From Low-Cost Spectrum Monitoring to 5G Networks: Algorithms and Systems for Localizing and Identifying Wireless Transmissions

by

Yago Lizarribar Carrillo

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Telematics Engineering

Universidad Carlos III de Madrid

Advisor(s):

Dr. Domenico Giustiniano

February 2025

From Low-Cost Spectrum Monitoring to 5G Networks: Algorithms and Systems for Localizing and Identifying Wireless Transmissions

Prepared by:

Yago Lizarribar Carrillo
IMDEA Networks Institute
University Carlos III of Madrid
Contact: yago.lizarribar@imdea.org

Under the advice of:

Domenico Giustiniano
IMDEA Networks Institute

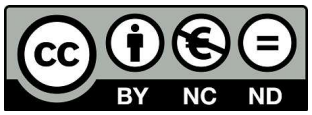
In the department:

Telematics Engineering Department, Universidad Carlos III de Madrid

This work has been supported by:



Unless otherwise indicated, the content of is thesis is distributed under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA).



Onin: indarra eman zenidan bide honetan jarraitzeko.
Hau lortu dut zuri esker. Betirako gara.

A mi padres: me lo habéis dado todo, ojalá algún día
devolveros la milésima parte. Gracias.

Acknowledgements

Phewww... What a journey! There are many people to thank for these five years, but I will start with the obvious ones. To my parents. Throughout my life, you have supported and helped me reach all my goals, no questions asked. Every time I wanted to try something new, you were there to lift me up and let me reach new heights. You provided shelter and financial support whenever I needed it, without asking anything in return. I look up to both of you and can only wish to be as good a father to my children as you were to me. Also to my brothers. You always gave me your views clearly and bluntly. You have always helped me stay on the ground and thanks to it I have learned to be the person I am today. I hope I was also a somewhat good brother to you guys.

The people of IMDEA Networks. In order, I would like to thank Prof. Antonio Fernandez de Anta for introducing me to the institute and the incredible people within. Then to Prof. Domenico Giustiniano, my advisor, for taking me in and giving me the amazing opportunity to start my PhD journey in your group. The experiences of the past five years will forever remain with me and it was in great part thanks to you and the Pervasive Wireless Team: Roberto, Héctor, Ander, Maurizio, Alessio, Pippo, Tim, Stavros, Dayrene, Javi and Steven. Thanks for putting up with me all these years!

After COVID hit and we came back, the institute changed a lot, including the people. Fortunately, the environment and the good atmosphere remained the same. To all of you, thank you for these years, they have been absolutely unforgettable! But, of course, there are some who sticked a bit more, and to those I would like to extend my gratitude for all the great moments and experiences: Nina, Dolo, Jesús, Ori, Guillermo, and Rafa.

I would also like to thank the people from the CYD Campus in Switzerland, and in particular Vincent and Gérôme. You welcomed and helped me a lot during my two internships there and I learned many new things thanks to your guidance and support.

I cannot finish without thanking all my friends from Haritzeolo: Asier, Gaizka, Ignacio, The Iñigo's, Julen, Manu, Martin, Miguel, Mikel, Pablo, Sergio and Josue. It has been such a pleasure to grow up with you guys all these years. Hope we can stay together for many more years and more adventures.

To all the friends of Madrid who made living in the big city much more enjoyable and less stressful: Martin, Andrea, Asier, Paula, and Nico. I will never forget those days of beers and fun together.

And finally, to Onin. You appeared 3 years ago again in my life and changed it forever. I cannot thank you enough for all the love and support you provided throughout the hardships and joy and fun in the good times. I just wish you would allow me to stay with you as long as possible so that I can start repaying all that you gave me. I love you.

Published Content

The publications considered for this thesis consist of: 5 research articles published or submitted to peer-reviewed conferences and journals. The research items and the author's role in each of them are described below.

- **[1] Yago Lizarribar**, Roberto Calvo-Palomino, Alessio Scalingi, Giuseppe Santaromita, Gérôme Bovet, Domenico Giustiniano (2024, May). ORAN-Sense: Localizing Non-cooperative Transmitters with Spectrum Sensing and 5G O-RAN. *In IEEE International Conference on Computer Communications.*
  - This work is fully included in Chapter 3.
  - The author's role is focused on the design, implementation and experimentation of the proposed architecture. The main contributions for O-RAN have been made by the co-authors.
  - The material from this source included in this thesis is not singled out with typographic means and references.
- **[2] Yago Lizarribar**, Domenico Giustiniano, Gérôme Bovet, Vincent Lenders (2023). SkyPos: Real-world evaluation of self-positioning with aircraft signals for IoT devices. *IEEE Journal on Selected Areas in Communications.*
  - This work is fully included in Chapter 4.
  - The author's role is focused in the ideation of the architecture proposed in the paper, as well as the implementation and evaluation with real data.
  - The material from this source included in this thesis is not singled out with typographic means and references.
- **[3] Yago Lizarribar**, Alessio Scalingi, Domenico Giustiniano, Pedro Miguel Sánchez, Roberto Calvo-Palomino, Gérôme Bovet, Vincent Lenders (2024). Can't Hide Your Stride: Identifying Driver Patterns from a Network of Low-Cost Spectrum Receivers. *Under Review*
  - This work is fully included in Chapter 5
  - The author's role is focused in the design and implementation of the architecture mentioned in the paper, the development of the algorithms, and the experimentation and analysis.
  - The material from this source included in this thesis is not singled out with typographic means and references.

- **[4]** Ivan Palamà, Giuseppe Santaromita, **Yago Lizarribar**, Lorenzo Maria Monteforte, Stefania Bartoletti, Domenico Giustiniano, Giuseppe Bianchi, Nicola Blefari Melazzi (2023, June). From experiments to insights: A journey in 5G new radio localization. *In 2023 21st Mediterranean Communication and Computer Networking Conference (MedComNet) (pp. 74-82). IEEE.*
  - This work is partially included in Chapter 6
  - The author's role was focused on the experimentation analysis. Also took part in the writing of the whole experimentation section.
  - Whenever material from this source is included in this thesis, it is singled out with typographic means and an explicit reference.
- **[5]** Ivan Palamà, **Yago Lizarribar**, Lorenzo Maria Monteforte, Giuseppe Santaromita, Stefania Bartoletti, Domenico Giustiniano, Giuseppe Bianchi, Nicola Blefari Melazzi (2024). 5G Positioning with Software-Defined Radios. *Elsevier Computer Networks*
  - This work is fully included in Chapter 6
  - The authors role is on the algorithm optimization, building the deployment, as well as the localization infrastructure and experiment analysis. The author was also a co-leader in the full-paper writing.
  - The material from this source included in this thesis is not singled out with typographic means and references.

Other research merits

The publications not included in the thesis are the following:

- **[6] Yago Lizarribar** (2020). Signal Transmitter Localization using Low-cost SDR receivers (Master's thesis). *Carlos III University*
  - This work, although not part of the thesis, lays the foundations for Chapters 1-2 as well as the motivation for the research in Chapter 3.
  - The author's role was focused on the design and implementation of the concept proposed in this thesis.
- **[7]** Brecht Reynders, Franco Minucci, Erma Perenda, Hazem Sallouha, Roberto Calvo-Palomino, **Yago Lizarribar**, Markus Fuchs, Matthias Schäfer, Markus Engel, Berthold Van den Bergh, Sofie Pollin, Domenico Giustiniano, Gérôme Bovet, Vincent Lenders (2020, June). Skysense: Terrestrial and aerial spectrum use analysed using lightweight sensing technology with weather balloons. *In Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services (pp. 352-363).*
  - The author's role was focused on collecting data on cellular, and digital radio/television technologies and its analysis. Also took part in the writing of the paper.
- **[8]** Maurizio Rea, Domenico Giustiniano, Pablo Jimenez Mateo, **Yago Lizarribar**, Joerg Widmer (2021). Beam searching for mmWave networks with sub-6 GHz WiFi and inertial sensors inputs: An experimental study. *Computer Networks, 198, 108344.*
  - The author's role was focused on experiments with mobile-phone platforms

Abstract

Electromagnetic Radiation is the effect that Electromagnetic fields can exert at a distance from their source. Depending on the frequency of this radiation, we can divide it into categories such as: Radio-Frequency (RF), Infrared, Visible Light, X-rays or Gamma rays. In particular, the RF spectrum is the foundation for all modern wireless communications. Our lives would be difficult to imagine should we not have access to cellular, GPS, WiFi, Bluetooth, IoT and many other technologies. Although the RF spectrum is heavily regulated, current advances and the availability of hardware make it easier than ever to disrupt legitimate communications. In fact, at the beginning of the decade 2020, attacks on the spectrum are becoming more and more common. Currently, spectrum regulators must deploy bulky and expensive hardware to monitor these threats, which does not scale due to the amount of manual labor required.

Software-Defined Radio (SDR) devices are a flexible alternative for monitoring the RF spectrum due to how easy it is to reconfigure their measuring parameters like sampling rate or center frequency in real-time. This flexibility has allowed researchers and engineers to apply them in numerous applications. It is possible to perform low-level tasks like signal identification, localization or even self-positioning, up to building car tracking networks via tire transmissions or 5G standard compliant core networks. Not all SDR receivers can perform all tasks, as lower cost ones tend to have limitations in their hardware, but each category of device can enable a wide range of applications nonetheless.

In this thesis, we present four applications for localizing and identifying wireless transmissions that can be achieved taking advantage of the flexibility offered by SDR receivers. The first three demonstrate that with low-cost receivers it is possible to perform a broad range of activities. The first application is a transmitter localization system that can be deployed at the city level and achieves notable accuracy without the need for additional hardware for synchronization. The second application shows that it is possible to build a self-positioning architecture that employs aircraft broadcast signals and the metadata contained within. We show that by careful modeling of the hardware and precise selection of signals, it is possible to provide results comparable to Global Positioning System (GPS) or other commercial solutions, without the need for it. The last of this set explores a higher-level application, where we build a car tracking infrastructure based on capturing the signals transmitted by the pressure sensors installed on wheels. We then show that with careful processing it is possible to robustly track cars over long periods of time and

infer driver patterns from this tracking. Our results not only show the feasibility of building monitoring networks with low-cost receivers but also highlight the urge to redesign less privacy-aware protocols in this and many other areas.

The fourth application makes use of higher-end SDR receivers that support better synchronization hardware to build a custom 5G compliant network. With this network, we are able to explore the localization performance that these higher-end platforms can offer, after doing several modification to the underlying platform. Our extensive experiments with different geometric configurations, number of sensors, optimization routines and processing techniques show that it is possible to achieve highly accurate positions.

This thesis makes several contributions to the proposed architectures for all applications, as well as the techniques employed. In order to validate our proposals, we perform an extensive analysis with data coming from real-world scenarios.

Contents

Acknowledgements **VII**

Published Content **IX**

Other Research Merits **XI**

Abstract **XIII**

Table of Contents **XIV**

List of Figures **XXI**

List of Tables **XXV**

List of Acronyms **XXVII**

I Background **1**

1 Introduction: Wireless Signals **3**

1.1 The Ubiquity of Wireless Communications 3

1.1.1 How the spectrum is regulated 3

1.1.2 Threats to the RF spectrum and its users 6

1.2 Detecting and Capturing Wireless Signals 7

1.2.1 Types of SDR Receivers 7

1.2.2 Applications of SDR 8

1.3 Crowdsensing platforms 9

1.3.1 Electrosense 10

1.3.2 Why do these platforms fail . . . 11
1.4 Custom 4G/5G deployments . . . 11
1.5 Challenges of this Thesis . . . 12
1.6 Contributions of this Thesis . . . 13
1.6.1 Published Research . . . 13
1.6.2 Public Software . . . 15
1.7 Outline of the Thesis . . . 15

2 Positioning for Wireless Signals 17
2.1 Main Positioning Families . . . 17
2.1.1 Received Signal Strength . . . 18
2.1.2 Angle of Arrival . . . 19
2.1.3 Time of Arrival . . . 19
2.1.4 Time Difference of Arrival . . . 20
2.1.5 Other Methods . . . 20
2.2 Time Difference of Arrival . . . 21
2.2.1 Closed-Form Solution . . . 22
2.2.2 Least Squares . . . 25
2.2.3 Non-Linear Methods for TDOA . . . 26
2.3 Measuring the Accuracy of TDOA Systems . . . 27
2.3.1 Cramér-Rao Lower Bound . . . 27
2.3.2 Dilution of Precision . . . 28
2.3.3 What are Good Geometries? . . . 29

II Low-cost SDR in low-level applications 31

3 ORAN-Sense: Localizing Non-cooperative Transmitters with Spectrum Sensing and ORAN 33
3.1 Introduction . . . 33
3.2 Related work . . . 35
3.3 O-RAN Primer: Components and Interfaces . . . 36
3.3.1 Components . . . 36

3.3.2 E2 Interfaces, E2 Node and Service Model 37
3.4 ORAN-Sense 37
3.4.1 Frequency Offset Correction 39
3.4.2 Synchronization and TDOA Estimation 40
3.4.3 Multipath mitigation 42
3.4.4 Multilateration 43
3.4.5 Localization embedded into ORAN 44
3.5 Evaluation 44
3.5.1 Reference Signal Suitability 45
3.5.2 Deployments 48
3.6 Conclusion 51

4 SkyPos: Real-world evaluation of self-positioning with aircraft signals for IoT devices 53

4.1 Introduction 53
4.2 Related Work 55
4.3 Challenges 56
4.4 ADS-B for positioning information 57
4.5 Methodology 59
4.5.1 User Localization 60
4.5.2 Reference Receiver Offset Estimation 61
4.5.3 Dilution of Precision (DOP) 62
4.6 SkyPos: Self-Positioning for IoT devices 63
4.6.1 Backend 63
4.6.2 User 66
4.7 Evaluation 66
4.7.1 Dataset 67
4.7.2 Evaluation of the dataset 67
4.7.3 Intuition on Aircraft Selection 73
4.7.4 Architecture performance 75
4.8 Discussion 77
4.9 Conclusion 78

III Low-cost SDR in high-level applications 79

5 Identifying Driver Patterns with Low-Cost Spectrum Receivers 81

5.1 Introduction . . . 81
5.2 Related work . . . 84
5.3 Primer on TPMS systems . . . 85
5.4 Threats of a TPMS surveillance system . . . 87
5.4.1 Threat Model . . . 88
5.4.2 Tracking Car Owners . . . 88
5.5 Deployment . . . 89
5.5.1 Data collection setup . . . 89
5.5.2 Feasibility . . . 90
5.6 Dataset . . . 91
5.6.1 Receiver Sites . . . 91
5.6.2 Recorded data . . . 92
5.6.3 Ethical Considerations . . . 93
5.7 Car inference from tires . . . 94
5.7.1 Why Car Matching . . . 94
5.7.2 Aggregation based on cross-correlation . . . 95
5.7.3 Car aggregation based on Jaccard index . . . 97
5.7.4 Robust Car Matching . . . 100
5.8 Privacy Analysis Results . . . 102
5.8.1 Temporal Trends . . . 103
5.8.2 Analysis of individual cars . . . 104
5.9 Discussion . . . 107
5.10 Conclusion . . . 108

IV High-end SDR in low-level applications 111

6 5G Positioning with SDR radios 113

6.1 Introduction . . . 113
6.2 5G Positioning: State of the Art . . . 114

6.2.1 5G Positioning Architecture . . . 115
6.2.2 Mobile Network Synchronization Solutions . . . 117
6.2.3 Open-source platforms . . . 118
6.3 System Model . . . 119
6.3.1 Comparative Analysis of PRS and SRS for Positioning . . . 119
6.3.2 TOA Estimation . . . 120
6.3.3 TDOA-Based Positioning Algorithm . . . 121
6.3.4 Optimization Model for Positioning . . . 121
6.3.5 Geometric Dilution of Precision . . . 122
6.4 Experimental Setup . . . 122
6.4.1 Mobile Network Software . . . 123
6.4.2 Hardware . . . 123
6.4.3 Signal Processing Techniques and Localization Algorithms . . . 124
6.4.4 Network configuration . . . 126
6.5 Experimental validation . . . 126
6.5.1 Deployment Scenario . . . 127
6.5.2 Moving Average Experimentation . . . 128
6.5.3 Oversampling improvement . . . 128
6.5.4 Network Configuration Experimentation . . . 130
6.5.5 Localization Algorithms . . . 131
6.5.6 LOS vs NLOS scenarios . . . 132
6.5.7 Computational cost and Time to First Fix . . . 133
6.6 Conclusion . . . 134

V Last Remarks **137**

7 Conclusions **139**

Bibliography **141**

Appendices **157**

A Communication Activities **159**

B IRB Protocol **161**

List of Figures

1.1 Global mobile network traffic 4
1.2 FCC Spectrum allocation chart 5
1.3 Attacks on GPS Network 6
1.4 Errors in different SDR receivers 8
1.5 SDR Applications 9
1.6 Electrosense network high-level overview [15]. 10

2.1 Different scenarios when solving TDOA 24
2.2 Closed-Form solution for TDOA 25
2.3 CRLB of 4 configurations. 29

3.1 ORAN-Sense Architecture 38
3.2 Sensor Elements 39
3.3 Effect of multipath on cross-correlation 43
3.4 Feasibility study results 45
3.5 Splitter setup for feasibility tests 46
3.6 Sensor locations for the benchmark 48
3.7 Results in real environments 49
3.8 Maps of results in Madrid and Zurich 52

4.1 General overview of SkyPos 54
4.2 Structure of an ADS-B message frame 57
4.3 Number of distinct aircraft received 59
4.4 SkyPos proposed architecture overview. 64
4.5 TDOA table 65

4.6 Sensor locations across Europe 67
4.7 Reference receiver clock offsets 68
4.8 Accuracy with offset 69
4.9 DOP visualization 70
4.10 Error varying message rate and time windows. 71
4.11 Time and accuracy varying the amount of data. 71
4.12 Results for dump1090 sensors. 71
4.13 Error with respect to drift and message count. 72
4.14 Normal vs Robust Estimation of the residuals. 72
4.15 ECDF of the error using normal vs robust initialization 73
4.16 Measured aircraft in an hour 74
4.17 Results of sectorizing 76

5.1 Conceptual overview of a TPMS surveillance system. 82
5.2 dTPMS Sensor components. 86
5.3 TPMS sensor surveillance architecture. 87
5.4 Selected receiver and deployment site. 89
5.5 TPMS measurement characteristics for selected locations. 92
5.6 Tracking capacity with number of IDs captured 94
5.7 Correlation of messages between 3 tires. 96
5.8 Correlation matrices with 1 minute aggregation. 97
5.9 Detected cars when using 1 sensor vs 5. 98
5.10 Jaccard matrix for all verified IDs. Aggregation of 30 seconds. 100
5.11 Temporal pattern analysis 103
5.12 Weekly patterns of different worker profiles 105
5.13 Tire pressure of the selected car. 106
5.14 Probability of no message collision given a number of cars on the road. . 108

6.1 3GPP 5G Positioning architecture 115
6.2 Mobile network synchronization for 5G networks. 118
6.3 Example network deployment with 3 gNBs 119
6.4 Experiment setup 127

6.5 CDF for different window sizes 128
6.6 CDF when using different bandwidths 129
6.7 Impact of oversampling 129
6.8 CDF for different optimization algorithms 130
6.9 HDOP of the experimental setup 131
6.10 Error distribution at different bandwidths and windows 132
6.11 CDF when adding errors to the TOA values 133
6.12 Time to obtain position vs. number of nodes 134

List of Tables

1.1 Popular SDR receivers 7
1.2 Public software developed during the thesis. 15

2.1 Ranged-based localization methods 18

3.1 Correlation methods studied in this chapter. 41
3.2 Feasibility study experiment setup 46

4.1 Example of decoded ADS-B message 58
4.2 Sensor dataset structure 65
4.3 Aircraft dataset structure 65
4.4 Predicted positions for verified sensors 73
4.5 Time to execute different components of the proposed system. 76

5.1 TPMS information. 90
5.2 Approximated receiver sites, their availability and generated data. 91
5.3 Jaccard distance with the sensors related to the ID of C4.FL. 98
5.4 Jaccard indices for Car 9 99
5.5 Cross-Correlation vs Jaccard 102

List of Acronyms

3GPP	Third Generation Partnership Project	12
5GCN	5G Core Network	12
ADS-B	Automatic Dependent Surveillance-Broadcast	5
AM	Amplitude Modulation	4
AMF	Access and Mobility Management Function	115
AOA	Angle of Arrival	18
CIR	Channel Impulse Response	12
CFO	Carrier Frequency Offset	8
CRLB	Cramér-Rao Lower Bound	27
CSI	Channel State Information	21
CU/DU	Centralized Unit/Distributed Unit	12
DAB	Digital Audio Broadcasting	4
DL	Downlink	11
DL-AOD	Downlink Angle of Departure	116
DL-TDOA	Downlink Time Difference of Arrival	116
DOP	Dilution of Precision	28
dTPMS	Direct TPMS	85
DVB-T	Digital Video Broadcasting-Terrestrial	4
NR E-CID	NR Enhanced Cell ID	116
ECU	Electronic Control Unit	85
EM	Electromagnetic	3
EMR	Electromagnetic Radiation	3
ePRTC	enhanced Primary Reference Time Clock	117
E-SMLC	Enhanced Serving Mobile Location Centre	115
FDOA	Frequency Division of Arrival	21
FFT	Fast Fourier Transform	120
FIM	Fisher Information Matrix	27
FM	Frequency Modulation	4
GDOP	Geometric Dilution of Precision	28
GNSS	Global Navigation Satellite System	5
GPS	Global Positioning System	XIII
GPSDO	GPS Disciplined Oscillator	8
GSM	Global System for Mobile Communications	4

HDOP Horizontal Dilution of Precision . . . 74
ICAO International Civil Aviation Organization . . . 57
IDFT Inverse Discrete Fourier Transform . . . 12
IoT Internet of Things . . . 3
IQ In-Phase and Quadrature . . . 10
IRB Institutional Review Board . . . 93
IRLS Iteratively Reweighted Least Squares . . . 66
ISM Industrial Scientific and Medical . . . 5
iTPMS Indirect TPMS . . . 85
LCS LoCation Service . . . 116
LLS Linear Least Squares . . . 25
LM Levenberg-Marquardt . . . 61
LMF Location Management Function . . . 115
LOS Line of Sight . . . 88
LTE Long-Term Evolution . . . 4
mmWave millimeter Wave . . . 4
Multi-RTT Multi-Round Trip Time . . . 116
NG-RAN Next Generation Radio Access Network . . . 115
NG-eNB Next Generation evolved NodeB . . . 115
NLLS Non Linear Least Squares . . . 125
NLOS Non-Line of Sight . . . 42
NR New Radio . . . 113
Non-RT RIC Non-Real-Time RIC . . . 36
NTP Network Time Protocol . . . 40
OAI Open Air Interface . . . 11
OCXO Oven-Controlled Crystal Oscillator . . . 8
OEM Original Equipment Manufacturer . . . 90
ONF Open Networking Foundation . . . 12
ONOS Open Network Operating System . . . 12
O-RAN Open Radio Access Network . . . 13
OTA Over-The-Air . . . 114
PPM Parts Per Million . . . 8
PSD Power Spectral Density . . . 10
PRC Primary Reference Clock . . . 117
PRS Positioning Reference Signal . . . 11
PSR Primary Surveillance Radar . . . 57
PTP Precision Time Protocol . . . 118
QoS Quality of Service . . . 116
RAN Radio Access Network . . . 33
RAT Radio Access Technology . . . 116
RF Radio-Frequency . . . XIII
RIC RAN Intelligent Controller . . . 12

RTOA Relative Time of Arrival . . . 117
Near-RT RIC Near-Real-Time RIC . . . 36
RTT Round Trip Time . . . 117
RS Reference Signal . . . 40
RSS Received Signal Strength . . . 18
RSTD Reference Signal Timing Difference . . . 116
SBC Single-Board Computer . . . 81
SMO Service Management and Orchestration Framework . . . 36
SDR Software-Defined Radio . . . XIII
SFO Sampling Frequency Offset . . . 8
SLP SUPL Location Platform . . . 115
SNR Signal-to-Noise Ratio . . . 28
SRS Sound Reference Signal . . . 11
SSR Secondary Surveillance Radar . . . 57
SyncE Synchronous Ethernet . . . 118
TCAS Traffic Collision Avoidance System . . . 5
TCXO Temperature Compensated Crystal Oscillator . . . 8
TDOA Time Difference of Arrival . . . 15
TOA Time of Arrival . . . 11
TPMS Tire Pressure Monitoring System . . . 16
TRP Transmission-Reception Points . . . 115
TBS Terrestrial Beacon System . . . 116
TTFF Time to First Fix . . . 113
USRP Universal Software Radio Peripheral . . . 19
UE User Equipment . . . 11
UL-TDOA Uplink Time Difference of Arrival . . . 116
UL-AOA Uplink Angle of Arrival . . . 116
UMTS Universal Mobile Telecommunications Service . . . 4
US Unknown/Target Signal . . . 41
xApp eXtended App . . . 12

PART I

BACKGROUND

1 Introduction: Wireless Signals

1.1 The Ubiquity of Wireless Communications

Electromagnetic Radiation (EMR) can occur across a wide range of frequencies called the Electromagnetic (EM) spectrum. Within the EM spectrum, there exist different categories according to the frequency they occupy: Radio-Frequency (RF), infrared, visible-light, X-rays and Gamma rays. Each category has a role in today's society with applications ranging from communications, healthcare or fundamental particle physics research.

In particular, the RF spectrum is the portion of the EM spectrum that occupies frequencies up to 3 THz as defined by ITU [9]. RF plays a crucial role in our lives, as it is the foundation of all modern wireless communications. In this thesis, we explore wireless communications from different perspectives, by developing techniques that allow for monitoring the different technologies present in each frequency band and then positioning the source of such signals.

It is difficult to imagine today's society if wireless communications did not exist. They are the basis of a myriad of communication technologies including: WiFi, cellular communications, radio / television, aviation navigation, Internet of Things (IoT), Bluetooth, or satellite communications, to name a few. With the advent of wireless devices and technologies, the wireless communication infrastructure is growing at a rapid pace. The greatest example of such a tendency is mobile communications [10], which reach hundreds of exabytes per month (Fig. 1.1). In addition, this trend does not seem to slow down, and future forecasts predict even more penetration of wireless technologies.

1.1.1 How the spectrum is regulated

The RF spectrum is heavily regulated and each use case has a set of frequencies allocated. Fig. 1.2 shows the spectrum allocation chart for the United States [11], but almost all countries and regions around the world have similar allocations defined.

