to-machine communication in lte-a," in *2010 IEEE 72nd Vehicular Technology Conference - Fall*, Sept 2010, pp. 1–4.

[24] A. Lo, Y. W. Law, M. Jacobsson, and M. Kucharzak, "Enhanced lte-advanced random-access mechanism for massive machine-to-machine (m2m) communications," in *27th World Wireless Research Forum (WWRF) Meeting*. WWRF27-WG4-08,, 2011, pp. 1–5.

[25] "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, Dec 2016.

[26] A. Rahman and P. Gburzynski, "Hidden problems with the hidden node problem," in *23rd Biennial Symposium on Communications, 2006*, 2006, pp. 270–273.

[27] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O'Reilly Media, Inc., 2005.

[28] T.-S. Ho and K.-C. Chen, "Performance analysis of ieee 802.11 csma/ca medium access control protocol," in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, vol. 2, Oct 1996, pp. 407–411 vol.2.

[29] "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 amendment 10: Mesh networking," *IEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007 as amended by IEEE 802.11k-2008, IEEE 802.11r-2008, IEEE 802.11y-2008, IEEE 802.11w-2009, IEEE 802.11n-2009, IEEE 802.11p-2010, IEEE 802.11z-2010, IEEE 802.11v-2011, and IEEE 802.11u-2011)*, pp. 1–372, Sept 2011.

[30] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "Ieee 802.11s: The wlan mesh standard," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, February 2010.

[31] G. R. Hiertz, S. Max, R. Zhao, D. Denteneer, and L. Berlemann, "Principles of ieee 802.11s," in *2007 16th International Conference on Computer Communications and Networks*, Aug 2007, pp. 1002–1007.

[32] "Ieee standard for information technology–local and metropolitan area networks–specific requirements–part 11: Wireless lan medium access control (mac) and physical layer (phy)

specifications - amendment 8: Medium access control (mac) quality of service enhancements," *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, pp. 1–212, Nov 2005.

[33] M. Zogkou, A. Sgora, P. Chatzimisios, and D. D. Vergados, "Edca mechanism and mobility support evaluation in ieee 802.11s wmns," in *2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct 2014, pp. 204–209.

[34] S. R. Das, C. E. Perkins, and E. M. Belding-Royer, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, Jul. 2003. [Online]. Available: https://rfc-editor.org/rfc/rfc3561.txt

[35] V. Rishiwal, S. K. Agarwal, and M. Yadav, "Performance of aodv protocol for h-manets," in *2016 International Conference on Advances in Computing, Communication, Automation (ICACCA) (Spring)*, April 2016, pp. 1–4.

[36] M. M. U. Gul, S. Lee, and X. Ma, "Robust synchronization for ofdm employing zadoff-chu sequence," in *2012 46th Annual Conference on Information Sciences and Systems (CISS)*, March 2012, pp. 1–6.

[37] R. Frank, S. Zadoff, and R. Heimiller, "Phase shift pulse codes with good periodic correlation properties (corresp.)," *IRE Transactions on Information Theory*, vol. 8, no. 6, pp. 381–382, October 1962.

[38] R. Frank, "Polyphase codes with good nonperiodic correlation properties," *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 43–45, January 1963.

[39] D. Chu, "Polyphase codes with good periodic correlation properties (corresp.)," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 531–532, July 1972.

[40] B. M. Popovic, "Generalized chirp-like polyphase sequences with optimum correlation properties," *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1406–1409, Jul 1992.

[41] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 155–162, Apr 1981.

[42] R. R. Lyons, "How discrete signal interpolation improves digital-to-analog conversion."

[43] *lpm_divide Megafunction User Guide*, Altera®, June 2007.

[44] *Cyclone V GX Starter Kit user manual*, Altera®, June 2014.

[45] *Cyclone V Device Datasheet*, intel®, May 2018.

[46] *RS232 UART for Intel DE-Series Boards*, intel® FPGA, November 2016.