The use cases defined by the FCC or similar regulatory bodies are very diverse, but

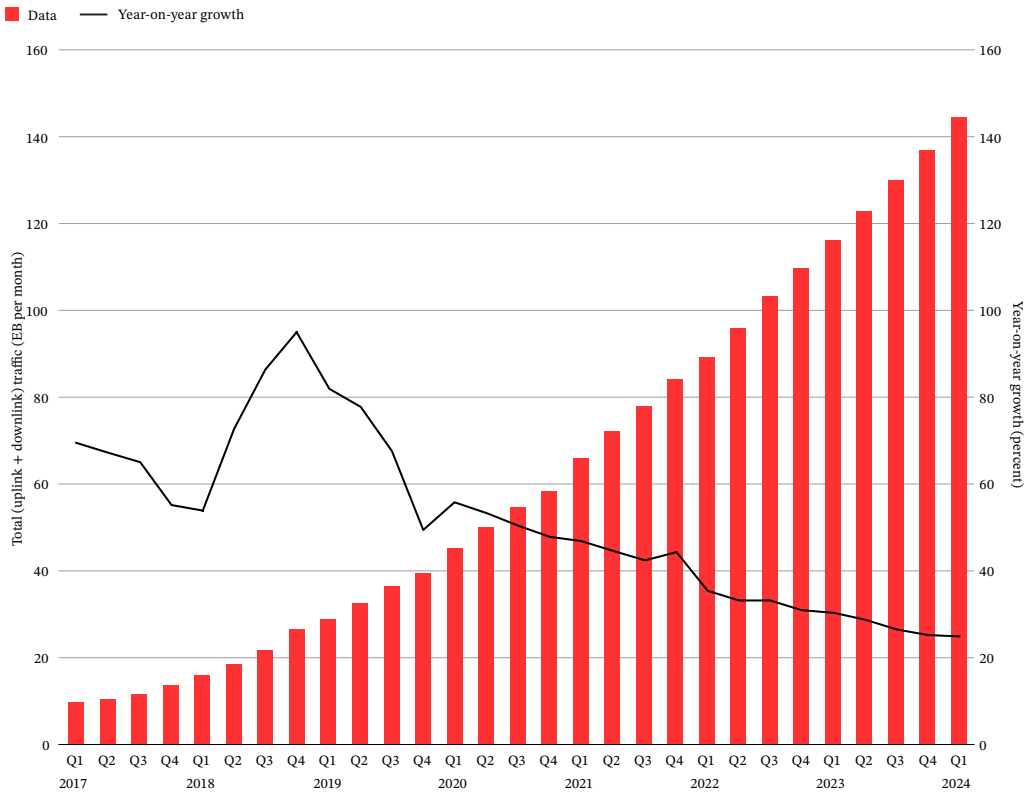


Figure 1.1: Global mobile network data traffic measured by Ericsson [10]

the most important categories for the vast majority of citizens are:

- **Cellular Communications**: Cellular communications are one of the most prominent uses of RF spectrum and span over multiple bands. Within cellular communications there are Global System for Mobile Communications (GSM), 3G/Universal Mobile Telecommunications Service (UMTS), 4G/Long-Term Evolution (LTE) and the relatively recent 5G NR. The channels span the 700-800 MHz bands, 2.1 GHz, 3.5 GHz and 6 GHz among others. There exist also large variations across countries or regions. In the past years, there has been development to bring cellular communications to millimeter Wave (mmWave) bands around the 24-39 GHz.
- **Broadcast Radio/Television**: One of the first uses of the RF spectrum for commercial applications were for radio (and later television) broadcasts. The bands from 3.3 to 26 MHz and 80-100 MHz are typically reserved for analog Amplitude Modulation (AM) and Frequency Modulation (FM) transmissions respectively, although many countries are phasing them out progressively. Today there are newer digital modes for radio and television, namely Digital Audio Broadcasting (DAB) and Digital Video Broadcasting-Terrestrial (DVB-T)2 standards in Europe for example. These technologies operate at higher frequencies as well. For instance, DAB uses the bands from 170-240 MHz, whereas DVB-T operates in the 470 to 790 MHz range.
- **Aeronautical communication**: Another important area of communications, although it does not impact individuals directly are aircraft communications. However, due to the ubiquity of air transport and travel it is worth mentioning. There are several proto-

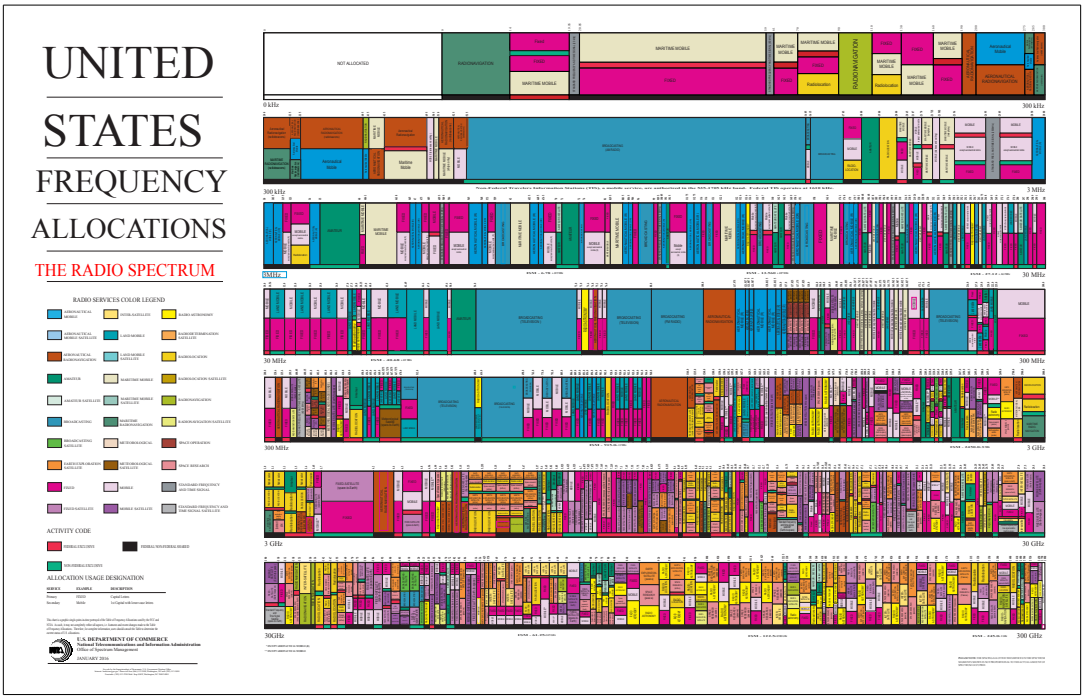


Figure 1.2: FCC Spectrum allocation chart [11].

cols used by aircraft, being some of the most important ones Traffic Collision Avoidance System (TCAS) and Automatic Dependent Surveillance-Broadcast (ADS-B). Both technologies operate at the beginning of what is known as the L-Band (1 to 2 GHz). Usually the 1030 MHz band is used for the uplink (ground stations to aircraft) and 1090 MHz for downlink. The latter is also where ADS-B transponders send their broadcast messages which is the basis for all aircraft tracking websites.

- **Satellite and Positioning**: Satellite communications use different parts of the spectrum for various tasks. On one hand we have traditional weather satellites which transmit in the bands around 140 MHz. One of the most popular use cases for satellite communications is positioning, through Global Navigation Satellite System (GNSS) networks. These networks operate in the L-band, with frequencies centered around 1200 and 1500 MHz. Another quite popular application of satellite communications this decade is "internet-over-satellite", being Starlink the most popular as of today. Their satellites operate within 10-11 GHz (downconversion is applied at the receiver side).
- **Unlicensed Bands**: Not all bands are under heavy regulation as the ones mentioned above, in fact there exist some frequencies at which lighter regulations are applied, which are for example the Industrial Scientific and Medical (ISM) bands. These can vary over countries and regions, but typical ones are the 434, 868, 915, 2400 or 5800 MHz bands. They are essential for everyday life as protocols like WiFi or Bluetooth are designed to work in these bands. The lower frequencies like 434 and 868 are also extremely popular for IoT protocols like LoRa, car key fobs, Smart-Home appliances and many others. These bands are usually limited by a duty cycle (each device can only transmit for a fraction of time) and a maximum transmit power, but access is not limited to particular brands or devices.

1.1.2 Threats to the RF spectrum and its users

Although the RF spectrum is under heavy regulation, today there is little knowledge of what is present in it at a given point in time. Currently, whenever a threat is suspected to occur, for example, a malicious actor disrupting 5G communications for a particular operator, the operator or government agencies of the country must go in person (or hire an external company) to the affected area and use bulky and expensive equipment to monitor and locate the source of such anomaly[1].

Furthermore, in recent years, attacks on the RF spectrum have become easier to deploy. The availability of more diverse hardware at lower prices, also combined with the unstable political situation in many regions in the beginning of the 2020s decade, has made attacks on the RF spectrum quite pervasive. From time to time, we see different attacks in the research literature, such as attacks on the LTE messaging infrastructure [12], or the almost eternal problem of GPS spoofing [13]. The latter is particularly damaging, and it can have serious consequences, not only for particular users (which might get lost or sent to malicious locations), but even to aviation industry. It is actually such a widespread problem that there exist websites that can track in almost real-time the GPS quality in different regions of the world (Fig. 1.3[2]).

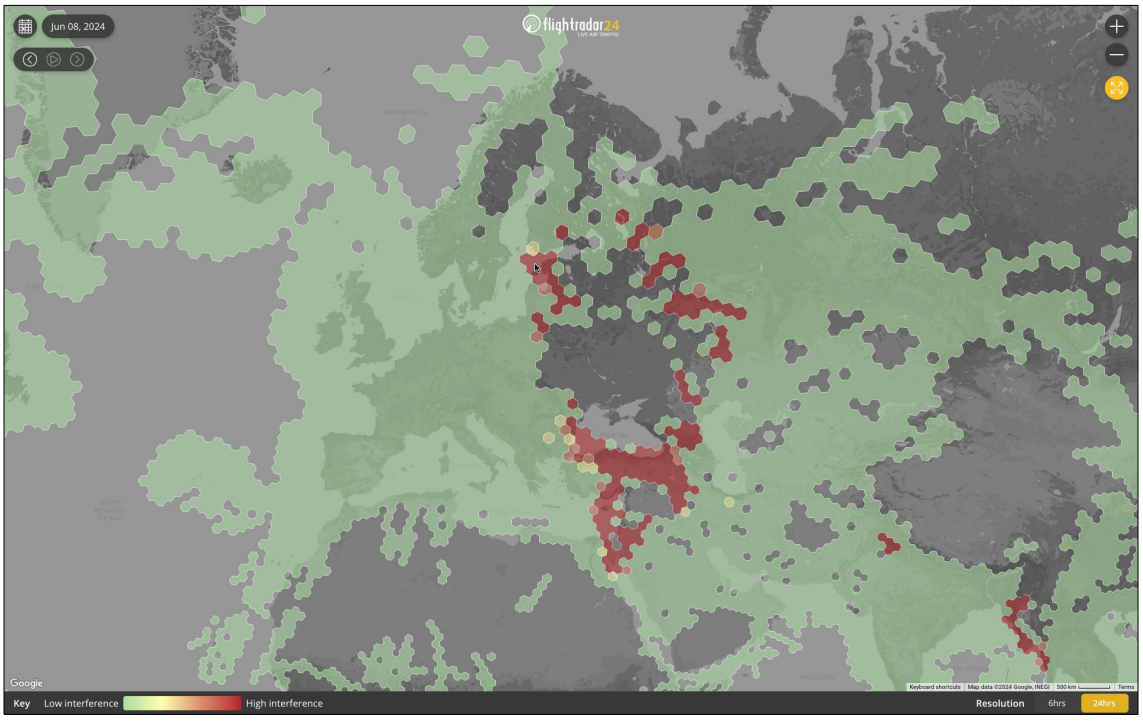


Figure 1.3: GPS quality degradation as reported by aircraft. Source: Flightradar24.

[1] https://spectrum.ieee.org/radio-frequency-hunters-saving-your-cell-signal
[2] https://www.flightradar24.com/data/gps-jamming

1.2 Detecting and Capturing Wireless Signals

Traditionally, devices that operate on the RF spectrum worked on a narrow portion of it. This is because many of the components and filters are designed for the specific bands that are monitored and thus lack flexibility. The radio or television systems in the home belong to this category. To monitor RF signals accurately over larger bandwidths, spectrum analyzers or oscilloscopes can be used. They tend to be costly and bulky, so they cannot be used to monitor the RF spectrum continuously over large areas.

However, since the late 1970s and early 1980s, the US Department of Defense developed a new type of device that quickly gained popularity, the Software-Defined Radio (SDR) [14]. These devices utilize an analog radio frontend to perform initial signal reception and frequency conversion. Subsequent signal processing, including filtering, demodulation, and decoding, is then executed digitally using software algorithms. Thus, they are extremely flexible and enable a wide range of applications. In the following sections, we describe the most typical commercial SDR receivers and different applications in which they are used.

1.2.1 Types of SDR Receivers

There exists a great variety of SDR receivers each with its own advantages and use cases. Table 1.1 shows five very popular SDRs and some of their characteristics. The RTL-SDR that is shown in the first two rows is an inexpensive dongle that is relatively popular in amateur radio communities but also in research. In fact, in this thesis we explain how we have used this inexpensive device to localize various signals, monitor the RF spectrum or even self-position itself.

The frequency range of most off-the-shelf SDR devices is quite similar. A key notion is that most SDR operate below the 6 GHz frequency. Currently, for larger frequencies, custom frontends or downconverters are needed that can handle them before reaching the digital side. The key differences between SDR receivers reside mostly in the maximum

SDR	**Cost**	**Frequency Range**	**Max. BW**	**PPM**	**Special HW**
RTL-SDR v1-2	15-20$	3 MHz - 1.7 GHz	2.4 MHz	$\pm$80	-
RTL-SDR v3-4	15-20$	3 MHz - 1.7 GHz	2.4 MHz	$\pm$1	TCXO
USRP B210	1300$	70 MHz - 6 GHz	60 MHz	$\pm$0.075	GPSDO
HackRF One	300$	1 MHz - 6 GHz	20 MHz	$\pm$0.5	TCXO
PlutoSDR	180$	325 MHz - 3.8 GHz	20 MHz	$\pm$25	-

Table 1.1: Popular SDR receivers and their maximum PPM errors according to their datasheet.

bandwidth that they can handle and their hardware imperfections. Larger bandwidths allow for more complex or higher-precision applications, which come at a higher cost.

Another important aspect to consider when selecting an SDR receiver is how its hardware imperfections are compensated. All hardware is subject to manufacturing errors and SDR receivers are not immune to it. One of the most common hardware imperfections in these devices are the Carrier Frequency Offset (CFO) and Sampling Frequency Offset (SFO) which are caused by the fact that the clocks in these devices do not measure time perfectly. Moreover, changes in temperature have a significant affect on the accuracy of clocks. Depending on the application, these errors might be tolerable (decoding), or must be corrected (high-precision timestamping for localization). In Fig. 1.4 we show the measured Parts Per Million (PPM) errors for different SDR receivers compared to an accurate source.

There exist multiple solutions to mitigate this. Lower-cost SDR receivers use a hardware component called Temperature Compensated Crystal Oscillator (TCXO), changing the clock frequency according to temperature changes. A more accurate version of these oscillators are Oven-Controlled Crystal Oscillator (OCXO), by ensuring the clock remains at a constant temperature. For high-precision timing applications, there exists another type of correction mechanism called GPS Disciplined Oscillator (GPSDO). These devices synchronize with the GPS network and reduce, for instance, timing errors to few nanoseconds.

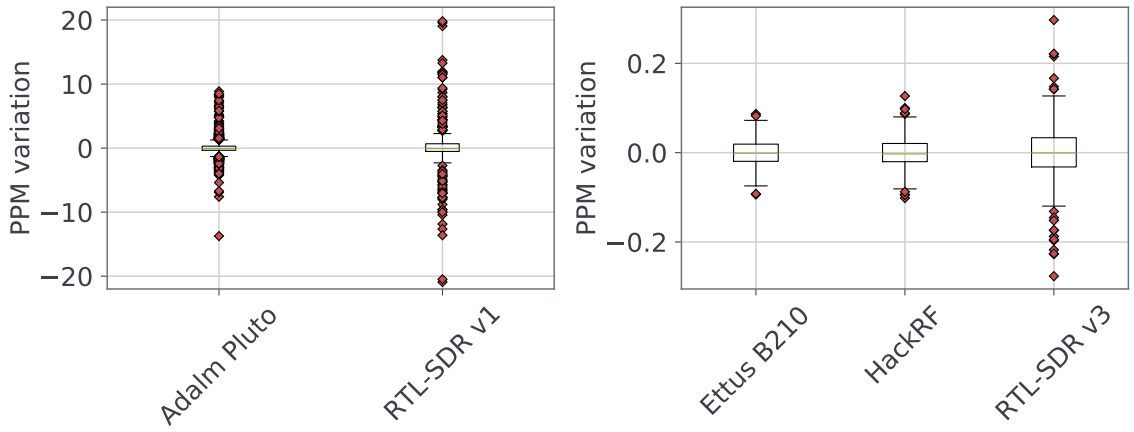


Figure 1.4: PPM measured by different SDR devices [1].

1.2.2 Applications of SDR

After talking about SDR types, one question remains: where can SDR devices be used? In Fig. 1.5 we trace a map where the horizontal axis represents the cost of SDR device and the vertical axis represents the application level. By low application level, we mean fundamental applications like positioning of wireless signals, whereas by high level, we mean tasks that are abstracted from the raw physical layer information captured by the SDR receivers.

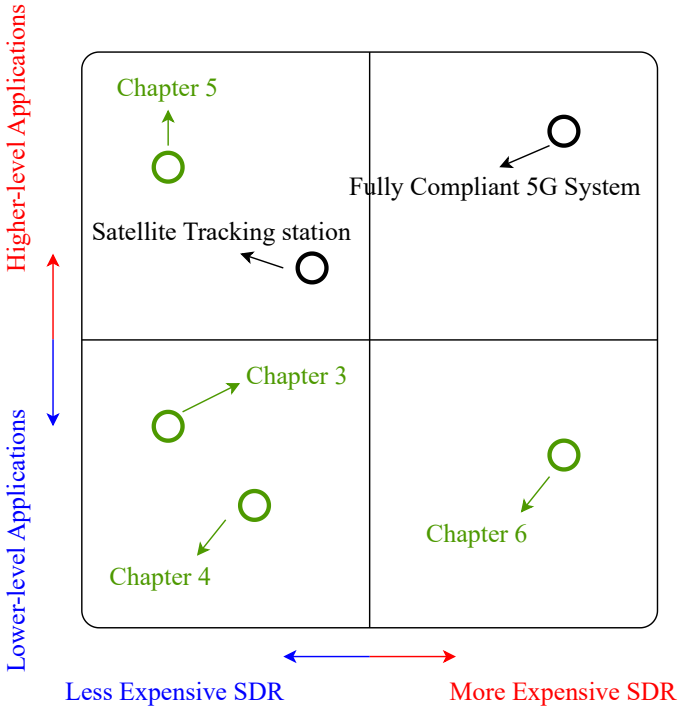


Figure 1.5: Applications of SDR depending on the cost. In green we mark where the publications of this thesis lay in this chart. In black, other types of applications.

For example, in Chapters 3,4 of this thesis, we explore the use of low-cost SDR receivers like the RTL-SDR shown in Table 1.1 to perform localization of wireless sources. In Chapter 5, however, we utilize the same sensors but we use them to demonstrate that it is possible to build a car tracking network, which is abstracted from the signal itself and relies on the metadata that can be decoded after capturing these signals. In Chapter 6 we use higher-end GPS-synchronized receivers to evaluate localization performance in a custom 5G deployment, which is again a lower-level application.

1.3 Crowdsensing platforms

In the advent of SDR receivers, the last decade also saw the birth of many crowdsourced spectrum monitoring solutions. Examples of this can be the Microsoft Spectrum Observatory or the Google Spectrum Database. Other platforms specifically geared towards a particular techonology also saw success. Particularly known are the aircraft tracking websites, like Flightradar24, Flightaware, or Opensky, which operate only on the 1090 MHz band, where aircraft broadcast their flight status information.

Part of this thesis was developed within the realm of another crowdsourced spectrum monitoring network: **Electrosense** [15], [16]. In this section, we will provide more details on this initiative.

1.3.1 Electrosense

Electrosense is a crowdsourced spectrum monitoring network (Fig. 1.6) [15], [16] that used the inexpensive RTL-SDR together with a Raspberry Pi, to monitor the spectrum mostly in the range from 30-1700 MHz. Later, an expansion board was developed that allowed the radio frontend to scan frequencies up to 6 GHz.

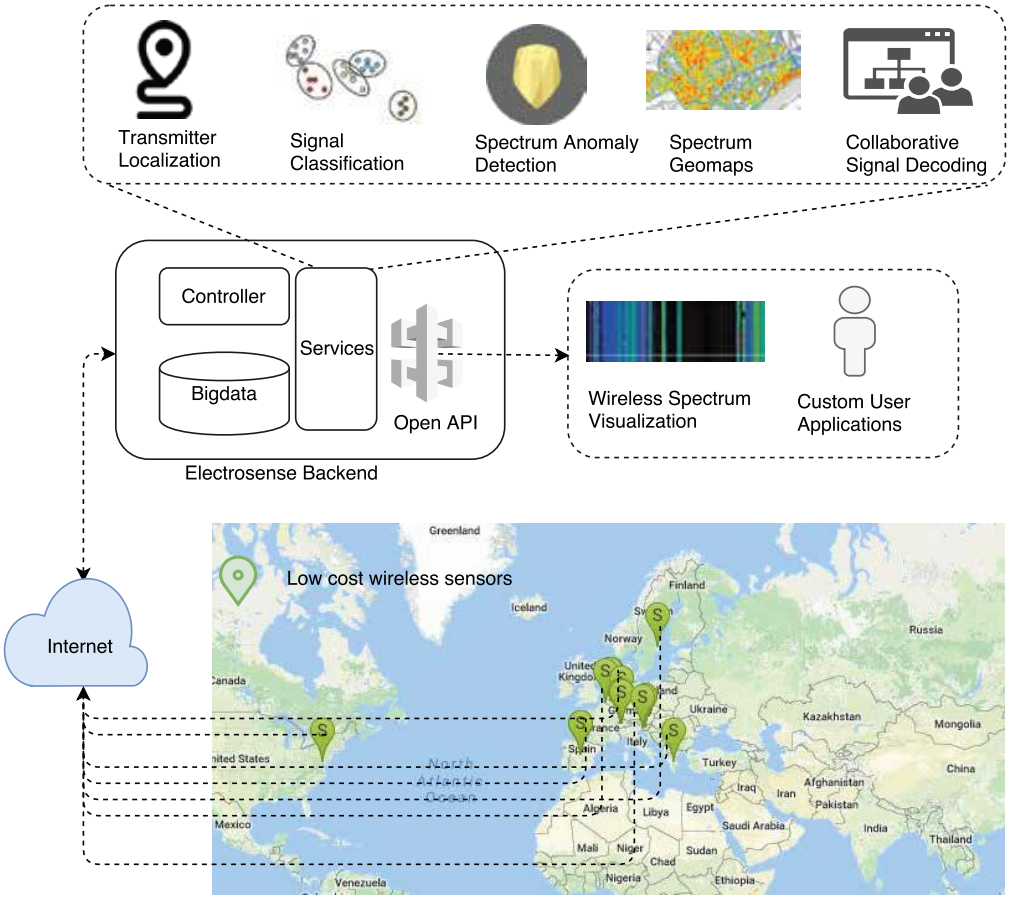


Figure 1.6: Electrosense network high-level overview [15].

The receivers could be installed indoors or outdoors and needed internet connection. These sensors then scanned the spectrum and sent the data to the backend in 2 forms: Power Spectral Density (PSD) or In-Phase and Quadrature (IQ). By default, aggregated PSD data was sent to the backend to reduce the usage of the Internet bandwidth. The data were then processed and made available via a public API. Users could also access the data and visualize the spectrum in different parts of the world.

Together with this, the infrastructure supported adding different services for other tasks than visualization. An example was that sensors could be coordinated to collect raw data and then perform transmitter localization. Another was a live spectrum decoder, where data was sent via WebRTC to the user's browser, and then different decoders could be applied to the incoming spectrum samples.

1.3.2 Why do these platforms fail

Even though some platforms still remain, especially aircraft tracking ones, many of the other spectrum sensing platforms do not operate anymore at the time of writing. There are several factors that play a role, which also lead to the downfall of our Electrosense platform.

First, there is the lack of incentives for users to install and maintain receivers. Except in aviation, spectrum sensing is not an interesting topic for the majority of citizens, and thus it becomes difficult to reach the necessary density of sensors to properly monitor the spectrum.

The second issue is the quality of the data, which comes from the installation of sensors. Users often install these sensors in places where they will not be seen or near routers, which then affects the quality of the measurements. This then has a greater impact for applications that require high data quality and precision, such as transmitter geolocation.

The last of the reasons are the development and maintenance costs. These platforms produce large amounts of data every day that must later be stored and processed. Although big data solutions exist, they require constant maintenance and supervision even for just a few tens of sensors, dramatically increasing operational costs. For these reasons, many crowdsourced spectrum monitoring solutions are out of operation.

1.4 Custom 4G/5G deployments

Linked to the availability of a plethora of SDR devices, creating custom 4G/5G and beyond deployments is relatively easy nowadays. To emulate the most heavily loaded components of these networks (like gNBs), high-end SDR receivers are needed. But for elements like User Equipment (UE), devices such as the Ettus USRP B210 series can be used. Overall, these advancements have made possible for many researchers to have localized setups to test their ideas. In the following lines we describe the most popular Open-Source libraries that implement parts of a compliant 5G Network.

OpenAirInterface: Open Air Interface (OAI) provides NR Release 15/16 compliant 5G Core, gNB and UE implementations. Written in C for real-time performance, OAI is distributed under the OAI Public License. This license allows free redistribution, use, and modification of source and object code for academic and research purposes, but it mandates Fair, Reasonable, and Non-Discriminatory (FRAND) negotiations for commercial or non-research uses, differing from Open Source Initiative-approved licenses in terms of unrestricted use. OAI is compatible with Intel and ARM architectures running Linux distributions and supports the most commonly used SDR platforms (e.g., USRP and LimeSDR). In recent times, OpenAirInterface has aligned with O-RAN standards through FlexRIC integration and implemented both Downlink (DL), Positioning Reference Signal (PRS) and UL Sound Reference Signal (SRS) positioning signals, providing Time of

Arrival (TOA) estimation based on Channel Impulse Response (CIR) and finer TOA estimation using Inverse Discrete Fourier Transform (IDFT) with up to 16x oversampling for CIR. It allows receiving DL PRS from multiple gNBs synchronized via a GPSDO. This makes OAI the ideal platform to test 5G positioning.

srsRAN: srsRAN platform provides software implementations of the gNB and UE with functionality up to NR Release 15. It does not implement its 5G Core Network (5GCN) and instead relies on Open5GS, an open-source implementation of the 5G Core conforming to Third Generation Partnership Project (3GPP) Release 16 written in C language. srsRAN is written in C and C++ and is distributed under the GNU AGPLv3 license. The srsRAN platform comprises two projects: srsRAN 4G [17] and srsRAN Project [18]. The srsRAN 4G project provides an end-to-end 4G and 5G NSA network implementation and a prototyping implementation of 5G SA UE and gNB. Instead, srsRAN Project is an ORAN-native 5G Centralized Unit/Distributed Unit (CU/DU) solution compliant with 3GPP and O-RAN Alliance specifications. srsRAN is the most similar platform to OAI; in fact, it is compatible with Linux distributions and supports the most popular SDR platforms. But, unlike OAI, where development is carried out daily by multiple branches, in srsRAN the release of new features is cadenced approximately semiannually; this makes srsRAN a relatively more stable project than OAI, but offers fewer features, including 5G positioning signal support. In fact, currently, srsRAN does not support positioning signals and protocols.

Aether: Aether is a project promoted by the Open Networking Foundation (ONF) for the implementation of private cellular networks. Unlike OAI and srsRAN, Aether not only offers an implementation of RAN and 5GCN, integrating SD-RAN [19] and SD-CORE [20], respectively, but also provides a control and orchestration interface for RAN, a cloud edge platform with support for cloud computing and a central cloud. The source code is released under an open-source Apache 2.0 license. The main goal of SD-RAN is to provide an O-RAN-compliant, near-real-time RAN Intelligent Controller (RIC) based on micro-ONOS, a cloud-native variant of the Open Network Operating System (ONOS), and an O-RAN-compliant platform for eXtended App (xApp). SD-RAN is based on OAI gNB and UE implementations. Since Aether's RAN is based on OAI without adding additional positioning support and we were not interested in the O-RAN extra functionalities, we decided to use OAI directly in our experimental setup.

1.5 Challenges of this Thesis

Monitoring and localizing signals with a variety of hardware and frameworks have several challenges. On the one hand, lower-cost hardware requires many corrections to enable precise applications. Higher-cost hardware solves many issues, but there is little knowledge of what the limits are. In the following lines, we describe the most notable challenges of this thesis:

- **Signal localization with low-cost spectrum receivers**: Low-cost hardware has many limitations in terms of bandwidth, frequency, dynamic range, and computational power among others. Also, even though they would be easier to integrate in crowdsourced networks, these are not always viable so an alternative is needed.
- **How to achieve accurate positioning with no satellites**: Another issue that low-cost hardware must face is the lack of hardware for accurate positioning. Or GNSS systems are under attack and the reported positions are not accurate. In such cases, what possibilities are there to maintain knowledge of one's positions?
- **What is feasible to monitor with low-cost sensors**: We have described how spectrum sensor networks could be used to monitor RF signals. But what kind of signals can be reliably monitored? Is it feasible to establish such a network that could operate autonomously? Can we extract some information from the monitored signals and track particular transmitters? What are the privacy implications of this?
- **What are the limits of custom 4G/5G deployments**: Several custom frameworks have emerged recently that are able to replicate 4G/5G core networks. Localization is an essential feature of cellular networks, but it is unclear the accuracy these custom deployments can achieve. Also, what physical parameters can we measure in order to obtain the most accurate position estimation.

1.6 Contributions of this Thesis

1.6.1 Published Research

The main contributions of this thesis have been published in five articles. Two of them have been published in the **IEEE Journal on Selected Areas in Communications (JSAC)** and **Elsevier Computer Networks**, both indexed in Journal Citation Reports (JCR). Another publication was published in a tier-1 conference, **IEEE INFOCOM**. Other article was published in the **Mediterranean Communication and Computer Networking Conference** as the groundwork for the journal article. The last of the articles is under review.

Contribution 1: *Transmitter localization with low-cost sensors and ORAN.* A framework that uses low-cost spectrum sensors to localize transmitters with notable precision [1]. To overcome the challenges posed by maintaining spectrum sensor networks, we also propose to use the interfaces defined by Open Radio Access Network (O-RAN), so that the processing can be integrated within the network operator, thus reducing costs. Using low-cost sensors also has challenges, as hardware impairments require several corrections to be made. Our architecture is able to correct and estimate the position within the time limits established by the O-RAN specifications. To validate our architecture, we deploy sensors in 2 cities and localize real signals from both.

- **Yago Lizarribar**, Roberto Calvo-Palomino, Alessio Scalingi, Giuseppe Santaromita, Gérôme Bovet, Domenico Giustiniano (2024, May). ORAN-Sense: Localizing Non-

cooperative Transmitters with Spectrum Sensing and 5G O-RAN. *In IEEE International Conference on Computer Communications.*

Contribution 2: *Self-positioning for low-cost sensors using aircraft signals.* We propose, implement and validate an architecture that collects aircraft data and then provides location services to low-cost sensors [2]. By opportunistically using aircraft broadcast metadata, with aircraft tracking sensor networks, we can build a service that collects this information and is made available for other sensors to compute their own position. Due to the low-cost nature of the sensors, we have to account for their defects in our model. Another important point is the selection of aircraft for better accuracy and efficiency. We build the architecture and test it with real data from the OpenSky network. This work was done in collaboration with *armasuisse Science and Technology*.

- **Yago Lizarribar**, Domenico Giustiniano, Gérôme Bovet, Vincent Lenders (2023). SkyPos: Real-world evaluation of self-positioning with aircraft signals for IoT devices. *IEEE Journal on Selected Areas in Communications.*

Contribution 3: *Vehicle tracking and driver pattern reconstruction through tire pressure signals.* We implement a vehicle tracking network that captures the signals sent by the tire pressure sensors inside cars [3]. From these transmissions, we can extract unique identifiers together with other tire parameters to accurately identify individual cars. We collect data over several months and show that it is possible to extract driver patterns during this period, highlighting the privacy risks associated with this technology. We also develop an algorithm that can robustly track cars, further outlining the need for a redesign in these sensors.

- **[3] Yago Lizarribar**, Alessio Scalingi, Domenico Giustiniano, Pedro Miguel Sánchez, Roberto Calvo-Palomino, Gérôme Bovet, Vincent Lenders (2024). Can't Hide Your Stride: Identifying Driver Patterns from a Network of Low-Cost Spectrum Receivers. Under review.

Contribution 4: Positioning with SDR radios in 5G networks. We deploy a custom 5G network and evaluate the positioning accuracy that can be obtained with it. To achieve this, we modify one popular 5G compliant framework to output the timing information. From there, we are able to perform difference of arrival positioning. We test different gNB configurations, positions and optimization algorithms, and provide insights into the computational cost, as well as the effect of geometry or non-line of sight conditions into the accuracy of such systems.

- Ivan Palamà, Giuseppe Santaromita, **Yago Lizarribar**, Lorenzo Maria Monteforte, Stefania Bartoletti, Domenico Giustiniano, Giuseppe Bianchi, Nicola Blefari Melazzi (2023, June). From experiments to insights: A journey in 5G new radio localization. *In 2023 21st Mediterranean Communication and Computer Networking Conference (MedComNet) (pp. 74-82). IEEE.*
- Ivan Palamà, **Yago Lizarribar**, Lorenzo Maria Monteforte, Giuseppe Santaromita, Stefania Bartoletti, Domenico Giustiniano, Giuseppe Bianchi, Nicola Blefari Melazzi

(2024). 5G Positioning with Software-Defined Radios. *Elsevier, Computer Networks*

1.6.2 Public Software

During the thesis, each of the research articles that is part of the thesis also resulted in different software repositories, some of which have been made publicly available. Table 1.2 describes the public repositories created for each of the research articles included in the thesis, together with the license and chapter in which they were used. They can all be found under the GitHub username https://github.com/yagoliz:

Name	Description	Language(s)	License	Reference
localization-tdoa	Implementation of the Receiver-Target-Receiver localization method	Matlab	MIT	[1]
pytdoa	Python translation of localization-tdoa	Python	MIT	[1]
es_tcp	Implementation of the rtl-sdr tcp server to support the Electrosense downconverter	C++	MIT	[1]
skypos	Algorithms and Client for the SkyPos article	Matlab, Python	MIT	[2]
tpms-analysis[3]	Algorithms and processing for TPMS data	Python	MIT	[3]
5gpositioning	Localization algorithms and processing for the 5G Positioning with SDR article	Python	MIT	[4], [5]
tdoapp	C++ TDOA equation solver with CLI and webserver implementations	C++	MIT	[4], [5]

Table 1.2: Public software developed during the thesis.

1.7 Outline of the Thesis

Having set the groundwork for this thesis, in the following lines describe each of the chapters that compose this thesis. In total there are four parts to this thesis. In Part I, we provide the reader with the necessary context to understand the rest of the document. Apart from this chapter, it also includes Chapter 2:

Chapter 2 gives an overview of existing positioning methods in wireless systems together with their advantages and drawbacks. In later sections, the Time Difference of Arrival (TDOA) method is described, which is the basis of the methods employed in Chapters 3,4 and 6. We also provide insights into how the accuracy of these methods can be estimated, together with typical example geometries that might increase or decrease accuracy.

In the next part, Part II explores the applications of low-cost SDR receivers in low-level applications. In particular:

[3]It will be made public once article is published

Chapter 3 shows how low-cost SDR receivers from a crowdsourced network could be integrated into the O-RAN paradigm. We provide an architecture that takes advantage of the standardized interfaces in the O-RAN concept, to perform tasks like the localization of unauthorized transmitters. Our study validated the localization performance in 2 cities, showing notable accuracy for such low-cost sensors

Chapter 4 offers a variant into the problem where instead of localizing unauthorized transmitters, we opportunistically use aircraft signals to perform self-positioning. This approach is interesting for sensor networks that cannot rely on typical positioning techniques like GPS. We validate our methods with real crowdsourced sensors from the OpenSky Network, developing an architecture that could be fit within this type of platforms. We show that it is possible to obtain notable accuracy as well as provide the mathematical framework to estimate the accuracy of such system.

Part III shows an example of a higher-level application that can still be achieved with low-cost SDR receivers:

In **Chapter 5** we show the privacy risks that are associated with Tire Pressure Monitoring System (TPMS), a system designed to report the tire pressure information of a car to the central processing unit, which is done wirelessly for a large proportion of modern cars. Furthermore, data are sent with a unique identifier and are unencrypted, so anyone close enough might collect data from cars on the road and build a tracking infrastructure without anyone noticing. We show that with low-cost sensors and little processing, it is possible to track cars robustly and extract driver patterns over long periods of time.

In Part IV we apply the higher-end SDR devices to build a localization testbed:

Chapter 6 we explore the achievable localization accuracy when building a custom testbed using high-end SDR receivers. The testbed was built using one of the most popular open-source frameworks to deploy 5G systems, which we then modified to get the timing information. We later explore how different post-processing techniques and sophisticated optimization routines can increase the localization performance and reduce errors to the 1m level.

Finally, in Part V (**Chapter 7**), we provide the conclusions of this thesis and also explore future lines of work.

2 Positioning for Wireless Signals

Wireless signal positioning has numerous applications that permeate everyday life. A prime example of such applications is GPS (or GNSS like Galileo or GNLONASS), which most of us use in their day-to-day lives. In these cases, phones use satellite position information sent wirelessly to estimate their own position (and incidentally their users'). Many types of wireless technologies can also be used to achieve accurate positioning, specially in indoor settings where satellite-based positioning may fail [21]. The technologies used for such purpose can be WiFi, Bluetooth, Cellular Networks and many more. Wireless signal localization has also been proposed in search and rescue missions in challenging environments. In Chapter 1, we already mentioned that the RF Spectrum is a scarce resource and malicious actors might want to disrupt regular operations. Locating the sources of such disruptions is essential for spectrum regulators and governmental agencies to ensure the safety of all users.

The popularity and general availability of SDR devices has enabled the research community to test and build a myriad of localization schemes over the last years [5], [22]–[24]. A notable proportion of articles written in the past decade on localization take advantage of the flexibility of these devices, instead of using custom platforms.

There is not a 'silver bullet' method in the localization world. In this chapter, we analyze different positioning strategies present in the literature, what are their advantages, disadvantages and application scenarios. We also show the different SDR platforms that exist for each of the proposed methods. In the latter part of this chapter, we look in more detail the main positioning scheme used throughout this thesis: TDOA.

2.1 Main Positioning Families

Wireless signal positioning can be achieved by a variety of methods, which exploit different physical characteristics of the EM signals. We can distinguish two main categories: range-based and range-free methods. Range is generally tackled from four different perspectives, as it was summarized in [25]:

- **Received Signal Strength (RSS)**: The core of this method is to measure the **power received at different sensors**, and by means of a path loss (PL) model, estimate the distance to the transmitter. Sensors (RXs) need to be calibrated and a path-loss model is required to estimate the decay in power;
- **Angle of Arrival (AOA)**: It is based on the use of directionally sensitive antennas and requires specialized hardware. These antennas are then capable of measuring the **target signal's angle** of incidence.
- **Time of Arrival (TOA)**: In this technique, the **travelling time from the emitter to the different receivers is measured**, and then position is obtained by performing a multilateration. For this approach to work, an accurate time synchronization among sensors and transmitters is needed.
- **Time Difference of Arrival (TDOA)**: It is similar to the TOA technique, but measuring the **differences in arrival times of the signal to all the sensors**. Since only sensors need to be synchronized in this scenario, it fits the case of unknown transmitter localization.

Table 2.1 summarizes the main localization families, by showing the principle of operation of each, together with the main weaknesses of each approach.

2.1.1 Received Signal Strength

In recent years, great effort has been focused on RSS based systems [26], [27]. The situations in which these methods have been applied range from static transmitter localization with static sensors [23], [28] or mobile robots [29], distance sensing to avoid collisions between robots [30] or even wildlife tracking [31]. The main appeal of RSS based methods is the low hardware requirements [32], which makes an attractive use case for deployments in large sensor networks. However, these techniques have their drawbacks such as:

1. Sensors need to be calibrated;

Name	Principle of Operation	Weaknesses
RSS	Measured power of the signal at the receiver	Attenuation;
AOA	Angle of incidence of the signal on antenna	Low accuracy Number of antennas Receiver chains;
TOA	Measure time taken from transmission to reception	RX/TX Cooperation Synchronization
TDOA	Difference in when signal arrives at receivers	Synchronization

Table 2.1: Range-based localization methods and their most prominent features.

2. A path-loss model is needed to convert from power measurements to distances;

3. If there are multiple transmitters, an algorithm is needed to discriminate the contributions of each one, and they often have high time complexities [28];

4. It requires either knowing the transmitter power or a dense sensor network.

RSS based research being the most popular in the past years, it comes as no surprise that SDR receivers are widely employed in these papers. Such examples appear in [27], where they make use of Universal Software Radio Peripheral (USRP), Pluto and other high-end receivers. The aforementioned receivers offer a great deal of features like GPS disciplined oscillators and higher bandwidths, although their cost is higher. In [24], [31] we see examples of RSS based localization using the RTL-SDR although the covered distance is no more than 100 m, thus it cannot be considered a large scale deployment. Another example of RSS based system is the one proposed in [33]. However, this system again was validated in simulations and on a relatively small area, and no realistic environment was considered.

2.1.2 Angle of Arrival

AOA requires a receivers chain to compute the angle of incidence locally. The amount of information needed to perform localization is therefore less than with other methods [34]. It also has a notable accuracy both in theory [35] and in practice [36]. AOA methods require less receivers, which could be interesting in environments where receiver quantity was a constraint. Theoretically, in a 2D scenario only 2 receivers would be needed to estimate the position of the transmitter.

One of the main issues of AOA based methods is that in order to compute the angle of incidence, a special array of antennas is required. Depending on the frequency of interest, the footprint of the antenna array and its cost can be quite large. This requirement is quite challenging to fulfil when it comes to deploying embedded AOA sensors. Besides, the same angle error causes larger positioning errors at longer distance.

AOA based methods are typically used in millimeter-Wave (mmWave) settings, as transmitter/receiver chains are usually composed of multiple antennas in array formations. On the other hand, devices like the KrakenSDR[4] which consist of 5 inexpensive RTL-SDRs [37] sharing the same clock, make it possible to obtain angle measurements.

2.1.3 Time of Arrival

TOA has been widely employed as a localization method in sensor networks, since it provides accurate results and does not need to make assumptions on the signal [38]. The

[4]https://www.crowdsupply.com/krakenrf/krakensdr

general idea is to calculate the time at which the signal arrives at each sensor (hence the name TOA), compute the circumferences with radius equal to the distance between the transmitter and each receiver, and calculate their intersection by means of trilateration. TOA can be implemented using one-way transmissions, or with echo techniques [39]. In both cases, for the algorithm to work, it is necessary that the transmitter has an active role in the positioning process, thus making it unsuitable for the case of identifying and localizing rogue/unknown transmitters.

TOA based methods are the primary choice in GNSS. Satellites are equipped with high-precision Rubidium clocks, and stations on the ground with even higher-precision clocks, periodically send them corrections. This way, the whole network stays synchronized and provides meter-accuracy (or even better with more sophisticated techniques).

2.1.4 Time Difference of Arrival

TOA based positioning requires cooperation between all entities, so that a known waveform at a specified time is transmitted to the receivers. However, there are situations in which this might not be possible. Spectrum regulators might want to locate signal sources using technologies not designed to be localized. In other cases, the timing information might not be present [2], [40]. In such cases, instead of computing the time the transmission took from the transmitter to each receiver, we can measure the difference in arrival times at each receiver pair. This approach is what is called TDOA.

While TDOA has been shown to have similar accuracy as TOA [38], it also requires synchronization between receivers. This can be achieved via GNSS or with signals of opportunity [1]. In this thesis, we make extensive use of TDOA methods to achieve the desired positioning estimates. In the next section (2.2), we expand this methodology and provide different algorithms to tackle these problems.

2.1.5 Other Methods

The methods described above are by no means the only ones that appear in the literature. There exist situations in which different approaches might be equally or more accurate in obtaining position estimates. In this section, we show a few of those options.

Hybrid Methods

So far, we have described the methods separately, but it is not uncommon to see them used in tandem. The combination of TDOA and AOA is mentioned several times [34], [35]. For example, this approach would require fewer receivers than in a pure TDOA scenario.

We have not talked about the mobility of receivers or transmitters. All the methods described above assume static scenarios or only "snapshots" if the environment was more

dynamic. To overcome this, there exists a complementary method to TDOA, namely Frequency Division of Arrival (FDOA). The core idea behind this method is to compute the differences in Doppler shifts between different receivers. With this, it is possible to estimate not only the position but the heading and total speed of the transmitter.

However, this last method is not as accurate and is often used in hybrid systems with TDOA measurements. Examples of this can be found in [41], [42].

Fingerprinting Methods

Another family of methods involves the collection of Channel State Information (CSI) values at different points and perform a reverse look-up at the time of estimating the position [43]. The advantage of these methods is that they are usually fast, as the computation requirements are very low. However, these methods do not scale well in larger settings, as the number of measurements increases quadratically with the length of the grid. Therefore, these methods are usually used in smaller indoor environments.

2.2 Time Difference of Arrival

TDOA algorithms have been employed for a long time for various types of localization tasks. In theory, it has slightly less accuracy to TOA based localization approaches [38] but with the advantage of only needing synchronization between receivers, and it is therefore an interesting approach for the task of localizing unknown transmitters. After TDOA values are computed for each pair of receivers, the position of the transmitter is estimated by means of multilaterating, in a similar manner as in TOA.

The basic principle of TDOA is that a signal emitted by a transmitter and received by multiple receivers located in different positions arrive at different times due to the different distances between transmitter and receivers. After TDOA values are computed for each pair of receivers, the position of the transmitter is encountered at the intersection of hyperbolas generated by each pair of sensors.

The TDOA model is based on the TOA of a signal. When a signal is transmitted and captured by the receiver i, the model for the received signal time is as follows.

$$t_i = t_0 + \frac{||\mathbf{x}_i - \mathbf{x}_t||}{c} + \delta_i \cdot t + \theta_i + \eta_i = t_0 + \frac{d_i}{c} + \delta_i \cdot t + \theta_i + \eta_i \tag{2.1}$$

where t_0 is the time at which the original signal is transmitted, $\mathbf{x}_i$ and $\mathbf{x}_t$ are the position of the receiver and transmitter, respectively (distance that we can represent by d_i), δ_i and θ_i are the drift and time offset associated with the receiver i, and η_i is the noise. If the receivers are synchronized with each other, the offset is negligible. The drift can be estimated with various techniques, so we can remove it.

For each pair of receivers i, j, we can subtract Eq. 2.1 to obtain the TDOA between them:

$$\tau_{ij} = t_i - t_j = \frac{d_i}{c} - \frac{d_j}{c} + \eta_{ij} \tag{2.2}$$

Since the timings involved are usually on the order of nanoseconds (10^{-9}s), for numerical reasons, it is typical to multiply both sides by the speed of light c and to denote $d_{ij} = d_i - d_j$ as the distance difference between receivers. Thus, Eq.2.2 is also written as:

$$c \cdot \tau_{ij} = d_i - d_j + \mu_{ij} \tag{2.3}$$

which represents the measured distance difference between both receivers.

In the next sections, we show the different methods for obtaining a position utilizing different methodologies.

2.2.1 Closed-Form Solution

In this section, we describe the Closed-Form Solution [44], which shows how to obtain the coordinates of an unknown transmitter for the case of TDOA when there are 3 receivers in a 2D scenario or 4 in a 3D scenario. In these cases, the TDOA equations form a closed-form system, and the solution is unique.

Derivation

Given 3 sensors with the following positions:

$$\mathbf{x}_1 = [0\ 0]^T; \quad \mathbf{x}_2 = [b\ 0]^T; \quad \mathbf{x}_3 = [c_x\ c_y]^T$$

And for a transmitter whose position is $\mathbf{x}_t = [x\ y]^T$, we can write the TDOA equations for ab and ac pairs in the following form:

$$\tau_{12} = \sqrt{x^2 + y^2} - \sqrt{(x-b)^2 + y^2} \tag{2.4}$$

$$\tau_{13} = \sqrt{x^2 + y^2} - \sqrt{(x-c_x)^2 + (y-c_y)^2} \tag{2.5}$$

We can isolate the terms on the rightmost part of both equations and square both sides. After carefully operating, we can obtain the following forms:

$$\tau_{12}^2 - b^2 + 2bx = 2\tau_{ab}\sqrt{x^2 + y^2} \tag{2.6}$$

$$\tau_{13}^2 - c^2 + 2c_x x + 2c_y y = 2\tau_{13}\sqrt{x^2 + y^2} \tag{2.7}$$

Dividing 2.6 with 2.7 and operating on the terms, we are able to express the y coordinate in terms of x:

$$y = gx + h \tag{2.8}$$

Where g and h are:

$$g = \frac{\left(\frac{\tau_{13}}{\tau_{12}}\right) b - c_x}{c_y} \tag{2.9}$$

$$h = \frac{c^2 - \tau_{13}^2 + \tau_{12}\tau_{13}\left(1 - \left(\frac{b}{\tau_{12}}\right)^2\right)}{2c_y} \tag{2.10}$$

Introducing these values into Eq. 2.6, we are able to obtain the following expression for the x coordinate:

$$0 = dx^2 + ex + f \tag{2.11}$$

Where the values of d, e and f are:

$$d = -\left(1 + g^2 - \left(\frac{b}{\tau_{12}}\right)^2\right) \tag{2.12}$$

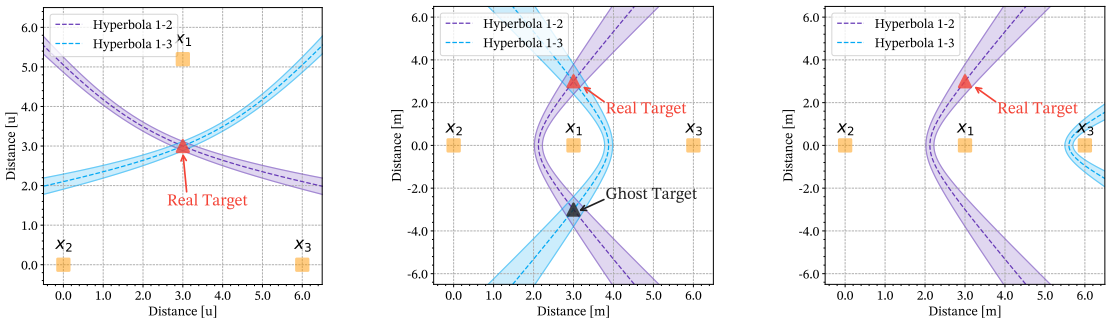
$$e = b\left(1 - \left(\frac{b}{\tau_{12}}\right)^2\right) - 2gh \tag{2.13}$$

$$f = \frac{\tau_{12}^2}{4}\left(1 - \left(\frac{b}{\tau_{12}}\right)^2\right)^2 - h^2 \tag{2.14}$$

After solving the quadratic equation, there are 3 possible scenarios:

- **One solution**: We will always obtain one solution, but there are cases where one of them gives a TDOA value with opposite sign to those that we had previously, so that one can be discarded.
- **Two solutions**: In other scenarios, it can happen that both solutions produce the same TDOA values as those measured and further information would be needed to discriminate the actual location of the transmitter.
- **No solution**: If the obtained values for x are imaginary, that means there is no intersection between the hyperbolas, and thus no location can be determined. This can happen if one of the receivers reports a wrong TOA value.

To better illustrate this, in Fig. 2.1 we show 3 examples with each of the cases described above.



(a) Unique solution (b) Dual Solution (c) No solution

Figure 2.1: Different scenarios when solving a TDOA system.

Basis changes

The above development takes the reference sensor **a** as the origin of the coordinates and places the sensor **b** on the x axis. In our case, we will generally have positions $\mathbf{x}_1 = [x_1\ y_1]^T$, $\mathbf{x}_2 = [x_2\ y_2]^T$ and $\mathbf{x}_3 = [x_3\ y_3]^T$. We will refer to $0'$ to the original reference system, 0 to the system with the same rotation but centered on sensor $\mathbf{s}_1$, and a to the system that has sensor $\mathbf{s}_2$ on its x axis. The rotation angle θ between systems 0 and a:

$$\theta = \arctan \frac{y_2 - y_1}{x_2 - x_1} \tag{2.15}$$

With this we can obtain the rotation matrix that changes vector from base a to 0

$${}^{\mathbf{0}}\mathbf{R}_{\mathbf{a}} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.16}$$

Thus we can obtain the coordinates for sensors b and c in this new reference system:

$${}^{a}\mathbf{x}_2 = {}^{0}\mathbf{R}_{a}^{T} \cdot (\mathbf{x}_2 - \mathbf{x}_1) \tag{2.17}$$

$${}^{a}\mathbf{x}_3 = {}^{0}\mathbf{R}_{a}^{T} \cdot (\mathbf{x}_3 - \mathbf{x}_1) \tag{2.18}$$

These new vectors ${}^{a}\mathbf{x}_2$ and ${}^{a}\mathbf{x}_3$ have the form required by the algorithm in Section 2.2.1. Once we obtain the position of the transmitter ${}^{a}\mathbf{x}_t = [{}^{a}x\ {}^{a}y]^T$, to transform it to our original reference system:

$$\mathbf{x} = {}^{0}\mathbf{R}_{\mathbf{a}} \cdot {}^{a}\mathbf{x}_t + \mathbf{x}_1 \tag{2.19}$$

The Closed-Form solution is relatively easy to implement and the computation cost is small compared to more complex methods. However, edge cases must be handled

with care. Moreover, TDOA measurements are subject to noise in real-life settings and that might cause the solution to change depending on the reference receiver, as shown in Fig. 2.2. Thus, special care must be taken when using closed-form approaches.

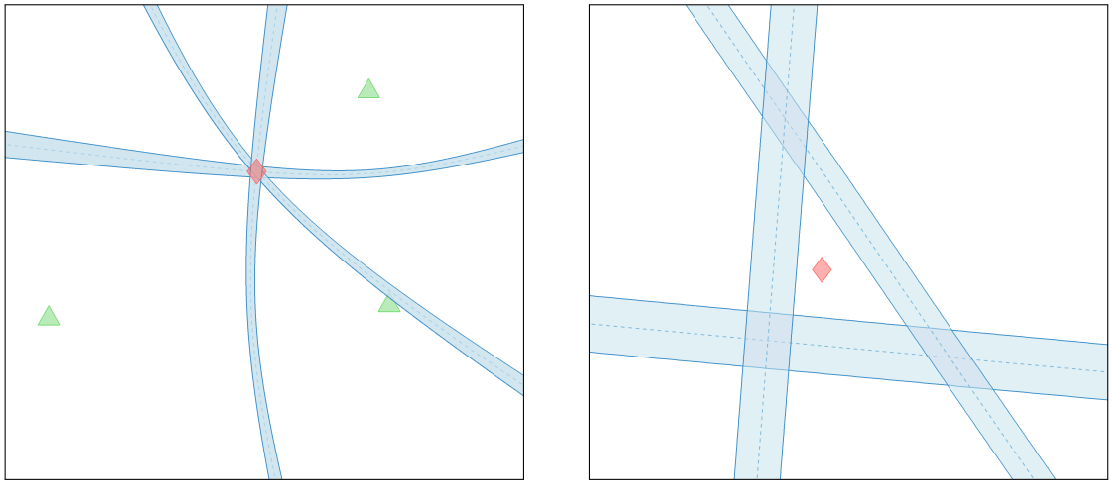


Figure 2.2: How the closed form solution works with noisy data. On the left the three sensors (green), the associated hyperbolas and the real transmitter location (red). On the right the region defined by the intersections after zooming.

2.2.2 Least Squares

When dealing with more TDOA measurements, the Closed-Form solution described in the previous Section cannot be applied. To overcome this, it is possible to linearize the TDOA equations and solve the overdetermined system via Linear Least Squares (LLS). To do so, we create a variable d_1, which is the distance from the reference sensor to the transmitter. In the following lines, we describe the mathematical process in more detail.

System Derivation

Now we proceed to derive the linear system that we will solve for the location of the transmitter. For that using sensor 1 as a reference and sensor i, we can write Eq. 2.2 as:

$$\tau_{1j} = d_1 - d_j = d_1 - \sqrt{(x_j - x_t)^2 + (y_j - y_t)^2}$$

$$\tau_{1j} - d_1 = -\sqrt{(x_j - x_t)^2 + (y_j - y_t)^2}$$

Squaring both sides and subtracting d_1^2 we obtain:

$$\tau_{1j}^2 + d_1^2 - 2 \cdot \tau_{1j} \cdot d_1 = (x_j - x_t)^2 + (y_j - y_t)^2$$

$$\tau_{1j}^2 - 2 \cdot \tau_{1j} \cdot d_1 = (x_j - x_t)^2 + (y_j - y_t)^2 - d_1^2 \tag{2.20}$$

Since d_1 is the distance from sensor 1 to the transmitter, we can rewrite the right hand side of Eq. 2.20 as:

$$\tau_{1j}^2 - 2 \cdot \tau_{1j} \cdot d_1 = (x_j - x_t)^2 + (y_j - y_t)^2 - (x_1 - x_t)^2 - (y_1 - y_t)^2 \tag{2.21}$$

Expanding the squared terms, eliminating and rearranging the variables we obtain:

$$2 \cdot (\tau_{1j} \cdot d_1 + (x_1 - x_j) \cdot x_t + (y_1 - y_j) \cdot y_t) = \tau_{1j}^2 + x_1^2 + y_1^2 - x_j^2 - y_j^2 \tag{2.22}$$

In Eq. 2.22 the unknowns are d_1, x_t and y_t. To make the expression more compact, we can call $r_i^2 = x_i^2 + y_i^2$. If we do this for all sensors from $i = 2 \dots M$ we can obtain a system of equations of the form $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$:

$$\begin{bmatrix} \tau_{12} & x_1 - x_2 & y_1 - y_2 \\ \vdots & \vdots & \vdots \\ \tau_{1M} & x_1 - x_M & y_1 - y_M \end{bmatrix} \cdot \begin{bmatrix} d_1 \\ x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cdot (\tau_{12}^2 + r_1^2 - r_2^2) \\ \vdots \\ \frac{1}{2} \cdot (\tau_{1M}^2 + r_1^2 - r_M^2) \end{bmatrix} \tag{2.23}$$

Eq. 2.23 has an exact solution when the number of sensors M is 4. If M is more, then the solution can be obtained by least squares:

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \tag{2.24}$$

2.2.3 Non-Linear Methods for TDOA

The most accurate method to obtain a position is to take the measurements directly and optimize over the p-norm of the residual. Particularly, for the L2-norm we start from a weighted sum of squared errors between the measured and the expected TDOA:

$$S(\mathbf{x_t}) = \frac{1}{2} \sum_{\forall i,j} w_{ij} \cdot (\hat{\tau}_{ij} - \tau_{ij})^2 = \sum_{\forall i,j} w_{ij} \cdot r_{ij}^2 \tag{2.25}$$

where each term of the summation corresponds to the TDOA value per each sensor pair i, j, and w_{ij} is the weight applied to each of the measurements. For this method, the solution lies at the point where the gradient of $S(\mathbf{x_t})$ is minimum, which can be computed per transmitter coordinate (x_t, y_t) as:

$$\begin{aligned} \frac{\partial S}{\partial x_t} &= - \cdot \sum_{\forall i,j} w_{ij} \cdot r_{ij} \cdot \left(\frac{x_t - x_i}{d_i} - \frac{x_t - x_j}{d_j} \right) \\ \frac{\partial S}{\partial y_t} &= - \cdot \sum_{\forall i,j} w_{ij} \cdot r_{ij} \cdot \left(\frac{y_t - y_i}{d_i} - \frac{y_t - y_j}{d_j} \right) \end{aligned} \tag{2.26}$$

These equations can then be solved by several methods, such as Gradient Descent, Levenberg-Marquardt, or BFGS, among others [45]. In fact, it is quite common to use hybrid approaches, where linear methods serve as an initial coarse estimate for the position and then fine-tune them with non-linear approaches.

2.3 Measuring the Accuracy of TDOA Systems

So far, we have shown how to compute a position based on the differences in arrival times at our receivers. However, often we need a method to estimate the "quality" of the estimation. In this section we explore 2 methods to obtain such estimations.

2.3.1 Cramér-Rao Lower Bound

The Cramér-Rao Lower Bound (CRLB) is a typical metric to estimate the accuracy of a localization system. More formaly, it is the minimum variance that can be achieved by an unbiased estimator. It is thus a very useful estimator for the maximum achievable accuracy for a given set of boundary conditions. The minimum variance can be obtained by the inverse of the Fisher Information Matrix (FIM). Starting from the TDOA model in Eq. 2.2 and with the assumption of Gaussian nose, we can represent the likelihod function $p(\tau_{ij}|\mathbf{x})$ for a single measurement as:

$$p(\tau_{ij}|\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \cdot \exp\left(-\frac{1}{2\sigma_{ij}^2}(\hat{\tau}_{ij} - \tau_{ij})^2\right) \tag{2.27}$$

The FIM is then calculated as the expectation of the gradient of the logarithm of the likelihood function:

$$\mathbf{J} = \mathbb{E}\left[\nabla_{\mathbf{x}} \ln p(\tau|\mathbf{x}) \left(\nabla_{\mathbf{x}} \ln p(\tau|\mathbf{x})\right)^T\right] \tag{2.28}$$

Assuming that the measurement noise is constant and does not depend on the position of the target to be localized, the FIM can be calculated as [46]:

$$\mathbf{J} = \frac{\partial \tau}{\partial \mathbf{x}} \mathbf{R}^{-1} \frac{\partial \tau}{\partial \mathbf{x}} \tag{2.29}$$

$\frac{\partial \tau}{\partial \mathbf{x}}$ is the Jacobian matrix of the measurement equation (2.2) and can be represented as:

$$\frac{\partial \tau}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \tau_{12}}{\partial \mathbf{x}} & \frac{\partial \tau_{12}}{\partial \mathbf{y}} \\ \vdots & \vdots \\ \frac{\partial \tau_{1N}}{\partial \mathbf{x}} & \frac{\partial \tau_{1N}}{\partial \mathbf{y}} \end{bmatrix} \tag{2.30}$$

The measurement noise covariance matrix $\mathbf{R}$ can be estimated like the following (assuming that we use receiver 1 as the reference) [38]:

$$\mathbf{R} = \begin{pmatrix} \sigma_1^2 + \sigma_2^2 & \sigma_1^2 & \cdots & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 + \sigma_3^2 & \cdots & \sigma_1^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^2 & \sigma_1^2 & \cdots & \sigma_1^2 + \sigma_N^2 \end{pmatrix} \tag{2.31}$$

The minimum variance achievable by the TDOA system can then be calculated as:

$$\sigma(\mathbf{x}) \geq \mathbf{J}(\mathbf{x})^{-1} \tag{2.32}$$

It is also essential to have an estimate of the maximum achievable accuracy when estimating the different TDOA pairs. As per [38] the CRLB of the TOA estimation is defined as:

$$\sigma_i^2 \geq \frac{c^2}{\mathrm{SNR}_i \cdot B^2} \tag{2.33}$$

where c is the speed of light, SNR_i is the Signal-to-Noise Ratio (SNR) at receiver i and B is the bandwidth of capture. This results shows the power of TDOA over large distances. While approaches like RSS decrease their performance with distance [47], the accuracy on the parameter estimation on time-based systems depends on the square of the bandwidth. Thus larger capture bandwidths result in higher possible TOA estimation accuracy.

2.3.2 Dilution of Precision

Another method to estimate the accuracy of a time-based positioning scheme, widely used in GNSS systems, is the Dilution of Precision (DOP). DOP quantitatively represents the impact of the geometric configuration of the receivers on the position estimation. The error in the estimation ($\Delta\mathbf{x}$) is proportional to the error in the TDOA ($\Delta\tau$) measurement multiplied by the DOP:

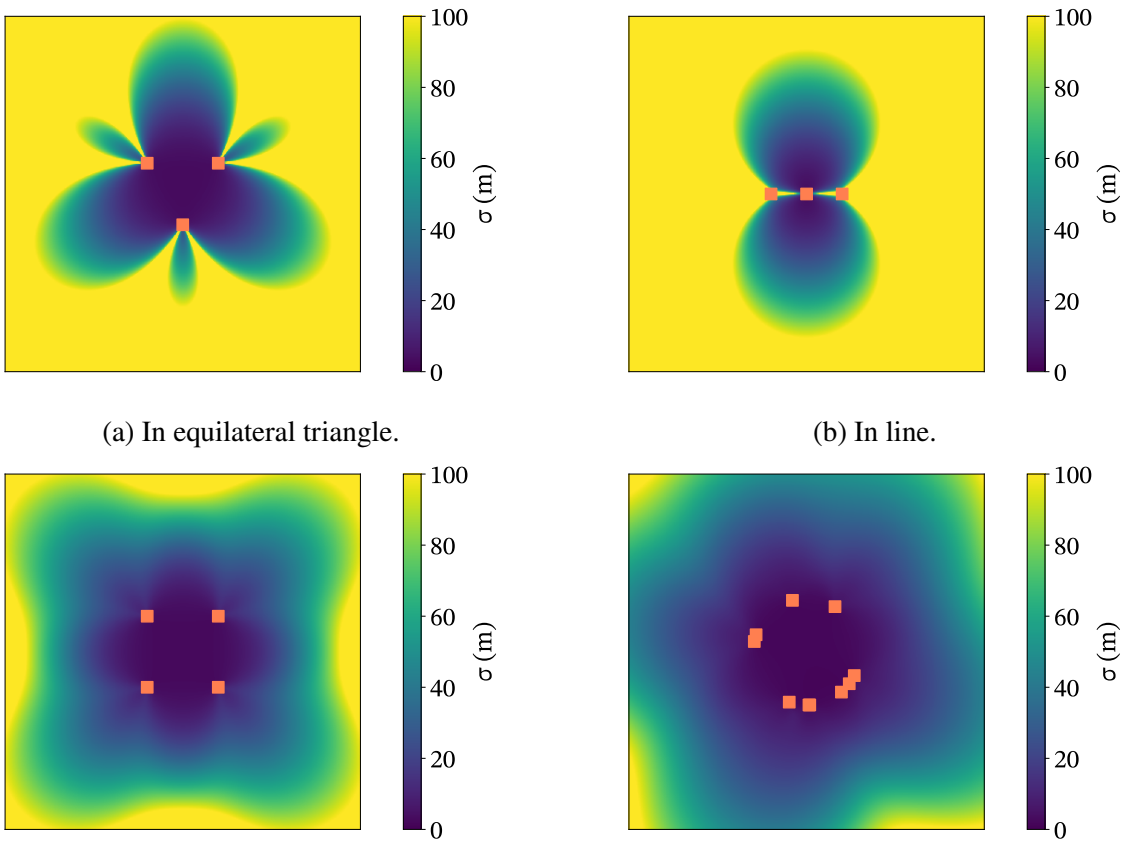
$$\Delta\mathbf{x} = \mathrm{GDOP} \cdot \Delta\tau \tag{2.34}$$

To compute the DOP we need to take the Jacobian of the measurement vector as described in Eq. 2.30.

Finally, DOP is calculated as the square root of the trace of the matrix product $(\frac{\partial\tau}{\partial\mathbf{x}}^T \frac{\partial\tau}{\partial\mathbf{x}})^{-1}$:

$$\mathrm{GDOP} = \sqrt{\mathrm{trace}((\frac{\partial\tau}{\partial\mathbf{x}}^T \frac{\partial\tau}{\partial\mathbf{x}})^{-1})} \tag{2.35}$$

A lower DOP value implies a more favorable geometric configuration, leading to enhanced accuracy in the position estimation. Conversely, a higher Geometric Dilution of Precision (GDOP) suggests a less ideal geometry, potentially decreasing the precision of location estimates. Note that this result is very similar to the CRLB estimation. In fact, under constant variance, both methods only differ by the measurement noise covariance matrix $\mathbf{R}$.



(a) In equilateral triangle.

(b) In line.

(c) In square.

(d) Randomly placed in a circle.

Figure 2.3: CRLB of 4 different geometric configurations. Bandwidth was chosen to be 5MHz and SNR 10 dB.

2.3.3 What are Good Geometries?

Once the accuracy estimators have been laid out, a natural question would be: What are the best/worst geometric configurations that can exist? In this paragraphs, we will provide some geometric configurations along with their expected error distribution. Fig. 2.3 shows 4 different geometric configurations and the computed CRLB around them.

Fig. 2.3a shows the typical equilateral triangle configuration. For positions within the area enclosed by the 3 transmitters, the expected error is low, but it increases rapidly when outside of it. A relatively bad configuration is depicted in Fig. 2.3b, where all receivers lay on the same line. This geometry has another problem which is that due to symmetry, each solution to the TDOA equations will be dual, thus further knowledge would be required to discriminate the actual position. In Figs. 2.3c,2.3d we can observe the effects of increasing the number of receivers. In general, the more sensors placed in a single area, the lower the CRLB, and thus the lower the expected error in the positioning results.

PART II

LOW-COST SDR IN LOW-LEVEL APPLICATIONS

3 ORAN-Sense: Localizing Non-cooperative Transmitters with Spectrum Sensing and ORAN

3.1 Introduction

Wireless communication infrastructures are growing at a rapid rate [48], and this trend will continue with the ongoing deployments of 5G/6G networks. As RF spectrum is scarce, monitoring it on a large scale is fundamental for gaining insights into the EM spectrum usage and its evolution, as well as detecting threats attempting to cause disruption. For instance, network providers spend billions of dollars in spectrum auctions, making the problem of protecting their spectrum and localizing non-cooperative transmitters, a key pillar to running their business.

However, localizing non-cooperative transmitters is a difficult task, and it has been so far decoupled from 5G deployments. Current methods require network operators to deploy temporary, bulky, and expensive equipment if an anomaly in spectrum usage has been found around a particular area. This method has several disadvantages, as it is impossible to continuously monitor the spectrum and operational costs are high. The advent of IoT devices in the last years equipped with SDR capabilities has significantly influenced the approach to spectrum monitoring, as they can establish a network of low-cost distributed sensors that transfer data in the cloud for processing, revolutionizing the spectrum monitoring and breaking down cost barriers.

In the past few years, several crowdsourced spectrum monitoring solutions have emerged using low-cost devices and custom backends, such as Microsoft Spectrum Observatory [49] and Electrosense [7], [15]. However, the maintenance costs of such platforms, particularly dedicated backends, are often too high, and these initiatives do not last long. Several of these initiatives have shut down after years of effort.

Rather than taking the traditional approach to spectrum crowdsensing of having a third party that processes spectrum data in its dedicated backend, we propose utilizing the backend infrastructure of the Radio Access Network (RAN) and maintained by the network operator themselves. We envision that IoT spectrum sensors will be deployed at users' fa-

cilities and can collaborate and share their data with the network operator. This is similar to femtocell technologies adopted by network operators where existing broadband connections are used to connect small form-factor base stations to the operator Core Network [50]. This way, network operators would have access to sites for deploying spectrum sensors that would not be available without using a crowdsensing paradigm. On the other hand, the network users will still benefit from access to similar services as in prior initiatives (e.g., decoding public signals and their audio and visual representation [16]). Still, the resources in the backend would now be shared among several software services following the 5G paradigm, significantly cutting operational costs.

To transform this idea into practical realization, we leverage the fact that the Next Generation of the cellular network is going under a radical evolution as the current architecture will be replaced by programmable and virtualized infrastructure, including the RAN [51], [52]. The O-RAN is the primary driver behind this deployment as it promotes softwarization, disaggregation, openness, and programmable hardware [53]. Furthermore, the O-RAN Alliance [54] is currently defining the concept of the O-RAN paradigm to streamline the implementation, testing, and deployment of flexible solutions at scale. One of the peculiarities of O-RAN is the introduction of RIC, programmable components that can run optimization routines to control and orchestrate the RAN effectively. Our idea is to leverage the open interfaces and RICs to perform the localization of non-cooperative transmitters, enriching cellular networks' capabilities that can already localize cooperative transmitters through 5G signals [55].

In summary, this chapter proposes merging crowdsensing networks with the O-RAN interfaces to make a robust system that leverages existing infrastructure to monitor the spectrum and collect data to perform more advanced analytics tasks. The contributions of this chapter are the following:

- A novel framework to integrate IoT devices into O-RAN networks to perform spectrum monitoring tasks at a more sustainable cost than in prior initiatives.
- We develop ORAN-Sense, an architecture that leverages this framework and can use these IoT spectrum sensors to localize non-cooperative transmitters within a city.
- We provide an experimental evaluation, with sensors deployed in the wild in 2 different European cities, and show that it is possible to build and use such a network.

The remainder of this chapter is organized as follows: Section 3.2 dives into O-RAN and its use cases and discusses the main localization approaches described in the literature. Section 3.3 provides background insights into O-RAN and its interfaces. Section 3.4 proposes an architecture that addresses the challenges in the design and the detailed steps to perform geolocation at scale, and Section 3.5 provides an evaluation of the different components of our architecture. Last, Section 3.6 provides the conclusions.

3.2 Related work

O-RAN provides a promising and flexible architecture for future cellular networks, and many applications might be possible thanks to it [53]. Localization, in particular, was proposed in [56] and even using passive sensors and signals of opportunity using DVB-T signals to monitor human activity. However, all localization approaches revolve around positioning users connected to the network. In this chapter, we propose to extend this emerging architecture by means of passive sensors that can perform different tasks and focus on the positioning of several types of transmissions and modulations.

When tackling the problem of localizing unknown transmitters, we can distinguish two main categories: range-based and range-free methods. Range-based positioning is generally tackled from four perspectives as summarized in [25]. In the following paragraphs, we discuss some of the most relevant research with each of the aforementioned localization methods described. Many articles related to localization are based on theoretical conjectures and simulations, but rarely are these assumptions tested on scalable real deployments or areas greater than 100×100 m^2.

RSS based methods In recent years, much effort has been focused on RSS based systems [26], [27]. The main appeal of RSS based methods is the low hardware requirements [32], which makes an attractive use case for deployments on large sensor networks. However, these techniques have drawbacks, such as the need for good calibration, complex path-loss models, and lower accuracy when distance increases.

AOA based methods AOA requires complex receiver chains to compute the angle of incidence locally. It also has notable accuracy in theory and practice [35]. One of the main issues of AOA based methods is that to compute the angle of incidence, they require antenna arrays which might be challenging when deploying embedded sensors.

TOA & TDOA based methods TOA has been widely employed as a localization method in sensor networks since it provides accurate results and does not need to make assumptions on the signal [38]. TOA can be implemented using one-way transmissions or with echo techniques [39]. TDOA methods have slightly less accuracy with respect to TOA based localization approaches [38] but only need time synchronization between receivers. We have several examples in [22], [57]–[60]. However, as for RSS based methods, the proposed systems are either focused on one specific technology, use high-end hardware or remain conceptual and do not pursue any larger scale implementation. [59] proposed a hybrid network of high- and low-end receivers, but it focused solely on aerial vehicle tracking and its protocols.

In our work, to minimize bandwidth usage, localization must be triggered only during a short period (e.g., after an algorithm has detected an anomaly in the spectrum). An-

other way to mitigate this is to compress this data by means of algorithms like the Fast Walsh-Hadamard Transform [61] or compressing sensing [62]. Sensors also need to be accurately geolocated. Work has been realized, for example, to opportunistically exploit time information in aerial signals as shown in [40], [63] without relying on GPS.

3.3 O-RAN Primer: Components and Interfaces

This section introduces background knowledge on the specific components of O-RAN networks and how the design can support the entities for localization.

3.3.1 Components

An overview of the general architecture is depicted in Fig. 3.1. O-RAN incorporates the 3GPP NR 7.2 logical split[64], dividing base station functionalities at different network locations. For instance, the CUs and DUs can be virtualized and deployed on different servers; meanwhile, the RUs are placed in proximity to the RF antennas[51].

Service Management and Orchestration Framework (SMO). There are numerous management domains in a service provider's network. The main goal of this component is to manage and orchestrate the RAN domain [65], e.g., Cloud Management, Orchestrator, and Workflow Management and resources optimization with RICs.

Collector in the SMO domain. The SMO hosts the Non-Real-Time RIC (Non-RT RIC). However, there might be other network functions that may provide additional services. Notably, the ORAN-Allicance standard does not establish a formal boundary with the Non-RT RIC and other logical functions within the SMO framework [66]. In this context, an Operations and Maintenance (OAM) Function can be used to manage components, like the RIC or the connection with external services (e.g., Cloud) [67], that we call Collector. It can gather information from external services, such as custom back-end external applications, and interact with controllers to deploy or update RAN policies.

Non-RT RIC. The Non-RT RIC is another key component of O-RAN enabling closed-loop control of the RAN with timescales larger than 1 s. Its main goal is to optimize the RAN intelligently by providing policy-based guidance and enrichment information to the Near-Real-Time RIC (Near-RT RIC) function. This is achieved through the standard open interfaces, using declarative policies expressed with formal statements, allowing Non-RT RIC to guide Near-RT RIC for specific optimizations, such as localization algorithms or particular areas and frequencies to explore.

Near-RT RIC. Like the Non-RT RIC, the Near-RT RIC enables control and optimization of RAN functions and resources in near-real-time. It facilitates data gathering and actions via the E2 interface, with a control loop ranging from 10ms to 1s. One or more *xApps*, i.e., AI Models or algorithms, can be hosted by the Near-RT RIC, and the E2 in-

terface enables a direct association between xApp and E2Node functions to consume the data.

3.3.2 E2 Interfaces, E2 Node and Service Model

The components introduced in Section 3.3.1 are connected with open interfaces, following the specifications established by the O-RAN Alliance.

The E2 Interface is an open interface between two endpoints, the Near-RT RIC and the E2Nodes. It enables the collection of metrics from the RAN to the Near-RT RIC, either periodically or after pre-defined trigger events.

An E2Node, comprising DUs and CUs, is a logical node and offers a variety of *RAN functions*, i.e., services or capabilities, and supports the connection with E2 Interface [68]. For example, DUs from different vendors might provide their ability to gather and report varying performance metrics. During the connection, the E2Node (the gNB) exposes the RAN Functions to which the Near-RT RIC can subscribe.

The E2 Service Model (E2SM) in the O-RAN E2 interface is a protocol detailing the E2Node's capabilities to the Near-RT RIC. It is a list of *RAN function* offerings that the E2Node makes available, including performance metrics and services. The communication between an E2 Node and Near-RT RIC uses a publish-subscribe paradigm. The E2Nodes can publish their data depending on whether the reporting is periodic or trigger-based.

Currently, the standard defines guidelines to monitor the Key Performance Measurements (E2SM KPM) provided to the xApps [68]. However, these are limited upper-layer metrics to optimize scheduling policies or RAN slicing [53]. In this paper, we propose enhancements that *enable the E2 interface to transmit localization-specific RF measurements from IoT devices*, broadening the service's utility model.

3.4 ORAN-Sense

The architecture of ORAN-Sense is shown in Fig. 3.1. We propose that low-cost sensors act as E2Nodes, consisting of a commercial off-the-shelf antenna, an RTL-SDR, and a Raspberry Pi (Fig. 3.2). Like in femtocell technology [50], network operators connect these sensors in different locations like homes or offices via Ethernet or wirelessly to increase the monitoring coverage. Localization starts in the Non-RT RIC by declaring the policy of the area and the frequencies of interest. The policy commands the E2 nodes, specifically those DUs equipped with sensors, that expose the network function to sense and gather IQ measurements for a specific short time. Then, the data are consumed by the localization xApp within the Near-RT RIC. The localization stages are explained in the following paragraphs.

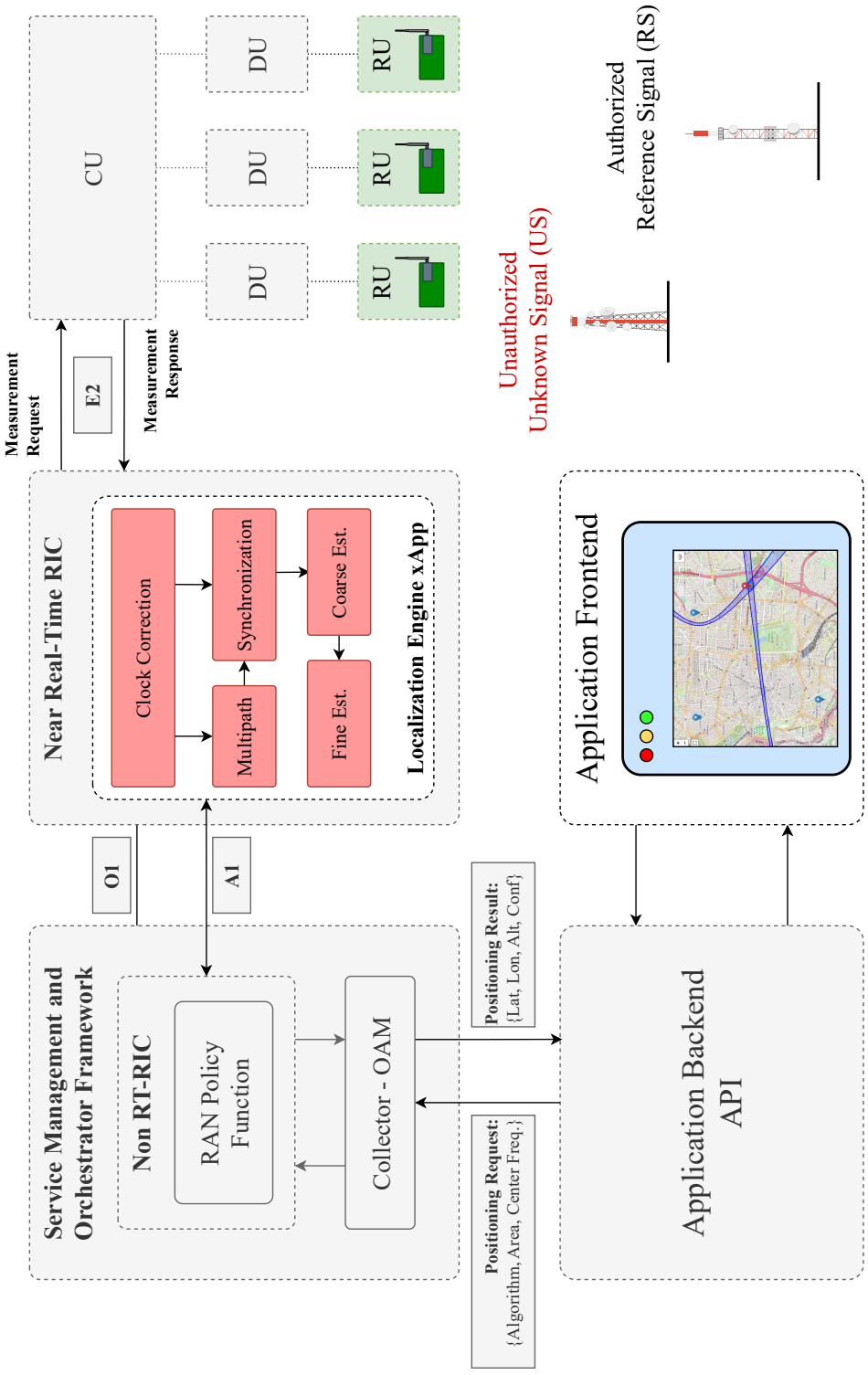


Figure 3.1: Proposed architecture for ORAN-Sense, a TDOA-based localization approach of non-cooperative transmitters that can be embedded in O-RAN by using its open interfaces. In color are the parts implemented by the author of the thesis, namely: software on the sensors, core logic of the xApp and an application frontend to visualize the results.

The localization architecture proposed in this chapter is divided into different parts: the **Sensor** that we consider as the entity within the E2Node with only reception capabilities, the localization **xApp** that runs in the Near-RT RIC and the **SMO** which contains a campaign manager.



Figure 3.2: The RU is comprised of a Raspberry Pi and an RTL-SDR

Spectrum sensors only measure IQ samples with a 2 MHz bandwidth (larger bandwidths are possible with higher-performing SDRs), tuned to the center frequency of the band to be measured. This also means that if the signal of interest has a larger bandwidth than the sampling rate of the RTL-SDR front-end, information outside this band is not used by the RTL-SDR-based spectrum sensor for localization purposes. As our system does not perform any decoding of the reference and target signals, this issue is irrelevant in the context of localization of non-cooperative transmitters.

To transmit the IQ measurement command, we use the E2 interface defined by the O-RAN. When the measurement command is received, sensors first estimate their local oscillator offset, similarly to [69]. After the given measurement time elapsed, these sensors send IQ samples and their corresponding local oscillator offsets, measured in PPM.

Once data is gathered in the Near-RT RIC, the localization process is subdivided into the following steps: *Frequency Offset Correction*, *Synchronization and TDOA Estimation* and *Multilateration*. More details are provided in the remainder of this section. We conclude the section by providing technical details on the O-RAN integration.

3.4.1 Frequency Offset Correction

At operating temperatures, SDR receivers generally have constant drifts in their clocks. For example, the latest versions of the RTL-SDR are guaranteed to have ± 1 PPM, but this drift can be even higher when it is heating up [69]. We cannot assure that the SDR will be at the correct operating temperature at any given time, so we need to estimate the drift accordingly. The following sections show that the clock drift can significantly influence the measurement results.

Our system corrects both the center and the sampling frequency offsets. We assume

that in the IoT receiver, a single Local Oscillator (LO) feeds both the down-conversion and sampling stage. Therefore, we assume that both errors in the center and sampling frequencies are similar:

$$\frac{\Delta f_s}{f_s} \approx \frac{\Delta f_c}{f_c} \approx \varphi \tag{3.1}$$

where f_s, f_c and φ are the sampling frequency, center frequency and PPM offset respectively.

To correct the center frequency, f_c, we must note that the actual measured center frequency, f_c', is:

$$f_c' = (1 + \varphi) \cdot f_c \tag{3.2}$$

where $\Delta f_c = \varphi f_c$. Each sample is received at time $n \cdot T_s'$, where T_s' is the observed sampling period, which can be related to the desired sampling period, T_s, as $T_s' = T_s/(1 + \varphi)$.

We must perform a frequency shift of the samples of the measured signal denoted as $s'[nT_s']$. We can then generate the new signal, shifted to the desired center frequency:

$$s''[nT_s'] = s'[nT_s'] \cdot e^{-j2\pi \cdot nT_s' \cdot \varphi f_c} \tag{3.3}$$

Now that the center frequency f_c is corrected, we can correct the sampling frequency f_s. If the received signal were continuous, it would be enough to take the samples at the time instants corresponding to the desired sampling rate. As the received signal is not continuous, we re-sample the data at the correct rate by interpolating samples. Thus the corrected signal $s[nT_s]$ can be regarded as changing from sampling rate $nT_s/(1 + \varphi)$ to nT_s:

$$s[nT_s] = I(s''[nT_s'], \frac{nT_s}{1 + \varphi}, nT_s) \tag{3.4}$$

3.4.2 Synchronization and TDOA Estimation

TDOA based systems require sensors to be time synchronized. One of the most standardized manners to synchronize the clocks over the Internet is to use the Network Time Protocol (NTP), which can achieve accuracies of a few milliseconds. However, for a localization system, this is not satisfactory.

Some sensor networks rely on GPS as their source of synchronization. There exist commercial solutions offering errors of around 50 ns. However, *the sensor cost would increase drastically*, as commercial solutions providing high accuracy are in the order of more than 1,000.00 USD[5]. We do not intend to rely on any GPS source, so even the lower end of SDR receivers like the RTL-SDR can be utilized. We also note that sensors cannot use fine network synchronization, as they are connected to the CU only through standard Internet connectivity. To solve this challenge, we propose an approach that utilizes NTP to coordinate the sensing processes of different sensors and then uses a Reference Signal (RS)

[5]https://www.ettus.com/all-products/gpsdo-mini/

as a signal of opportunity for synchronization, and the Unknown/Target Signal (US), which we aim to locate.

We start from the TOA equation (2.1). For each pair of IoT receivers, their sampled signals are sent to the Localization Engine xApp, where they are subtracted from one another. In this way, we can remove the transmit time from the equation:

$$\tau_{ij} = t_i - t_j = \frac{d_i}{c} - \frac{d_j}{c} + (\theta_i - \theta_j) + \eta_{ij} \tag{3.5}$$

We multiply both sides by the speed of light c and denote $d_{ij} = d_i - d_j$ as the distance difference between receivers. We can apply this to both the RS and US:

$$\begin{aligned} c \cdot \tau_{ij}^{us} &= d_{ij}^{us} + c \cdot (\theta_i - \theta_j) + c \cdot \eta_{ij}^{us} \\ c \cdot \tau_{ij}^{rs} &= d_{ij}^{rs} + c \cdot (\theta_i - \theta_j) + c \cdot \eta_{ij}^{rs} \end{aligned} \tag{3.6}$$

We can subtract one from the other, and then by knowing the distance difference from sensors i and j from the reference transmitter, we can isolate the desired TDOA value:

$$c \cdot (\tau_{ij}^{us} - \tau_{ij}^{rs}) = d_{ij}^{us} - d_{ij}^{rs} + c \cdot \eta_{ij}^{rs,us} \tag{3.7}$$

If we now add the distance to the RS, which is known, we are only left with the TDOA associated with our target transmitter:

$$c \cdot \tau_{ij} = d_{ij}^{us} = c \cdot \tau_{ij}^{us} - (c \cdot \tau_{ij}^{rs} - d_{ij}^{rs}) + c \cdot \eta_{ij}^{rs,us} \tag{3.8}$$

To obtain the delay between two signals s_i, s_j, we calculate the maximum delay τ_{ij} of their cross-correlation:

$$\hat{\tau}_{ij} = \arg\max_{\tau} \left[(s_i \star s_j)[\tau] \right] \tag{3.9}$$

Traditionally, only raw IQ samples are used for performing cross-correlation. In this chapter, we instead evaluate three different possibilities to perform the cross-correlation: using the raw IQ samples (`iq`), using the absolute value (`abs`) or the phase difference (`dphase`). We present these correlation methods in Table 3.1. Note that we rely on the I and Q samples measured by the RTL-SDR front-end in all methods.

Name	Computation Method
`iq`	$I[nT_s] + jQ[nT_s]$
`abs`	$\sqrt{I[nT_s]^2 + Q[nT_s]^2}$
`dphase`	$\angle s_{i,iq}[nT_s] - \angle s_{i,iq}[(n-1)T_s]$

Table 3.1: Correlation methods studied in this chapter.

One issue to consider relates to the quantization error one might have in a single sample when computing the TDOA. For the RTL-SDR, given that we sample at around 2 MSamples/s, that would mean that a difference of one sample can result in an error of 150 m. As studied in detail in the evaluation section, we propose to mitigate the uncertainty region caused by the low sampling capabilities of low-cost IoT sensors by upsampling the signals and computing the cross-correlation on the upsampled versions.

3.4.3 Multipath mitigation

In urban scenarios, signals are expected to be either reflected or entirely occluded by buildings, trees, or any other type of obstacle. This directly affects a time-based localization system like the one proposed in this chapter. When sensors are affected by multipath, they receive several copies of the signal at different times. Choosing the wrong copy might result in errors of tens or even hundreds of meters.

Under multipath and Non-Line of Sight (NLOS) conditions, the received signal by a spectrum sensor i with L_i different paths can be modelled as:

$$s_i = \sum_{l=1}^{L_i} \alpha_{i,l} \cdot s_{i,l}(t - \tau_{i,l}) + \eta_{i,l} \tag{3.10}$$

where $\alpha_{i,l}$ is the attenuation factor of the lth multipath component and $s_{i,l}$ are the time-delayed versions associated to that component, and $\eta_{i,l}$ is the noise in the measurement.

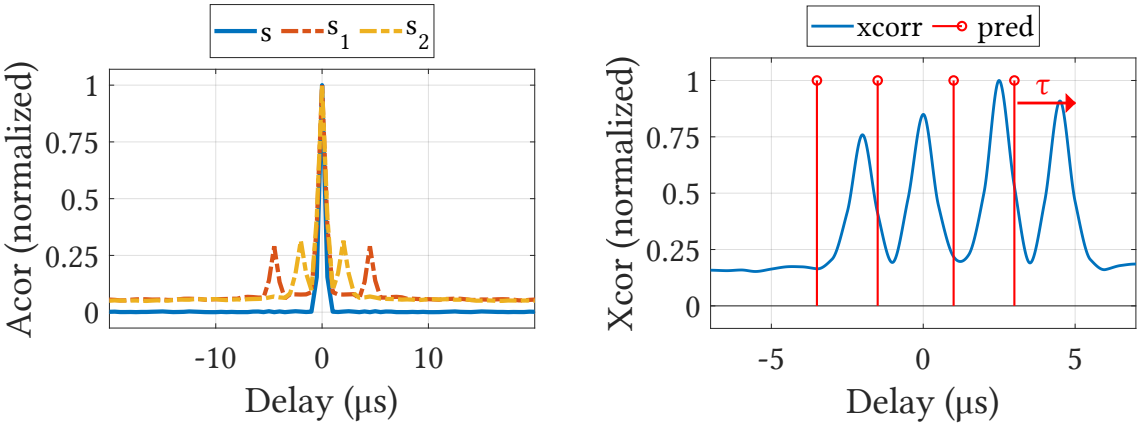
Differently from TOA measurements, the reflected peaks in TDOA measurements may also appear before the Line-of-Sight (LOS) peak, which further hinders identifying the shortest path. Thus, the naive approach of selecting the highest peak would not work. The effect of these multiple components on the TDOA estimation can be observed on the *autocorrelation* of signals with the appearance of peaks at times $(0, \pm m_{i,1}, \pm m_{i,2}, \ldots, \pm m_{i,L})$. If signals received from two sensors i and j are affected by multipath, then their *cross-correlation* function appears generally with $L_i \times L_j$ peaks, where L_i and L_j are the numbers of paths received by sensor i and j, respectively (see also the example in Fig. 3.3).

We estimate the multipath components per sensor from the autocorrelation function (see Fig. 3.3). This results in an array of the components for sensor i: $[0, m_{i,1}, \ldots, m_{i,L_i}]$. By combining all possibilities for a pair of sensors $\{i, j\}$, we generate a hypothesis of the generated multipath components at a given estimate for the direct path component delay τ:

$$C(\tau) = [\tau - m_{j,L_j}, \ldots, \tau + m_{i,L_i}]; \quad \forall m_{i,l_i}, m_{j,l_j} \tag{3.11}$$

Then, computing the cross-correlation function between these two sensors, we can obtain a set of the predicted peaks $\hat{C}$. The correct delay τ_{ij} for a pair of sensors $\{i, j\}$ will be such that it minimizes the distance between the hypothesis and the measured vectors:

$$\tau_{ij} = \arg\min_{\tau} ||C(\tau) - \hat{C}|| \tag{3.12}$$



(a) Autocorrelation

(b) Cross-Correlation

Figure 3.3: Effect of multipath on cross-correlation between 2 signals. On the left, we have the autocorrelation of each one. The signal affected by multipath will present multiple peaks. On the right is the cross-correlation between both. If the indirect path is stronger (as in this case), we might select the erroneous peak (instead of the one at 0 μs), leading to higher errors in the location estimation.

To visualize the effect of multipath on TDOA measurements, we provide a small example. Let us suppose we receive signal s from 2 receivers (s_1,s_2), and each one has another delayed component by 4.5 and 2 μs, respectively. To detect those extra components, we rely on the auto-correlation function (Fig. 3.3a). Receiver 1 will have components $[0, 4.5]$ and receiver 2 $[0, 2]$. If we compute the cross-correlation of both signals, peaks should appear at $[-4+\tau, \tau, 5+\tau, 9+\tau]$, where τ is the TDOA. We can now perform the cross-correlation between both signals and find the peaks (Fig. 3.3b). We can then apply the minimization routine in Eq. 3.12 and solve it with a brute-force algorithm.

3.4.4 Multilateration

In the last component of the architecture, the actual transmitter positioning is performed. Starting from a weighted sum of squared errors between the measured and the expected TDOA:

$$S(\mathbf{x_t}) = \sum_{\forall i,j} w_{ij} \cdot (\hat{\tau}_{ij} - \tau_{ij})^2 = \sum_{\forall i,j} w_{ij} \cdot r_{ij}^2 \tag{3.13}$$

where each term of the summation corresponds to the TDOA value per each sensor pair i, j, and w_{ij} is the weight applied to each of the measurements. Note that if $w_k = 1$, then this becomes the L2-norm. For this method, the solution lies at the point where the gradient of $S(\mathbf{x_t})$ is minimum, which can be computed per transmitter coordinate (x_t, y_t) as:

$$\begin{aligned} \frac{\partial S}{\partial x_t} &= -2 \cdot \sum_{\forall i,j} w_{ij} \cdot r_{ij} \cdot \left(\frac{x_t - x_i}{d_i} - \frac{x_t - x_j}{d_j} \right) \\ \frac{\partial S}{\partial y_t} &= -2 \cdot \sum_{\forall i,j} w_{ij} \cdot r_{ij} \cdot \left(\frac{y_t - y_i}{d_i} - \frac{y_t - y_j}{d_j} \right) \end{aligned} \tag{3.14}$$

These equations can then be solved by several methods like Gradient Descent, Levenberg-Marquardt, or BFGS, among others [45]. Referring to Fig. 3.1, we use linear methods as an initial coarse estimate for the position and then fine-tune them with non-linear approaches.

3.4.5 Localization embedded into ORAN

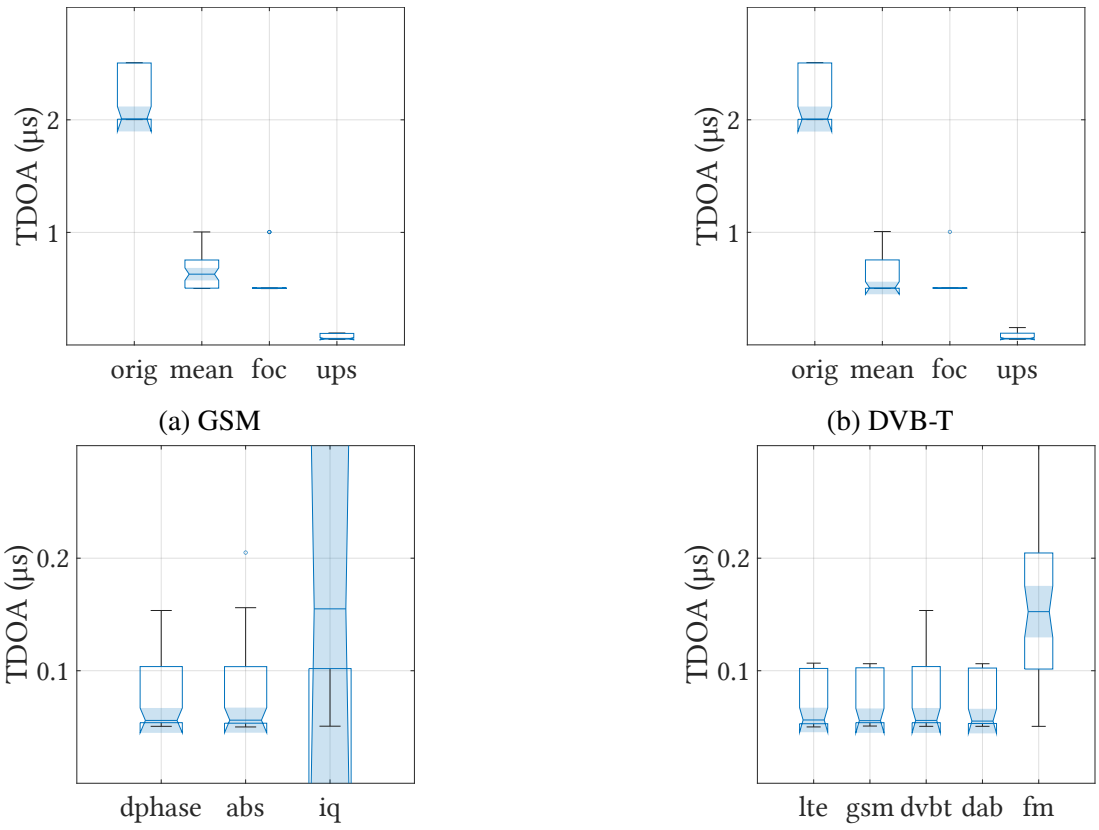
The localization procedure starts with a request the Collector receives from another automated RAN function, like anomalous spectrum usage with technology classification [70] or a request by the network's operator through an external service. The request must specify the number of sensors (3 for 2D localization, 4 for 3D), the algorithm, the area, the duration, and the frequencies to inspect (namely, the ones of the reference signal and the unknown one). The collector forwards the information to the Non-RT RIC, requesting RAN data for localization. The Non-RT RIC formulate the *RAN Localization Policy* delivered to the Near-RT RIC through the A1 Interface.

The Near-RT RIC subscribes to the E2Service model provided by the E2Nodes, specifically the CUs or DUs in the area of interest, specifying interested frequencies and trigger-based report measurement. The latter is managed by a timer set within the E2Node.

Upon timer activation, the IoT devices designed for spectrum sensing within the DUs (or in the RUs) tune to the designated frequencies, collecting the necessary measurements that will be packaged in the E2 Report Message. At the expiration of the timer, the E2 Node publishes the localization measurements over the E2 Interface, consumed by the xApp. In the proposed design, *the measurements are not transmitted as a continuous data stream, avoiding overwhelming the network*. The xApp runs the localization algorithm, e.g., TDOA based algorithm, with the measurements collected over different locations. The outcome of the TDOA localization is forwarded to the Non-RT RIC that either notifies the collector and the External Service along with positioning coordinates.

3.5 Evaluation

This section describes the evaluation of the system using realistic scenarios. To evaluate the system, we define three sets of experiments. The first is the feasibility studies, and we look at how frequency offset correction, upsampling, and different signals could affect the system's performance. In the second set, we evaluate the real-life applicability of different signals in our localization architecture. We combine the results of these evaluations into Subsection 3.5.1 Last, we study the performance of the different multilateration algorithms using real data from our deployments in Subsection 3.5.2.



(a) GSM (b) DVB-T

(c) Methods (d) Technologies

Figure 3.4: Feasibility study results. The two figures on the left describe the results with GSM and DVB-T, respectively. We observe that by applying frequency-offset correction (`foc`) and upsampling (`ups`), we can obtain synchronization errors lower than 100 ns. In (c), we can see the differences between the proposed correlation methods with DAB signals, and in (d), we compare the different technologies that are present in urban environments.

3.5.1 Reference Signal Suitability

For this part of the evaluation, we use three RTL-SDR v3 equipped with TCXO and collect samples for four different technologies: LTE, GSM, DVB-T, and FM radio. We need to collect reference and target signals in the same trace. To achieve this, we modify *librtlsdr* [37] to allow for a center frequency change while sampling. By continuously sampling data, we avoid losing samples between sensing processes, and thus we can obtain the actual synchronization delay between a pair of sensors. We first tune to the RS, then US, and back to the RS, splitting traces into 3 **chunks** with the same number of samples. We collect 200 traces for each receiver with 3 seconds of data, 1 per signal chunk. We also collect one second of LTE traces for each measurement to calculate the oscillator offsets. The experimental setup is summarized in Table 3.2.

To collect data in this first set of experiments, we connect three RTL-SDR devices to an active splitter, an antenna, and an external power supply. The three SDR devices are

then connected to the same machine. An schematic of the setup is shown in Fig. 3.5.

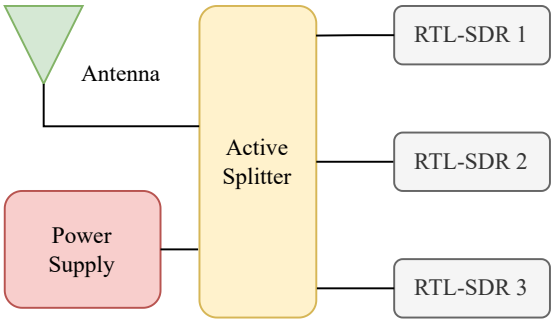


Figure 3.5: Splitter setup for the feasibility tests, with 3 RTL-SDRs sharing the same antenna.

Measuring with this setup, the signal is received by only one antenna, and the differences in the measurements are due only to their internal clock offsets or other hardware imperfections. After collecting data, we analyze all the traces and show the different results in Fig. 3.4. Since the antenna is the same for all sensors, the observed differences should be 0. Values different than that might be caused due to the offset between sensors not being properly corrected or the signal being too noisy to be suitable for a reference signal.

Effects of Offset correction and Upsampling

Figures 3.4a and 3.4b show the results of the feasibility studies using GSM and DVB-T respectively in four scenarios: when no offset correction is made (`orig`) when we use the offset in both chunks that contain RS and average the delay (`mean`), when we correct the local oscillator offset (`foc`) and when we correct offset and upsample the signals (`ups`).

In these plots, we can observe that even though the LO errors are relatively small (in our

RS	f_{RS} (MHz)	US	f_{US} (MHz)	Bandwidth (MHz)
LTE	816	LTE	806	10
GSM	938.8	LTE	806	0.2
DVB-T	562	LTE	806	5
DAB	196	LTE	806	1.5
FM	98.8	LTE	806	0.2

Table 3.2: Experiment setup for the feasibility studies. We vary the reference signal (RS), and we use as an unknown signal the LTE signal at 806 MHz. Bandwidths higher than 2 MHz were clipped to 2 MHz, as it is the limit of the SDR

experiments, we do not observe any values above 1 PPM, which matches the specification for the selected RTL-SDR), they can have a great influence on the TDOA values, and thus these errors must be corrected.

One approach to correct these errors is to assume the offset between two sensors is linear, which is not always a reasonable assumption since their oscillator errors can be unstable depending on the conditions. We can, however, have a coarse estimate by averaging the offset between both RS chunks and subtracting it from the delay obtained with the US chunk, and we can improve the results notably in all cases (this is the method `mean` on Fig. 3.4).

If we perform the offset correction, we can reduce the delays obtained in almost all experiments to 0. However, using a sampling rate of 2 MSamples/s, our computation is affected by the quantization error of 150 m, which still represents a high uncertainty region. To observe where the actual cross-correlation peaks occur, we upsample the signals with different factors (in these figures, we use a factor of 10). We observe a finer synchronization thanks to this approach, with around 60 ns median error.

Different Correlation methods

Fig. 3.4c compares the three different correlation methods explained in Section 3.4: `dphase`, `abs`, and `iq`. For clearness, we show only the results for DAB signal, but the results are very similar to other technologies. From this figure, we can see that `abs` and `dphase` have similar performances (although the former is slightly better), whereas the traditional `iq` method performs poorly on all the analyzed technologies. One of the reasons that explain the worse performance of `iq` method is the fact that it relies on both the magnitude and the phase of the samples, and phase alone is not a suitable descriptor in non-coherent sensor networks.

Comparison of technologies

The performance comparison of all the technologies analyzed is shown in Fig. 3.4d. From these results, we observe that digital signals offer the best performance and FM signals the worst. One explanation is that the former are digital signals with accurate amplitude information and richer phase information, whereas FM radio signals are analog with coarse amplitude information and no information encoded on the phase.

Furthermore, LTE, GSM, DVB-T and DAB have similar performance. We notice that LTE and DVB-T have larger bandwidths with respect to the IQ sampling rate of RTL-SDR front-end (spectrum sensors only measure IQ samples in 2 MHz bandwidths, with frequency tuned to the center frequency of the Reference and Unknown Signals), while the bandwidth of DAB is of about 1.5 MHz (thus it does not exploit fully the RTL-SDR capabilities). Furthermore, there are fewer TV and Digital Radio transmitters, so it is easier to guarantee that spectrum sensors capture signals from the same tower.

3.5.2 Deployments

For these experiments, we deploy sensors with the same characteristics as described in Fig. 3.2 in two different cities: Madrid and Zurich (Fig. 3.6). The maximum distance between sensors is 20 km and 15 km, respectively. To evaluate the performance of the full architecture, we select existing DAB transmitters as our target frequencies due to the high transmission power and well-defined locations, thus making them appropriate to benchmark the proposed system. For the sensors, we implement an extension to `es_sensor` C++ package from ElectroSense [16]. The resulting platform is similar to the one depicted in Fig. 3.2. For the localization backend, we implemented versions in MATLAB© and Python.



(a) Deployment in Madrid (b) Deployment in Zurich

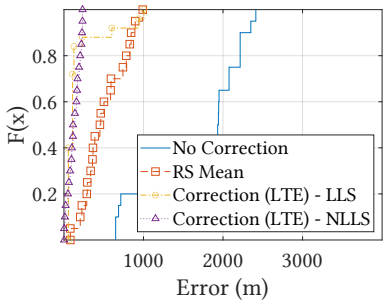
Figure 3.6: Sensor locations for the benchmark. The target transmitter (Unknown) is marked in blue and the Reference in pink. For the case of Madrid, both Target and Reference are the same.

Madrid deployment

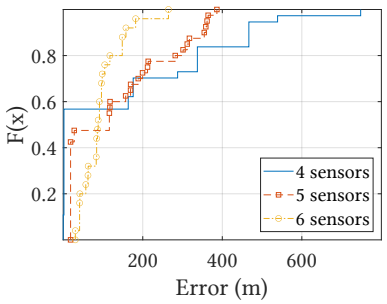
A total of six IoT sensors are deployed in the Madrid region, each having RTL-SDR v3 as their SDR frontend. The target transmission is the DAB signal with a center frequency of 208 MHz.

As our reference signal, we use the DAB transmission centered at 196 MHz coming from the same transmitter. Due to the limited number of DAB transmitters in the area of deployment, no other transmitters are interfering. By having reference and target on the same location, we can rule out inaccurate locations of transmitters. All the errors should be due to the limitations of the receivers and our approach. In total, up to 70 recordings over several months are gathered, each consisting of 1.5 seconds of data per sensor (divided into three continuous chunks containing RS, US, and RS respectively).

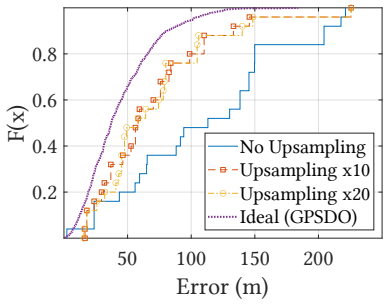
We first compare the results when we correct the frequency offset, Linear vs Non-Linear approaches, and present the results in Fig. 3.7a. We observe that each step of our



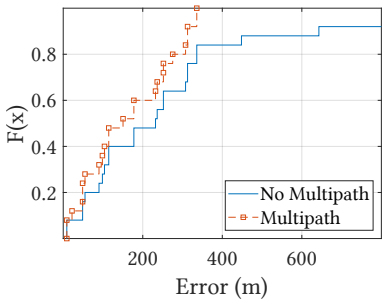
(a) FO correction (Madrid)



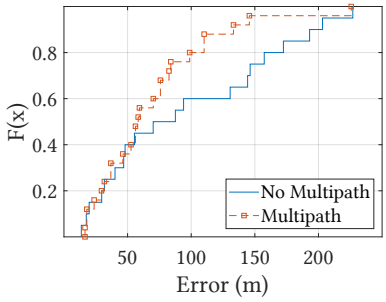
(b) Varying sensors (Madrid)



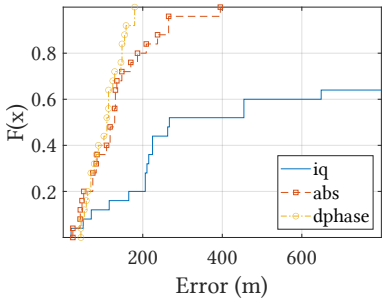
(c) Upsampling (Madrid)



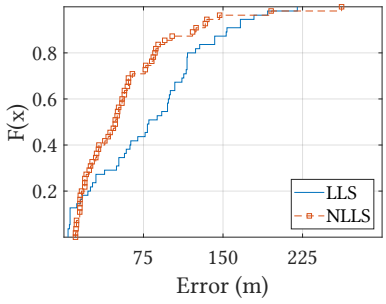
(d) Multipath 3 RX (Madrid)



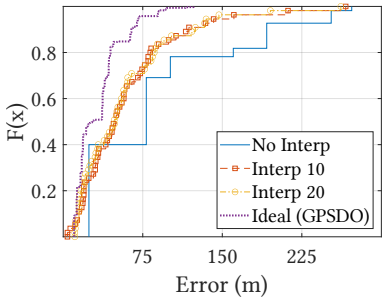
(e) Multipath 6 RX (Madrid)



(f) Corr Methods (Madrid)



(g) LLS/NLLS (Zurich)



(h) Upsampling (Zurich)

Figure 3.7: Results in real environments. The first 3 rows refer to the results in Madrid and the last to the results in Zurich. For the sensors in Zurich, we remove the results when no correction is done because the errors are much higher and would distort the visualization. The chosen correlation method is `dphase` since it proves to be the most accurate overall.

approach increases the overall accuracy of our system quite significantly. We observe that no offset correction implies km-level errors even though the transmitter is the same. If we make a rough estimate of the offset by averaging the two RS chunks, we can improve results notably. Correcting the offset provides the best results, and together with Non-Linear optimization methods, we can achieve a median accuracy of 97 meters, with all results located within 220 m of the target. When varying the number of sensors (see Fig. 3.7b), we observe that with fewer sensors, there are fewer synchronization issues so the lower tail of the ECDF is closer to almost perfect estimation. However, with lower sensors, single sample differences result in higher errors.

Next, we compare the effect of upsampling the correlation outputs and present the results in Fig. 3.7c. We observe that in this scenario, upsampling does improve results significantly, but there is not much gain from larger upsampling factors, which could be explained by the limits in CFO correction. In this case, we have a median accuracy of 50 m, and almost 90% of the results lie within 100 m. We compare this to a simulated scenario where sensors would have a commercial, yet expensive, GPSDO with a 50 ns timing alignment error [71]. We observe that our approach does not lie far from that ideal case.

We also compare the effects of multipath correction with 3 and 6 sensors and present the results in Fig. 3.7d and 3.7e, respectively. With 3 sensors, the correction does have a more significant effect, and we can improve the whole system accuracy by several 100 m on the distribution tail. For 6 sensors, since not all sensors are affected by multipath/NLOS equally, the localization algorithm helps to reduce the positioning error as the number of sensors increases, those effects are canceled out, but still, the multipath correction does have an effect on the overall accuracy.

Last, we also compare the different correlation methods and, as in Fig. 3.4, we see in Fig. 3.7f that `abs` and `dphase` methods have better results than using raw `iq` as the correlation input. Furthermore, `dphase` performs slightly better than `abs` in these real settings, particularly above the 60th percentile.

Zurich deployment

Up to three sensors are set in different locations of the Zurich region, two with the latest RTL-SDR v3 and one with the previous version. This is noticeable because the latter has an average frequency offset of 60 PPM. In this deployment, the target and the reference transmitters are two DAB signals located at different positions, with the first one transmitting at 194 MHz and the second one at 208 MHz. 100 measurements are collected with these sensors over two months with the same conditions as in the Madrid deployment.

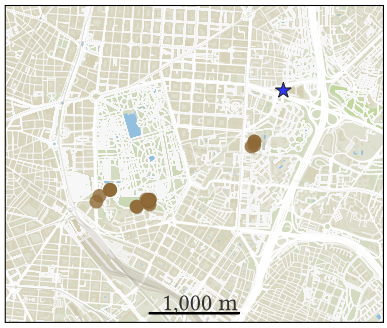
In Fig. 3.7g, we can see that, regarding the Non-Linear versus the Linear approaches, there is a significant difference between both methods. With Non-Linear methods, we can achieve a median accuracy of 50 m, 70% of the results within 65 m, and more than 90%

within 150 m, which is 1 sample difference at the sampling rates of these sensors.

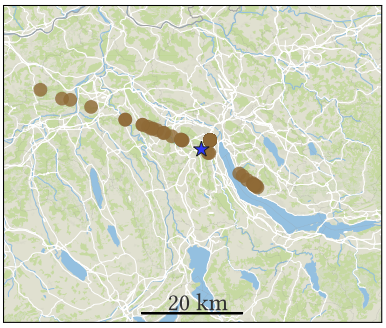
The effect of upsampling can also be seen in Fig. 3.7h, similar to the results in Madrid. Upsampling does allow for more fine-grained positioning compared to no interpolation. In the latter, one sample difference can result in a greater error in the distribution. Last, we can observe the distribution of estimations around the transmitter (Fig. 3.8). The spatial arrangement matches what we would expect theoretically with such sensor disposition. For instance, Zurich has only three sensors, and the geometry causes errors mainly on one axis.

3.6 Conclusion

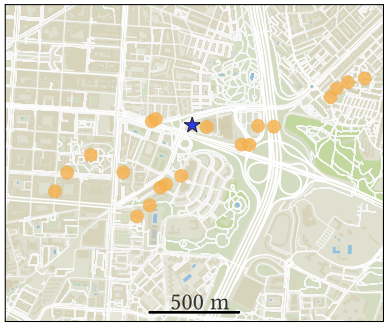
ORAN-Sense is a novel architecture that leverages low-cost IoT spectrum sensors and the O-RAN to perform localization of non-cooperative transmitters in the Next Generation of cellular networks, enhancing their current capabilities. Our system design considers aspects such as correcting sampling and frequency offsets with imperfect hardware and upsampling the signals. We also observed that digital signals provide much better performance than analog ones. In our real-world experiments in two European cities with IoT sensors spanning areas of tens of kilometers, we show that it is possible to localize signals with median accuracies of tens of meters and 1.5 s of data, and these results replicate over the cities we have studied, with little control over the position of sensors and big frequency offsets, as well as in the presence of multipath.



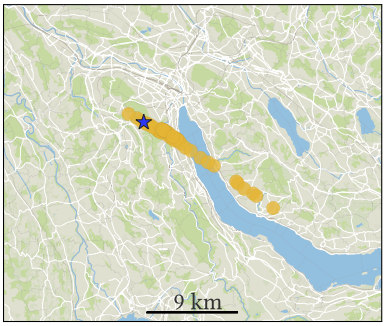
(a) Original (Madrid)



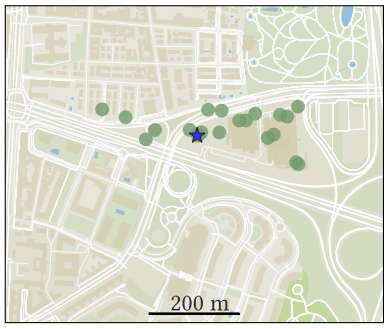
(b) Original (Zurich)



(c) Mean (Madrid)



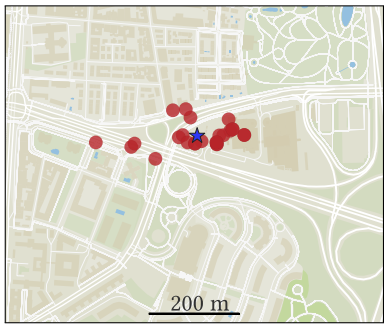
(d) Mean (Zurich)



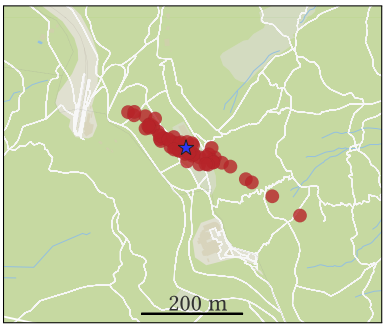
(e) FOC (Madrid)



(f) FOC (Zurich)



(g) Upsampling (Madrid)



(h) Upsampling (Zurich)

Figure 3.8: Positioning mapped for the cities of Madrid (first column) and Zurich (second column). We compare how the accuracy improves using no correction (orig), averaging reference signal (mean), frequency offset correction (foc), and correction+upsampling (ups). The transmitter position is marked with a blue star, and the dots represent the estimated locations.

4 SkyPos: Real-world evaluation of self-positioning with aircraft signals for IoT devices

4.1 Introduction

The literature on outdoor and indoor localization is vast. While indoor localization research makes use of different signals in the terrestrial radio frequency spectrum such as WiFi, UWB, Bluetooth, or Zigbee, outdoor localization is widely dominated by GNSS based positioning such as GPS, Galileo, GLONASS, and the like, to the point of becoming a commodity in all sorts of hardware and environment.

In recent years, aircraft signals have been proposed as an alternative means of positioning. In [40], they looked into opportunistically using aircraft signals (ADS-B) to perform indoor localization. In [63], this method has been refined to require less time for data collection, adding higher computational cost. Positioning with aircraft signals has several advantages over classical satellite-based systems. First, since aircraft signals are much stronger than satellite signals, they penetrate much better in buildings and work in obstructed or indoor environments. Second, getting a first position is much faster (under 3 seconds of data collection and a few minutes to obtain the fix in [63]). Finally, GPS based systems are highly vulnerable to spoofing attacks [72], [73] while positioning based on aircraft signals requires a more complex, distributed, and multi-device attacker setup [13].

In GPS spoofing scenarios, an additional source of location for verification purposes would increase the *location integrity* and quantify the trust of the provided position. Integrity is essential for systems where secure positioning is critical, such as vehicle to everything (V2X) communications, and it was standardized within the 3GPP in Release 17 and beyond for the design of 5G Advanced and toward 6G systems [74]. 3GPP has also envisioned several positioning applications and defines several positioning methods for 5G networks [75] that do not rely directly on GNSS information. There is also active research in improving accuracy and reducing latencies, as shown in [76].

Since [40], [63] laid the theoretical foundations for positioning based on aircraft signals a few years ago, we are not aware of any real-world deployed system making use of this localization technique so far. The actual performance one may get in the wild is thus yet

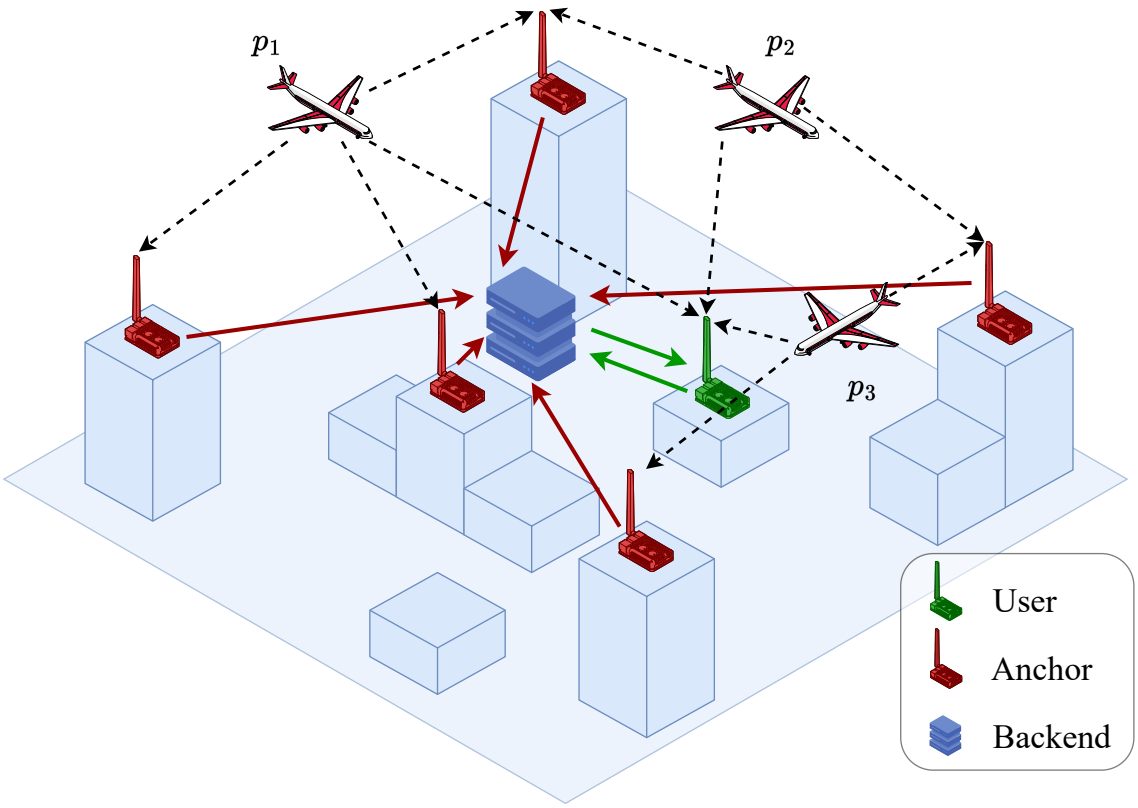


Figure 4.1: General overview of SkyPos. Several aircraft (p_1, p_2, p_3) pass through an area with anchors and the user that wants to self-localize. Each anchor (red) gathers the messages from each aircraft, timestamps them, and sends it to a central backend. The user (green) also collects messages and queries the backend. It receives the timestamps from the other anchors and self-localizes using the anchors' and its own timestamps.

not clear.

In this work, we report from SkyPos (Fig. 4.1), a first large-scale implementation that leverages thousands of crowdsourced ADS-B reference stations from the OpenSky Network [77] to position IoT devices across Central Europe. We systematically evaluate the performance of this real-world deployment comprising low-cost software-defined radios and higher-end ADS-B reference stations synchronized through GPS. We analyze the tradeoffs in measurement error, synchronization offset, clock drift, and sensor placement errors.

The contributions described in this chapter are the following:

- We introduce an architecture and an implementation for self-localization using aircraft signals that leverages the existing ADS-B crowdsourced sensors from the OpenSky Network;
- We provide an extensive evaluation of the architecture's performance and the accuracy of the localization with real-world data;
- We present an approach to solving the multivariate self-localization problem given the

measurement error, synchronization offset, clock drift, and sensor placement errors we find in our data, as well as insights into which aircraft to select to increase accuracy.

To the best of our knowledge, there has not been any attempt to design a real system for embedded sensors to self-localize using aircraft signals at a large scale, and the limited prior work in this field has addressed the problem only by means of simulations or in small setups with a few sensors [78]–[80]. In contrast, in our work, we perform our theoretical and experimental analysis with a large set of embedded sensors across Europe and provide a prototype architecture.

The chapter is divided as follows: First, we discuss the most relevant research regarding the opportunistic use of aircraft signals (Section 4.2). In Section 4.3, we describe the main challenges faced when designing the proposed system. Section 4.4 provides a quick primer on ADS-B and how it could be exploited for self-localization. Sections 4.5-4.6 provide the theoretical foundations and our proposed architecture, respectively. In Section 4.7, we evaluate our approach with data obtained from real sensors from OpenSky Network. Last, we discuss the main aspects of the proposed system in Section 4.8 provide some final remarks in Section 4.9.

4.2 Related Work

The literature regarding outdoor and indoor localization is vast. While indoor localization research uses different signals in the radio frequency spectrum, outdoor localization is widely dominated by GPS and the like. With techniques like Precise Point Positioning, it has been possible to achieve sub-decimeter level accuracies over the last 20 years [81].

However, since the position is computed at the receiver, GPS based systems are vulnerable to spoofing or jamming attacks [72], [73]. This area of research is currently very active, and even though there are some commercial solutions to avoid these issues, it remains an open challenge. Works to enable localization in GPS-constrained areas have been proposed in recent years via UAV swarms [82], [83], or integrating multiple information sources [84] for dense vehicular networks. However, the former requires specific and coordinated ad-hoc deployments or more complex changes in both infrastructure and users, as in the latter. This work explores leveraging existing signals and sensor networks and providing a location solution with little overhead for resource-contrained devices like a Raspberry Pi.

In recent years, the use of aircraft signals has drawn a great deal of attention to solve a variety of problems not only related to aircraft positioning. In [40], they looked into opportunistically using aircraft signals to perform indoor localization. However, their analysis only extended to a few sensors and did not consider scalability or further development of the proposed platform. In our research, we extend this analysis to hundreds of sensors and different topologies and provide results on the overall performance of the proposed architecture.

In [85], aircraft signals and their waveforms were exploited to obtain time of arrival measurements with less than a nanosecond time precisions. ADS-B signals have also been used to achieve sensor network synchronization with accuracies similar to the NTP, as it is stated in [79]. Recently, the research from [80] has looked into the privacy of sensor networks for avionics. In this last piece of research, the methodology used is similar to what is proposed, but most analysis is done in the simulation domain, and only one case with real sensors is presented.

4.3 Challenges

Self-positioning using aircraft signals poses significant challenges, as the specification was not designed with this use case in mind. We describe the main challenges in the following paragraphs.

System Design We show an overview of our system in Fig. 4.1. For our prototype architecture, several design choices have to be made. Unlike GPS satellites, aircraft do not send absolute timing information in their ADS-B messages. Therefore, to localize the device, we need several receivers to gather those messages. By considering the differences in arrival times among these sensors, it is possible to estimate the location of the desired sensor. We leverage existing air traffic surveillance networks like OpenSky [77], which already store aircraft positions in their backend received by ADS-B signals.

Each receiver only sends a list of ADS-B meta-data ($\{m_{p_1}, t^{r_1}_{p_1,A_1}\}, \{m_{p_3}, t^{r_3}_{p_3,A_1}\}, \ldots$), where the tuple $(m_{p_i}, t^{r_k}_{p_i,A_j})$ represents the message m sent by aircraft p_i and received by anchor A_j at time r_k. That meta-data can be composed of the fields present in Fig. 4.2. There are different ADS-B types, including position (obtained through the aircraft's own GNSS system), velocity, or status. We only collect those including position discarding the rest, thus reducing computation and storage requirements.

These messages are sent to the backend, reducing the data amount compared to sending the raw IQ signals. Besides, as we have access to a large set of data, we propose to rely on in-memory storage, which allows us to reduce query times. The user also would have to send a list of the aircraft messages collected in a period of time so that the backend can match the messages to those collected from reference sensors.

Synchronization and Timing constraints In time-based localization, good timing accuracy is a must. For users, we do not impose large timing constraints on the system clock (users in our study were using NTP, and is sufficient) nor on the timing of samples. Tools like `dump1090` [86] provide microsecond-level accuracy, which we show is sufficient.

For anchors/reference receivers on the other side, there are harder restrictions. We focus only on a subset of OpenSky's network sensors that have GPS-disciplined oscillators



Figure 4.2: Structure of an ADS-B message frame. The numbers in parentheses represent the number of bits of each field in the message.

that can provide $30\,ns$ accuracy. Using only a subset of all the sensors in the network also has the benefit of reducing the storage and bandwidth cost of the whole infrastructure.

Clock imperfections Receivers (specifically their RF frontends) are subject to imperfections in their internal oscillators which can decrease the localization accuracy. To overcome this, we propose a mathematical framework that relies on a few reference sensors with disciplined oscillators across the network. These *reference sensors* allow us to reduce the load when computing the position of users with lower-cost sensors.

Performance Validation Previous literature has relied heavily on simulations or controllable setups with few sensors. To have a more extensive analysis, we rely on LocaRDS [87], a dataset containing aircraft and sensor information from OpenSky [88], a well-known air traffic surveillance, for several hours and days. This dataset contains data from up to 700 IoT sensors spread all over Europe and with varying coverage.

4.4 ADS-B for positioning information

Traditional systems for air traffic surveillance include Primary Surveillance Radar (PSR) which are radars that detect the presence of aircraft by measuring radio wave reflections, and Secondary Surveillance Radar (SSR) which can request more information about their flight state. This is possible because aircraft include radar transponders that reply with the required information each time a ground station interrogates them. ADS-B can be regarded as an enhancement of SSR systems, which is also used to send flight status information, but in this case, without the need for interrogation (hence the term Automatic) [89]. ADS-B is steadily being adopted by many aircraft. Starting January 2020, all aircraft in the US were mandated to include ADS-B transponders, and Europe will follow this in 2023, thus making it a widely employed technology in aviation.

The downlink channel of ADS-B operates on the 1090 MHz band, and messages are encoded with Pulse Position Modulation (PPM). Each ADS-B frame is 112 bits long [89], and the frame structure is shown in Fig. 4.2. The International Civil Aviation Organization (ICAO) address is a 24-bit number that is used to uniquely identify an aircraft and the ADS-B message can be one of 9 types containing information regarding its position,

```
*8d3452c85851967ae91c9ae20503;
    - CRC: e20503 (ok)
    - DF 17: ADS-B message.
      - Capability     : 5
      - ICAO Address   : 3452c8
      - Extended Squitter  Type: 11
      - Extended Squitter  Sub : 0
        - F flag   : odd
        - T flag   : non-UTC
        - Altitude : 15225 feet
        - Latitude : 81268 (not dec)
        - Longitude: 72858 (not dec)
```

Table 4.1: Example of decoded ADS-B message containing aircraft position.

velocity, and status. Using common open source software like *dump1090*[6] we can capture, decode these messages and timestamp these messages (with relatively high precision even [85]), as it is shown in Table 4.1.

In recent years, several platforms have sprouted that, based on these ADS-B transmissions, provide real-time plane tracking and various related services. Examples of such platforms are flightaware[7] or flightradar24[8]. Within all these, a notable example is the Opensky-Network [77], [88], which is composed of a crowdsourced sensor network monitoring air traffic, also providing data for researchers in this field.

Recent literature shows that, apart from air traffic surveillance, it is possible to exploit aircraft messages for different purposes like synchronization [79], [85], positioning [40], [63] or privacy assessment [80]. The key insight is that by having multiple aircraft broadcasting status messages periodically, they effectively become an ad-hoc 'satellite network'. Moreover, aircraft travel at substantially lower altitudes than satellites, and messages are transmitted with relatively high SNR. Thus these signals can easily be received and decoded with low-cost off-the-shelf SDR and antennas. In Fig. 4.3, we show the number of distinct aircraft IDs for 3 sensors, 2 located indoors (one in an upper floor office with windows and the other in a lower floor with no windows) and 1 outdoors, received in a 24-hour window with a standard dipole antenna. Except for times at night (many airports close at night, thus reducing the air traffic volume), there are between 30-40 distinct aircraft IDs detected per second for the outdoor and the upper office sensor and up to 15 for the sensor in the office with no windows.

In these same places, we compare the GPS signal reception. For that, we use the gnss-sdr software [90] and an active GPS antenna. For the indoor locations, we could not obtain

[6] https://github.com/antirez/dump1090
[7] https://es.flightaware.com/
[8] https://www.flightradar24.com/

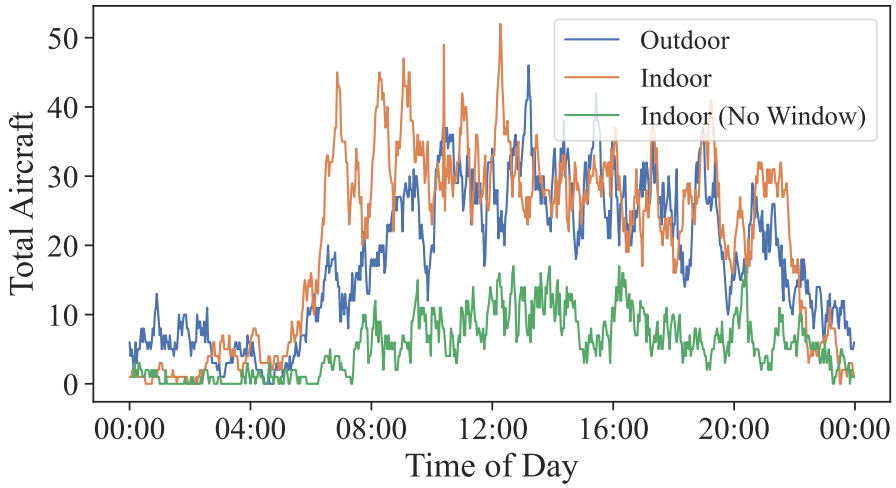


Figure 4.3: Number of distinct aircraft received by 3 sensors during the day, two indoor and one outdoor. Data was aggregated in 2-minute time windows

a fix at any point in time, whereas for the outdoor location, we could obtain a position at any point during the day. This further proves that we can exploit ADS-B information for self-positioning, but there could be hours at night where some accuracy might be lost due to fewer aircraft being present.

As mentioned in Section 4.1, aircraft do not provide the time the message was transmitted, so a sensor alone could not compute its position from multiple messages. However, with a network like OpenSky, where multiple sensors are receiving the same broadcasts, we exploit the difference in arrival times to provide positioning for other nodes. As mentioned earlier, we build on top of the work in [40] and extend it to cover more realistic scenarios with a greater variety of sensors and spatial distributions.

4.5 Methodology

There is a considerable amount of literature regarding the problem of aircraft multilateration (MLAT), where the objective is precise tracking of aircraft [91]–[93]. In contrast, in this paper, we consider the problem of positioning the user using aircraft signals, thus converting it into a *Reverse MLAT* problem. This problem was first formulated in [78].

The core idea of this method is for multiple receivers to decode aircraft messages which contain the aircraft's position, timestamp them with their local clocks, and send this information to a backend infrastructure for location processing. We can distinguish two types of receivers:

- **Target receiver** or **User**, whose position is unknown and to be estimated. The requirements are not as strict as for the reference receivers; in fact, virtually any equipment capable of decoding aircraft messages would qualify as a target receiver.
- **Reference receivers** or **Anchors**, which are accurately localized and have the necessary

equipment to correct their internal clock errors (offsets and drifts). These sensors are already part of an existing air traffic surveillance network (i.e., OpenSky [77]), which also reports aircraft locations to the backend.

4.5.1 User Localization

Let us define the 3D distance between a static user $\mathbf{u} = [x_u, y_u, z_u]$ and an aircraft position $\mathbf{p_i^r} = [x^r_{p_i}, y^r_{p_i}, z^r_{p_i}]$ at any time r as:

$$d^r_{u,p_i} = ||\mathbf{u} - \mathbf{p_i^r}||_2 \tag{4.1}$$

Every time an aircraft transmits an ADS-B message, the time at which the user receives it (in its local time) can be written as:

$$\hat{t}^r_{u,p_i} = t^r_{p_i} + \frac{d^r_{u,p_i}}{c} + \theta_u + \delta_u \cdot (\Delta t^r_{u,0}) + \varepsilon_u \tag{4.2}$$

being $\hat{t}^r_{u,p_i}$ the time at which the user received the message, $t^r_{p_i}$ the absolute time at which the message is transmitted by the aircraft, θ_u and δ_u are the receiver's clock offset and drift respectively, and ε_u additive white gaussian noise corresponding to measurement error. $\Delta t^r_{u,0}$ is the elapsed time from an absolute initial reference t_0 to the current sensor time t^r_u.

Assuming that the aircraft's position is known (ADS-B signals continuously broadcast this information), we have a total of 6 unknowns in our system: the 3D coordinates of the receiver, offset and drift of the receiver's local oscillator, and the transmission time of the message. Since the transmission time is unknown, multilateration systems take another receiver (υ) and subtract 4.2 for both receivers:

$$\tau^r_{u,\upsilon,p_i} = \frac{d^r_{u,p_i}}{c} - \frac{d^r_{\upsilon,p_i}}{c} + \theta_{u,\upsilon} + \delta_u \cdot \Delta t^r_{u,0} - \delta_\upsilon \cdot \Delta t^r_{\upsilon,0} + \varepsilon_{u,\upsilon} \tag{4.3}$$

To solve for all the unknowns present in 4.3, we would at least need 10 different measurements. To reduce the amount of data, the approach in this paper uses **reference receivers**, whose GPS coordinates are known and which possess stable oscillators so that their clock drift is negligible. Thus, 4.3 becomes (instead of υ we will represent reference receiver j as A_j):

$$\tau^r_{u,A_j,p_i} = \frac{d^r_{u,p_i}}{c} - \frac{d^r_{A_j,p_i}}{c} + \theta_{u,A_j} + \delta_u \cdot \Delta t^r_{u,0} + \varepsilon_{u,A_j} \tag{4.4}$$

We can collect all measurements for the target receiver u. The cost function is formed by the sum of the p-norm of all the residuals (where p can be any order of the norm, throughout this text, we will use the 2-norm). We then obtain the values for $S(\mathbf{u}, \theta_u, \delta_u)$ as:

$$\begin{aligned} S(\mathbf{u}, \theta_u, \delta_u) &= \frac{1}{2} \sum_{\forall A_j, p_i, r} (\tau^r_{u,A_j,p_i} - \hat{\tau}^r_{u,A_j,p_i})^2 \\ &= \frac{1}{2} \sum_{\forall A_j, p_i, r} \eta^2_{u,A_j,p_i,r} \end{aligned} \tag{4.5}$$

where τ^r_{u,A_j,p_i} is the right hand side of 4.4. The objective is to minimize the cost function S. To achieve this goal, we compute the gradients for each of the terms (3D coordinates x_u,y_u,z_u, offset θ_u and drift μ_u) as:

$$\frac{\partial S}{\partial x_u} = \sum_{\forall A_j, p_i, r} \eta_{u,A_j,p_i,r} \cdot \frac{(x_u - x^r_{p_i})}{c \cdot d^r_{u,p_i}} \tag{4.6}$$

$$\frac{\partial S}{\partial y_u} = \sum_{\forall A_j, p_i, r} \eta_{u,A_j,p_i,r} \cdot \frac{(y_u - y^r_{p_i})}{c \cdot d^r_{u,p_i}} \tag{4.7}$$

$$\frac{\partial S}{\partial z_u} = \sum_{\forall A_j, p_i, r} \eta_{u,A_j,p_i,r} \cdot \frac{(z_u - z^r_{p_i})}{c \cdot d^r_{u,p_i}} \tag{4.8}$$

$$\frac{\partial S}{\partial \theta_u} = \sum_{\forall A_j, p_i, r} \eta_{u,A_j,p_i,r} \tag{4.9}$$

$$\frac{\partial S}{\partial \mu_u} = \sum_{\forall A_j, p_i, r} \eta_{u,A_j,p_i,r} \cdot \Delta t^r_{u,0} \tag{4.10}$$

With these equations, it is possible to obtain the position, offset, and drift of the target with optimization methods widely employed in the literature like Levenberg-Marquardt (LM) [45], BFGS [94], or Gradient-Descent and its variants [95] among others. In practice, we observe little difference between these methods for this particular problem.

4.5.2 Reference Receiver Offset Estimation

One last step to reduce the amount of data needed is to precompute the reference receiver's clock offset, leaving only the parameters belonging to the user as unknowns. To estimate the offset for the reference receivers, we start from a similar equation as 4.4, noting that the drift for both receivers is negligible, that is, $\delta_u \approx 0;\ \forall u$. We can then transform the equation into the following:

$$\tau^r_{A_j,A_k,p_i} = \frac{d^r_{A_j,p_i}}{c} - \frac{d^r_{A_j,p_i}}{c} + (\theta_{A_i} - \theta_{A_k} + \varepsilon_{A_i,A_j}) \tag{4.11}$$

Or in its matrix form:

$$\boldsymbol{\tau} = \mathbf{d} + \mathbf{F} \cdot \boldsymbol{\theta} + \boldsymbol{\varepsilon} \tag{4.12}$$

Where $\hat{\boldsymbol{\tau}}$ and $\mathbf{d}$ are $M \times 1$ vectors (where M is the number of aircraft messages) containing the measured and the geometrical TDOA:

$$\hat{\boldsymbol{\tau}} = \begin{bmatrix} \cdots & \hat{\tau}^r_{A_j,A_k,p_i} & \cdots \end{bmatrix}^T \quad \forall i = 1 \dots M \tag{4.13}$$

$$\mathbf{d} = \begin{bmatrix} \dots & \frac{d^r_{A_j,p_i}}{c} - \frac{d^r_{A_k,p_i}}{c} & \dots \end{bmatrix}^T \quad \forall i = 1 \dots M \tag{4.14}$$

Matrix $\mathbf{F}$ is an $M \times N$ matrix with N being the number of sensors with 1 and -1 at the columns of sensors u and v respectively:

$$\mathbf{F} = \begin{bmatrix} 0 & \cdots & 1 & \cdots & -1 & \cdots & 0 \\ 0 & 1 & \cdots & -1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & \cdots & & -1 & 0 \end{bmatrix} \tag{4.15}$$

Finally, θ and ϵ are vectors of size $N \times 1$ containing the unknown offsets and noise, respectively.

Generally, this system of equations yields an over-determined system that can be solved via Least Squares, returning the offsets for the reference receivers in 3.8:

$$\hat{\boldsymbol{\theta}} = (\mathbf{F}^{\mathbf{T}}\mathbf{F})^{-1}\mathbf{F}^T(\hat{\boldsymbol{\tau}} - \mathbf{d}) \tag{4.16}$$

4.5.3 Dilution of Precision (DOP)

DOP has been a popular metric to evaluate the precision of localization systems for both TOA [96] and TDOA [97] approaches (which include GPS systems) for the last decades. In recent years, DOP has been proposed in several localization systems as a method to estimate optimal receiver configuration [98]. In essence, DOP and its variants quantify the expected error amplification between the estimated and the actual position of the object that will be localized. To evaluate the DOP for our scenario, we take 3.8 and calculate the Jacobian (for the sake of simplicity, we removed the drift):

$$\frac{\partial \tau^r_{u,A_j,p_i}}{\partial(\mathbf{u}, \theta_u)} = \begin{bmatrix} \frac{x_u - x^r_{p_i}}{d^r_{u,p_i}} & \frac{y_u - y^r_{p_i}}{d^r_{u,p_i}} & \frac{z_u - z^r_{p_i}}{d^r_{u,p_i}} & 1 \end{bmatrix} \tag{4.17}$$

We can construct the matrix $\mathbf{H}$ with several aircraft that contain the Jacobian for all measurements. We notice that this equation is actually that of the DOP for TOA [96], thus if we calculate the matrix $\mathbf{Q}$ by:

$$\mathbf{Q} = (\mathbf{H}^T\mathbf{H})^{-1} \tag{4.18}$$

This matrix will contain the DOP values per each of the directions and offset (this method can be extended to include drifts):

$$\mathbf{Q} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & \sigma_{y\theta} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & \sigma_{z\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_{z\theta} & \sigma_\theta^2 \end{bmatrix} \tag{4.19}$$

With this matrix, we can estimate how precise our position is based on the selected aircraft per direction in the 3D space. It is also interesting to note that the receiver's position does not depend directly on the chosen anchors, which was experimentally observed in [78]. However, having more anchors makes it possible to obtain a more diverse set of aircraft (i.e., aircraft surrounding the sensor and not in a specific region), which can reduce the uncertainty in the position estimation.

4.6 SkyPos: Self-Positioning for IoT devices

We propose to use the collected aircraft information and to provide users with an on-demand self-positioning service. The architecture is illustrated in Fig. 4.4, and it consists of two main components: the **Backend**, which collects data from the anchors and stores their messages, and the **User**, which collects its own messages and queries the Backend for the collected messages and then attempts to perform its self-positioning.

4.6.1 Backend

Database

The database contains two indices, from (i) the sensors and (ii) the aircraft, respectively. Their role is as follows:

- Within the sensors index, we store information on the sensor location and serial number.
- In the aircraft index, we find the message hash, the corresponding aircraft's ICAO address (a unique identifier per aircraft), the time it is processed at the server, the location of the aircraft at the transmit time, the sensors that receive the message and the local timestamp at which each sensor processes the message.

This database is continuously updated with OpenSky Network's data, with older data being regularly erased so that the storage requirements are reduced. To reduce the query times, we run in-memory database instances in Redis. The sensors that are present in the database are not all of the ones present in the OpenSky Network. As mentioned in Section 4.3, we select a subset of "trusted" anchors that also incorporate GPSDO for higher precision timestamping. The precision of these devices is 50 ns, which is enough for several localization applications [71].

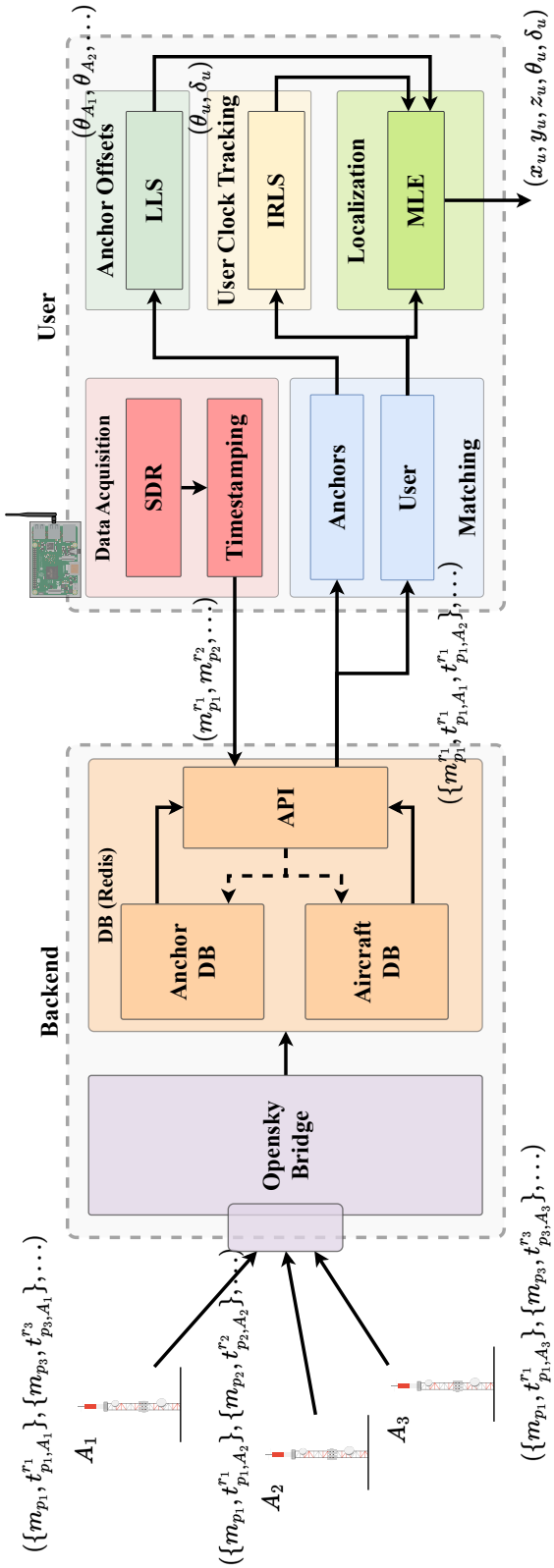


Figure 4.4: SkyPos proposed architecture overview.

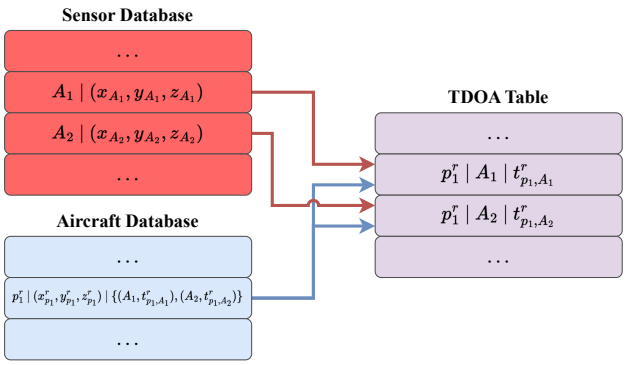


Figure 4.5: TDOA table pre-processing from aircraft and sensor databases.

API

The API gets the user request and processes the data as described in the remainder of this section.

After unpacking all the messages, it constructs a query for the aircraft index using the ICAO address. When results arrive, distinct sensor identifiers are selected, and another query is made to the sensors index. Once the results of both queries are gathered, data is prepared in the following manner: A table is constructed where each row represents the time difference of arrival measurement between the user and each of the anchors that have received the message. Each row contains the aircraft's ID, the reference receiver's serial number, and the measured TDOA in seconds. If a message m_i is received by N_{m_i} reference sensors, then N_{m_i} rows are added to the table. Thus, if the total number of valid messages is M, this table will have $\sum_{m_i=1}^{m_M} N_{m_i}$ rows (notation as seen in Fig. 4.1). The main reason for constructing such a table is to reduce the time in the optimization step since otherwise it would need to be constructed every iteration of the optimizer, slowing down the computation considerably. The whole process can be visualized in Fig. 4.5.

Serial	**Latitute**	**Longitude**	**Altitude**	**Type**	**Verified**
Integer	Float	Float	Float	String	Bool

Table 4.2: Sensor dataset structure

Message ID	**Time at Server**	**Aircraft ID**	**Latitude**	**Longitude**	**Baro. Altitude**
Integer	Float	Integer	Float	Float	Float
Geo. Altitude	**Num. Measurements**	**Sensors**	**Timestamps**	**RSS**	
Float	Integer	Array {Int}	Array {Float}	Array {Float}	

Table 4.3: Aircraft dataset structure

4.6.2 User

Data collection

The user or receiver is the service's client that wants to self-localize. The main requirements for the user are to possess a suitable receiver tuned at the 1090 MHz band and software able to decode ADS-B messages and timestamp the gathered messages. This can be easily achieved today with an inexpensive RTL-SDR receiver, some open source software like *dump1090* [86], and an embedded machine supporting such hardware and software (most widely available operating systems fulfill such requirements). The tool reports data in the format shown in Fig. 4.1 which is sent together with the timestamp at which it was received by the sensor. The precision achievable with a common board like the Raspberry Pi is $1\ \mu s$, which would be too high to achieve meaningful accuracy. However, by using higher-precision anchors, it is possible to overcome this limitation.

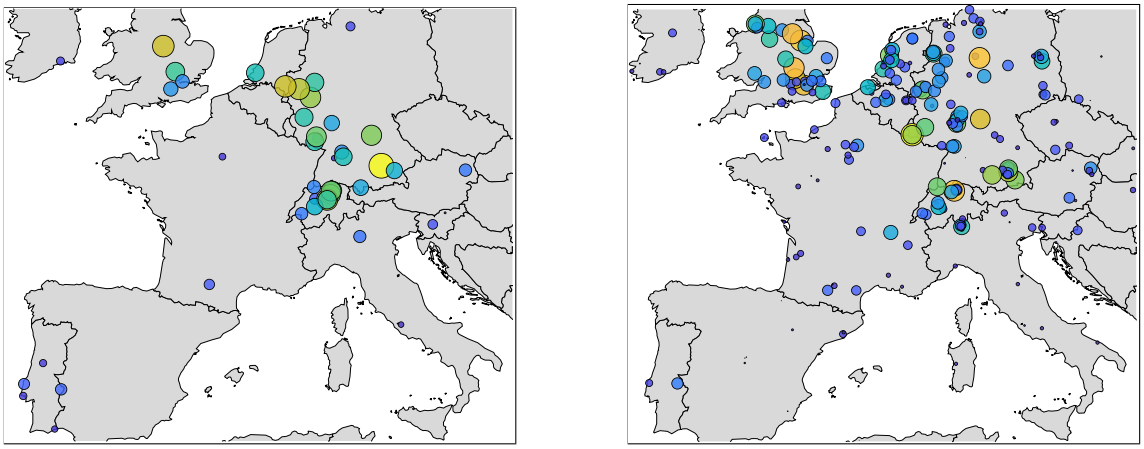
Fulfilling the aforementioned requirements, the user first collects and decodes aircraft messages, including the aircraft's ICAO address and its latitude, longitude, and altitude (see Fig. 4.1). After collecting messages for a certain time, the user serializes and sends the data to the service's endpoint through the API.

Localization

The user receives the data of the anchors from the API as a stream of $(m_1, t_1^{A_1}, m_1, t_1^{A_2}, \ldots)$, where m_i is the ith received message and $t_i^{A_j}$ is the time at with anchor j received that message (see Fig. 4.4). It then splits the process into the steps mentioned in Section 4.5. Data from anchors are used to estimate their offsets via linear estimators. In parallel, the user performs an initial estimation of its clock offset and drift. Depending on the type and quality of the receiver, measurements might be noisy, so we use the Iteratively Reweighted Least Squares (IRLS) technique [99] to obtain better initial estimations. After preprocessing, the user's position r is obtained via any of the methods described in Section 4.5.

4.7 Evaluation

The evaluation is performed in 2 different stages: In the first stage we utilize an already collected dataset [87], that contains millions of ADS-B messages from several days and hundreds of sensors. This allows us to demonstrate the feasibility of our approach. The second phase is focused on assessing the performance of SkyPos' architecture. For that, we use the same dataset and simulate the conditions of operation.



(a) Radarcape sensors

(b) dump1090 sensors

Figure 4.6: Sensor locations across Europe. The circle size and color represent the number of messages received per sensor.

4.7.1 Dataset

The dataset for our evaluation is described in LocaRDS [87], a dataset from the Opensky Network researchers that provides information on thousands of real aircraft collected by hundreds of sensors operated by volunteers across Central Europe.

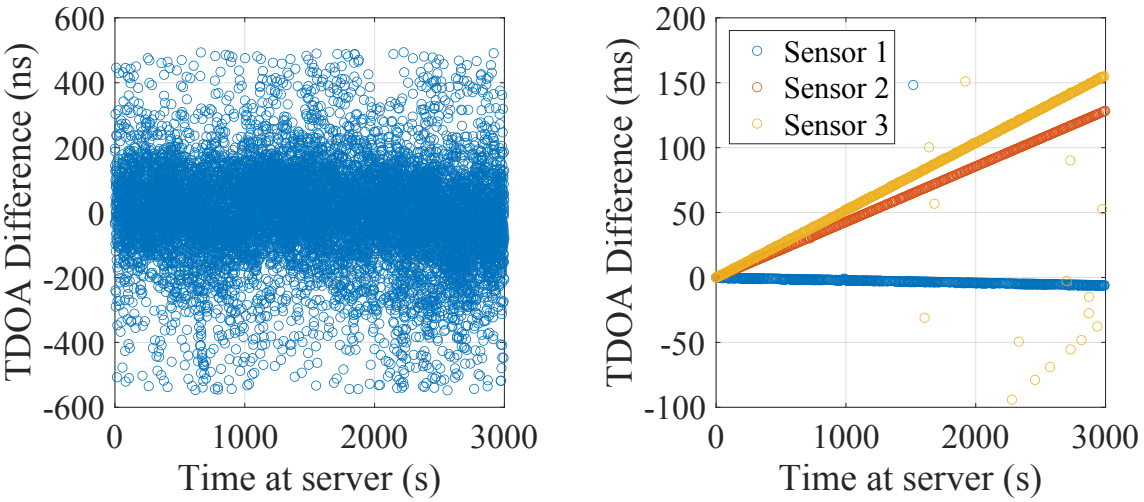
The data structure for sensor and aircraft databases is shown in Tables 4.2 and 4.3 respectively. The *Verified* field in Table 4.2 refers to whether the sensor position is verified by its operators. For our evaluation purposes, we will only use as reference sensors those that have been verified.

4.7.2 Evaluation of the dataset

To perform the evaluation, we first split the dataset between 2 types of sensors: *Radarcape* and *dump1090* sensors (their distribution across Europe is shown in Fig. 4.6). These sensors have different properties and are thus evaluated separately to understand the impact of the sensor type better.

Radarcape sensors

The first part of our evaluation is done with Radarcape sensors [100]. These sensors are specially designed to receive ADS-B messages. They incorporate GPS receivers and GPSDO, meaning their clocks remain stable within tens of nanoseconds. As mentioned earlier, we used only sensors that had verified positions, so for this set of experiments, we had 37 sensors shown in Fig. 4.6a. The density of such sensors is especially high in Switzerland and Germany, and it is also where these sensors receive more messages, as can be observed (a larger circle means a larger number of messages are received).



(a) Anchor clock offset (Radarcape sensors).

(b) User clock offset (dump1090 sensors).

Figure 4.7: Reference receivers' clock offsets over time and compared to each other. 4.7a shows the difference in geometric/measured TDOA for 2 radarcape receivers. 4.7b shows the offset over time for 3 sensors running the dump1090 software.

The fact that these sensors have GPSDO is more notable when computing the difference between the expected TDOA between 2 reference sensors and the measured TDOA as it is shown in Fig. 4.7a, where over the course of one hour, those differences remain constant except for variations due to measurement noise. Using Least Squares to compute the mean offsets (using one of the sensors as the absolute time reference), we can observe that those offsets remain within the $\pm$200 ns.

To perform the evaluation, we take one of the Radarcape sensors and randomize its position. We then compute the offsets for the remaining sensors used as a reference and select the 5 sensors with the highest common message count with the target receiver. The number of anchors can vary depending on the user's position with respect to them. In general, and as mentioned in Section 4.5, the number of anchors does not directly affect the performance, and we did not see much improvement from adding more anchors. This matches the observations shown in [78]. We then compute the position of the receiver with that data.

Fig. 4.8 shows how correcting the offsets in the model, using the estimation described in Section 4.5, Eq. 4.15, already has a notable effect on the accuracy. For the 2D scenario (latitude and longitude), the median error when considering the offset is 21 m, whereas it reaches 24 m when not considered. Moreover, in the former case, all the sensors are localized within a 41 m error and a 65 m error in the latter. We select 21 sensors with at least 10 other anchors and vary the number to observe the effect on accuracy. The results match what we predict in Section 4.5: The number of anchors does not significantly improve accuracy (even with 1 anchor, we can still maintain a median accuracy of 15 m).

It is also worth commenting that the errors for the 3D case (latitude, longitude, and altitude) are higher due to the receivers and aircraft geometry. The DOP is much higher

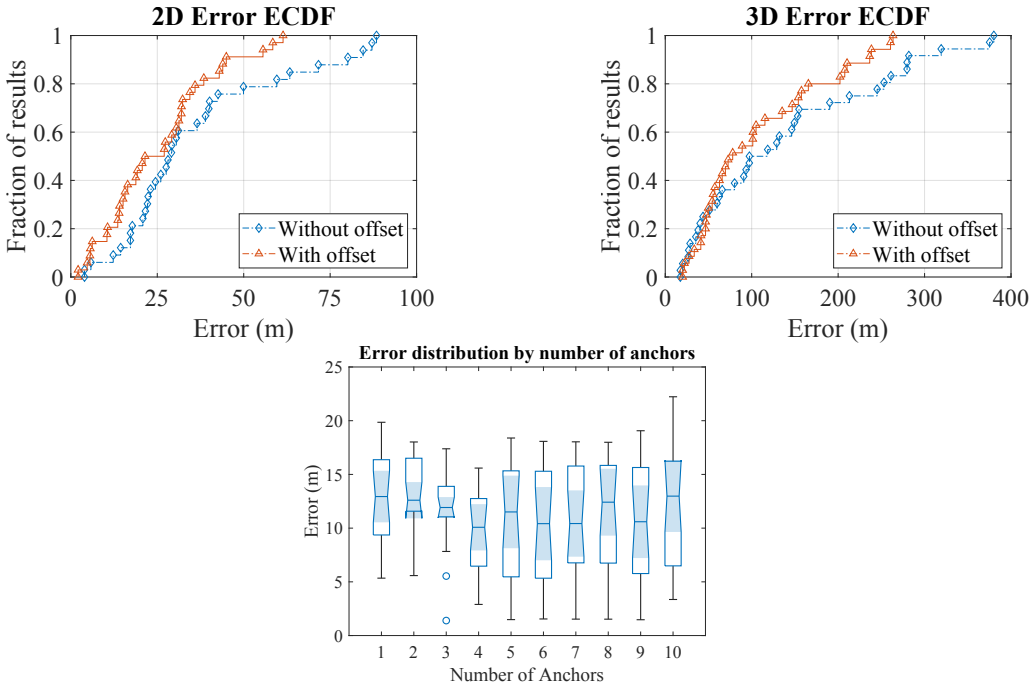


Figure 4.8: On top: The accuracy changes when clock offset is considered and when not. Below: How error is distributed when varying the number of anchors used for self-localization (for the 21 sensor subset).

in the vertical axis than in the horizontal axes, and thus errors propagate faster. This is better visualized in Fig. 4.9. Because distances on the ground plane are larger than vertical distances (100 km separating receivers and aircraft maximum altitude is around 10 km), small variations in the delay estimation generate larger uncertainties in the vertical axis (Δz) than in the horizontal axis (Δx). Despite this, the tendency is similar, and considering offset improves its counterpart notably in this case.

For our next set of experiments, we look into how the number of messages impact the location accuracy. In the experiments shown in Fig. 4.10a, we randomly select a fraction of the received messages (to simulate lower message rates). It is noticeable how even using 1% of the number of messages can lead to a similar accuracy rather than using the entire message dataset.

We show the 25th and 75th percentiles (black lines) and median (red dashed line) for the amount of time and localization accuracy, respectively, in Fig. 4.11. In these figures, it can also be seen that 1% of the data is a 'sweet spot' where the amount of time taken to process the location cannot be greatly reduced and where the location accuracy does not get impacted very much.

In the time between Fig. 4.10a and Fig. 4.10b, we also modify several optimization parameters, which result in the median error being reduced to 13 m, and we are also able to localize 90% of the sensors within 25 m.

In Fig. 4.10b, we perform a similar experiment, but instead of randomly selecting a

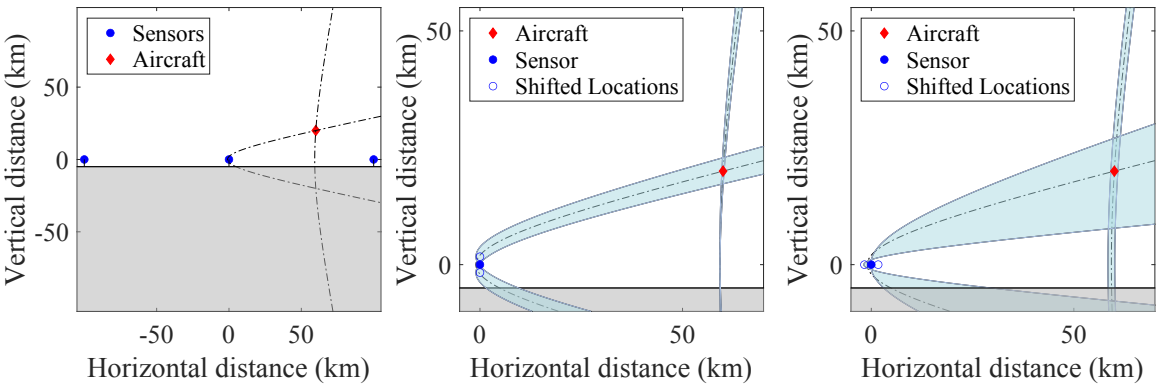


Figure 4.9: 2D visualization on why errors in altitude are generally larger than errors in the horizontal plane for the problem of self-localization with aircraft. The figure in the center represents a characteristic scenario for 3 sensors (in blue) and an aircraft (in red) flying at 10km altitude. In the rightmost figure, we show how an error of 1km in the vertical plane leads to a smaller error ϵ than when errors are on the horizontal plane. Thus, from the numerical point of view, the optimizing routine will converge faster to the correct horizontal position than the vertical one.

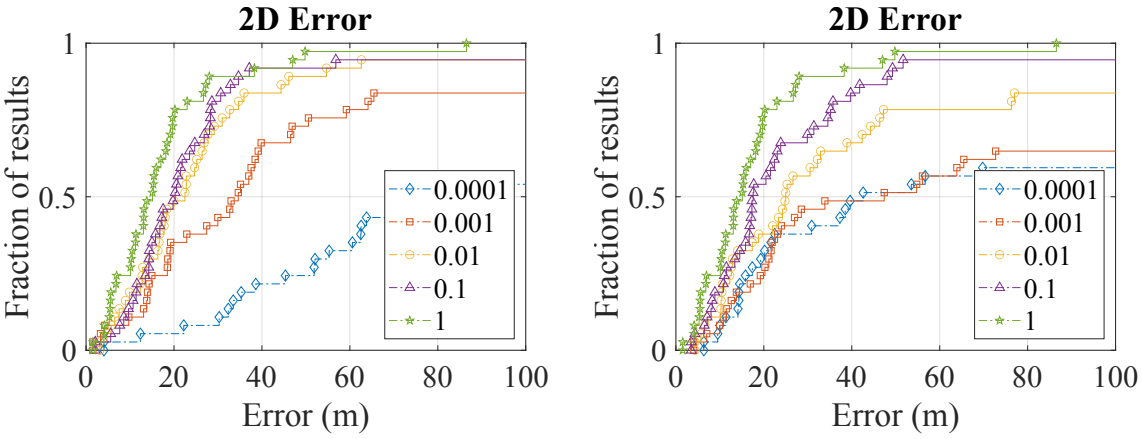
fraction of the messages, we select contiguous messages to observe the effect of short captures on accuracy. In this case, depending on the number of messages, we can observe that accuracy varies more noticeably. One of the reasons why accuracy drops at a larger rate could be that on shorter, contiguous streams of messages, fewer aircraft are seen, and the locations received are close to each other, thus worsening the DOP.

dump1090 sensors

When analyzing sensors running the dump1090 software only, from Fig. 4.7 we can observe that the offsets and drifts are much higher, in the order of tens of seconds, and looking at drifts, we can deduce 2 categories: receivers with drift of less than 2 PPM, which matches the specification for the version 3 of the popular RTL-SDR dongles [101]; and receivers with drifts in the order of tens of PPM which are older versions of the same SDR type.

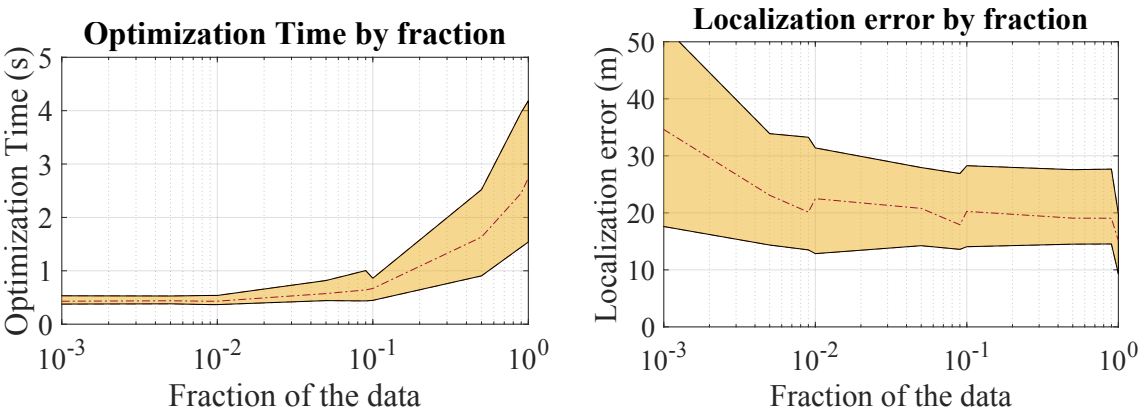
Fig. 4.12 and Fig. 4.13 show the results when using the whole dataset. For both the 2D and 3D cases, the positioning error is significantly higher than when using Radarcape sensors, with median errors of 450 m and 2000 m, respectively. The fact that offsets and drifts are higher could be one of the reasons to explain this accuracy decrease, but there are other factors. These sensors lack GPS, thus relying on the user to manually input their location, which might be inaccurate for a number of reasons. The ground truth could therefore be biased. Also, many of these receivers are located in interior spaces, which may also add errors due to sensors not seeing as many aircraft or due to multipath effects.

We asked 8 users to verify their positions and their RF frontend. The returned predictions are shown in Table 4.4. One important aspect is that the offset and drift are orders



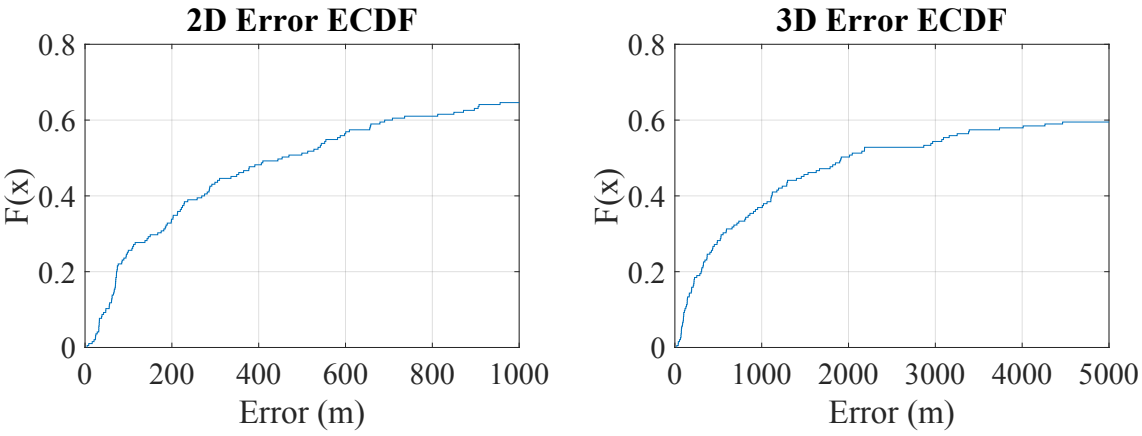
(a) Varying Message rate (b) Varying Time window

Figure 4.10: Error varying message rate and time windows.



(a) Time vs messages (b) Accuracy vs messages

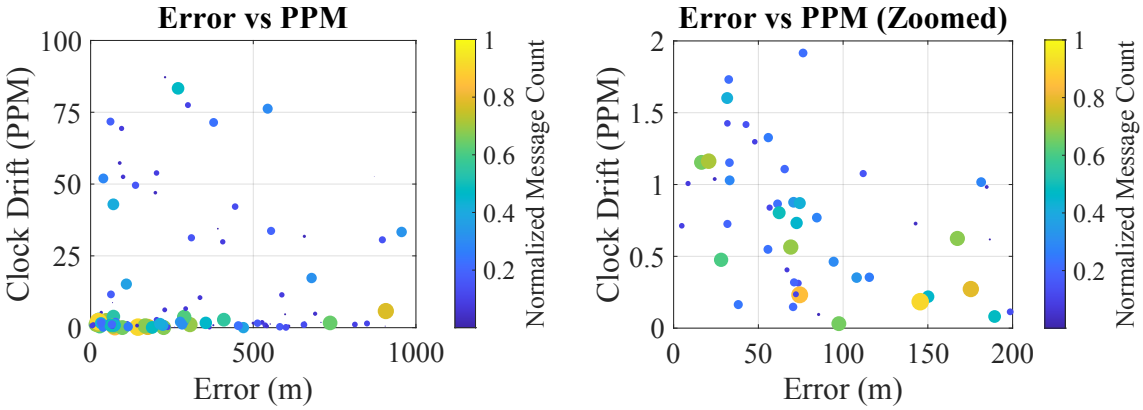
Figure 4.11: Time and accuracy varying the amount of data.



(a) 2D error (b) 3D error

Figure 4.12: Results for dump1090 sensors.

of magnitude bigger than the position parameters. Thus, choosing a good starting point is key to obtain an accurate position of the sensor. The naive approach, looking at Fig. 4.7b,



(a) All Recievers (b) Zoomed

Figure 4.13: Error with respect to drift and message count.

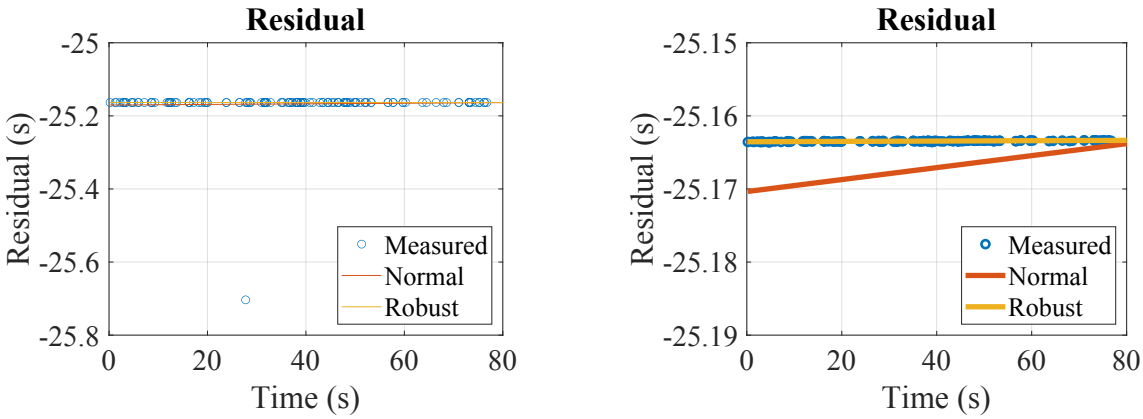


Figure 4.14: Normal vs Robust Estimation of the residuals.

would be to perform a linear regression and use the intercepts as the starting points for the algorithm.

However, methods based on naive linear regression can decrease the performance in the presence of outliers. To avoid this issue, we decide to use IRLS [99], which consists in adding weights to all measurements and iteratively updating the weights and intercept until the change is less than a specified tolerance. The difference between both approaches can be seen in Fig. 4.14, where for that particular sensor of the dataset, the presence of an outlier tilts the intercept estimation, whereas, for the robust case, it can discard it from the offset/drift estimation.

Both in Table 4.4 and Fig. 4.15, we can see the difference in results between using normal and robust initialization techniques. For the former case, the median error is around 100 m, with a few sensors above that value. In the case of robust initialization, 5 out of 8 sensors can be accurately localized within 30 m.

Another interesting fact is that it is possible to capture the device type with the drift parameter. For example, 3rd user's receiver, as per the documentation [101], has an oscillator offset of 0.5 PPM, which is indeed what we can measure with our approach. The

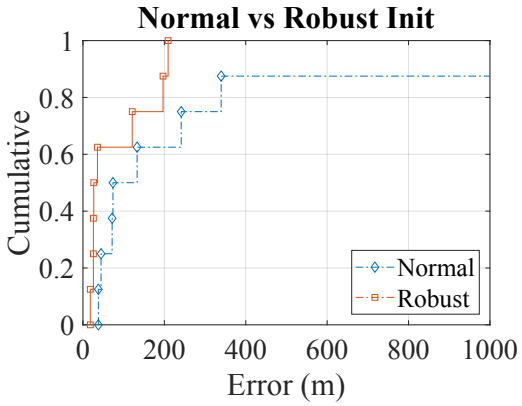


Figure 4.15: ECDF of the error using normal vs robust initialization

same case happens for the RTL-SDR receivers.

As for the offset, we have not explored the synchronization possibilities since our dataset contains no ground truth for the actual time. However, it is readily implemented as well. Therefore, simultaneous localization and synchronization could be a future line of research.

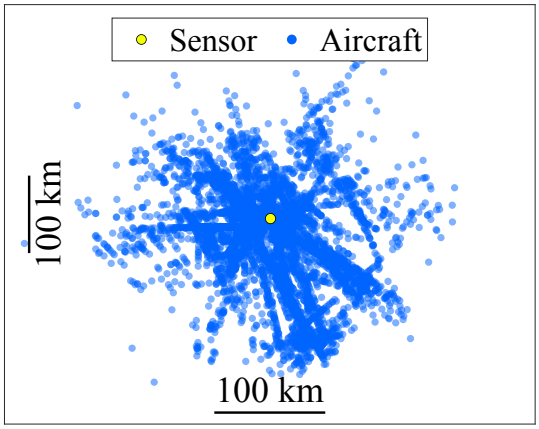
4.7.3 Intuition on Aircraft Selection

One important factor in assessing the quality of a localization system is to estimate how uncertain the position is. That uncertainty is primarily driven by which and how many aircraft are chosen to perform the localization, analogous to GPS satellites. In Fig. 4.16, we show the captured aircraft by a sensor in an hour window and 3 different scenarios: aircraft messages captured within 10-20 km, within 30-40 km, and clustered around a small region 40 km away from the sensor. The results vary notably in each scenario.

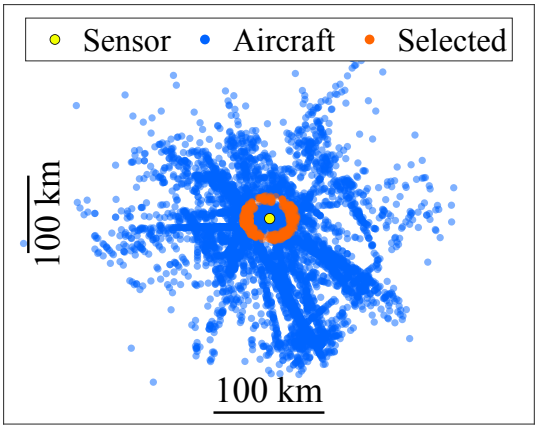
In the former, the vertical DOP is the lowest, but the uncertainty in the planar directions

Type	Location	$\mathbf{E_{nor}}$ (m)	$\mathbf{E_{rob}}$ (m)	Drift (PPM)
RTL-SDR	Indoor	339.60	18.07	1.9985
RTL-SDR	Indoor	133.08	121.10	38.7784
FlightAware	Outdoor	71.65	25.59	0.5646
RTL-SDR	Outdoor	241.42	97.49	63.1380
RTL-SDR	Indoor	73.72	196.81	42.9413
RTL-SDR	Outdoor	44.48	25.63	1.3269
RTL-SDR	Indoor	37.83	26.58	51.9510
RTL-SDR	Indoor	1140.12	35.64	63.1393

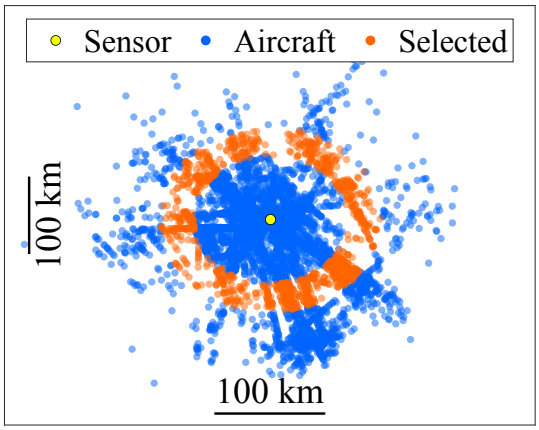
Table 4.4: Predicted positions for verified sensors. $\mathbf{E_{nor}}$ refers to positions obtained using regular least squares to predict the offset, whereas $\mathbf{E_{rob}}$ refers to the robust initialization method via IRLS



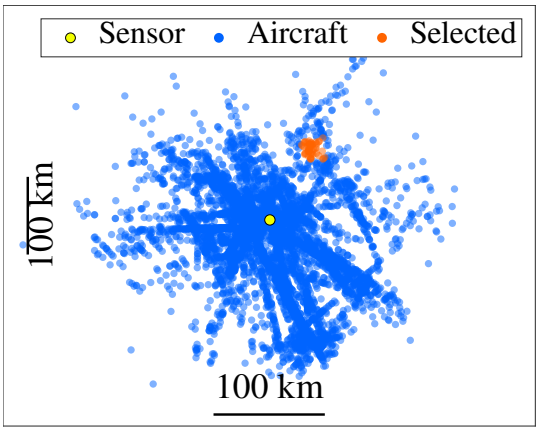
(a) Collected aircraft (blue) in an hour by a receiver (yellow)



(b) Bad Horizontal Dilution of Precision (HDOP) example



(c) Good DOP example



(d) Worst-case DOP example

Figure 4.16: Capture of a sensor and the measured aircraft during an hour. Except for the upper left image, in each we show different scenarios when selecting aircraft that affect DOP.

increases. On the other hand, for aircraft within 30-40 km, we observe a reduction in the planar DOP and an increase in the vertical. The positioning errors reflect the expected uncertainties in each of the directions. In the last case, where messages are clustered in a small region, we measure DOP values > 100 in all directions, and the optimizers fail to find an acceptable solution. This also is expected since close aircraft produce ill-conditioned problems for the optimizer.

The problem of selecting the best aircraft that minimize the DOP is non-trivial, especially with large amounts of data where combinations grow exponentially. It is out of the scope of this research, but it would be an interesting future line of research. Besides, geometric DOP minimization does not always work best; other approaches might yield better localization results, as shown in [98].

Even without an algorithm that yields the absolute minimum DOP, it is possible to establish heuristics to have a good enough set of aircraft to perform the localization task. The

best DOP values are obtained when aircraft are separated apart, which can be guaranteed by trying to select messages surrounding the sensor. It might appear as a 'chicken-and-egg' problem because to select aircraft that surround the sensor, its position should be known. A straightforward solution is to take the average position of the observed aircraft within a time window and use it as the starting point. We observed that this approach yields errors in the lower 10s of km, which is a good starting point. It could be possible to iteratively re-select sensors and perform the localization again should more accurately be needed.

These heuristics also reduce the messages needed for localization without significantly losing accuracy. For example, we can draw a circle around the sensor, divide it into equal sectors and select a few aircraft belonging to each sector. This way, the spatial diversity is preserved, but the message amount is reduced, reducing the time to compute a correct position.

For example, in Fig. 4.17, we show the results of applying sectorization (16 and 32 divisions) and their effect on the overall accuracy. We can observe a notable improvement not only in the number of messages but also in the accuracy. However, improving geometrical distribution provides better results for devices with lower PPM values. For older versions of the RTL-SDR, the improvement is not as significant, thus indicating that localization performance is also affected by the radio frontend quality.

4.7.4 Architecture performance

We set up a server and a Raspberry Pi 4 as a User to develop and test the architecture. The server has 16 GB of memory, an 8-core 2.2 GHz Intel Xeon CPU, and 100 GB of disk space. The API and the Localization Engine are implemented in Python. To develop the API, we use Flask-Restful [102], and the localization module for the Raspberry Pi relies on SciPy's optimization modules [103]. For the database, we choose Redis [104], a popular in-memory database in combination with Redisearch [105] that allows creating indices and querying data with more complex search parameters. Data for sensors and aircraft is introduced in the database, occupying 5.71 GB of memory.

To evaluate the performance of the different stages, we collect the messages from each sensor into a JSON object and send them through the API. Each aircraft message is encoded in around 62 bytes. Thus, 1000 messages can be encoded in about 60 kB, which is relatively small by today's commercial internet providers. We evaluate the time per stage with the number of messages ranging from 10 to 10,000, and the main results are summarized in Table 4.5.

The time taken to query the database and obtain the matching messages is relatively small, and even for high amounts of messages, it only takes around 1 second. On the other hand, when computing the TDOA pairs for the user with reference sensors, the time increases notably, taking up to 30 seconds to process 10,000 messages.

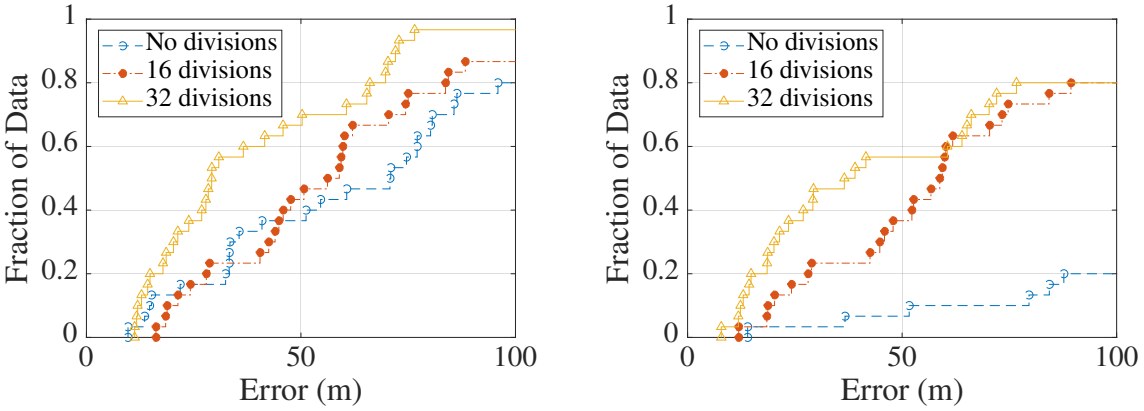


Figure 4.17: Results when sectorizing and picking aircraft per sector. For both Robust (left) and Non-robust (right) initialization.

Joining these results with the ones shown in Fig. 4.11, for 1000 messages which are around 40 seconds of data collection, it would take on average less than 4 seconds to compute the position of the user (not counting network latency) which is an acceptable delay for static or slowly-moving devices. After an initial estimation, with fewer messages, it would be possible to recompute the position, leading to lower estimation times.

Time to query Database (in seconds)							
Amount of Messages	μ	σ	**1%**	**25%**	**50%**	**75%**	**99%**
10	0.00658	0.00045	0.00531	0.00652	0.00659	0.00666	0.00808
50	0.01072	0.00115	0.0058	0.01082	0.01092	0.01103	0.01328
100	0.0153	0.00087	0.01031	0.0152	0.01529	0.01547	0.01774
500	0.05451	0.00338	0.04364	0.05379	0.05413	0.05483	0.06959
1,000	0.10819	0.00858	0.06626	0.10741	0.10835	0.11008	0.15233
5,000	0.54248	0.02363	0.48013	0.53748	0.54053	0.54483	0.61043
10,000	1.12815	0.04312	0.982	1.12037	1.12617	1.14578	1.20786
Time to compute TDOA values (in seconds)							
Amount of Messages	μ	σ	**1%**	**25%**	**50%**	**75%**	**99%**
10	0.03044	0.00492	0.02879	0.02904	0.02926	0.02954	0.05453
50	0.12524	0.00198	0.12383	0.12429	0.1245	0.1251	0.13488
100	0.24099	0.00193	0.23761	0.24031	0.24067	0.24116	0.25089
500	1.2122	0.02664	1.1984	1.2033	1.2062	1.2079	1.3417
1,000	2.5457	0.04442	2.5093	2.5169	2.5245	2.552	2.7252
5,000	13.9	0.12358	13.712	13.801	13.9	13.986	14.261
10,000	31.121	0.33737	30.669	30.941	31.063	31.195	32.603

Table 4.5: Time to execute different components of the proposed system.

4.8 Discussion

Apart from user positioning, SkyPos can also be used in location verification for sensor networks. ADS-B signals are sent in 'clear text,' some papers have attempted to fabricate fake signals to confuse surveillance systems [13], [106]. Nevertheless, these attacks generally require multiple transmitters, with accurate synchronization and calculated delays in the transmission times to be effective. From the scope of this chapter, such an attack would have little impact on the localization performance. If an attacker tried to inject fake ADS-B messages to the user, those would get discarded in the backend, so to make it feasible, it would need to target multiple reference receivers. This would become infeasible with more aircraft messages collected and reference sensors used.

The proposed solution in this chapter has several advantages, including notable accuracy in outdoor or high synchronization scenarios, requires little data to obtain an initial position, and can be implemented in low-cost sensor networks with a cheap SDR receiver like the RTL-SDR and readily available open source software. Work in [40], [63] has attempted indoor localization similarly using aircraft signals to this work. The former relies on sending fully decoded messages to a central backend and performs aircraft tracking simultaneously. In our work, we show that it is feasible to send less data and it is possible to achieve slightly better accuracies even for sensors with very high offsets and drifts. The latter relies on operating on the raw message, notably improving accuracy but increasing computation and storage requirements. Hence, running on low resource IoT devices is not feasible.

Our work aims to minimize the data exchange between centralized entities and allows the user to compute its position instead of offloading it to a single backend.

However, SkyPos is not free of limitations. In the following lines, we mention the following most important ones:

- Low altitude accuracy. As mentioned in previous sections, because of the geometry of the underlying positioning problem, optimization algorithms tend to lose numerical precision on the vertical coordinates. Other sources of altitude might be necessary to achieve higher altitude precision.
- Not suitable for fast-moving users. Due to the necessity of accumulating data, sending it to the backend, and later processing it, it cannot support highly dynamic users, such as drones. However, for slow-moving scenarios, it can be an additional source of position verification instead of relying on users' input.
- Variability due to time of day. Usually, we observed fewer aircraft at night so that it could affect the system's overall accuracy at night time.

Despite these limitations, we have shown that it is possible to build a localization infrastructure with aircraft signals and obtain notable results. Future works will look into better aircraft selection methodologies that reduce the data needed and, thus, the time to compute the user's position.

4.9 Conclusion

In this chapter, we have addressed the problem of localizing sensors using aircraft signals instead of the widely employed GPS. We build on previous research, develop a prototype architecture that can be embedded in existing aircraft monitoring solutions, and test it against hundreds of thousands of data points. The results obtained comparing GPS based with our proposed solution showed that we can get reasonable positioning accuracy in almost real-time using lower-cost hardware. Although it is not a complete substitute for all GNSS based positioning services, we demonstrated that it could be used as a positioning method for embedded sensors in IoT networks in GPS-denied areas for example, or as position verification mechanism for GPS based receivers.

PART III

LOW-COST SDR IN HIGH-LEVEL APPLICATIONS

5 Identifying Driver Patterns with Low-Cost Spectrum Receivers

5.1 Introduction

Tire Pressure Monitoring Systems (TPMS) were originally designed to increase vehicle safety and simplicity of maintenance [107], [108]. In the United States, TPMS is mandatory in all vehicles produced after September 2007. The regulation in Europe is more recent, with TPMS mandatory for all new vehicles released after November 2012. Today, a great portion of the world's cars make use of low-power wireless communications to transmit pressure and temperature readings from tires to the control unit inside the car. Wireless TPMS sensors rely on simple modulation schemes and are typically not encrypted. Thus, a passive attacker in the vicinity, could potentially listen to the content of the transmitted messages of someone's car and track them without consent. The seminal work in [109] showed a decade ago that Ettus USRP boards can be used to listen to TPMS transmissions. However, USRPs are bulky and expensive, increasing the bar for potential attackers. Other works have followed, mostly focusing on simulations, on data analysis and on privacy evaluation [110], [111].

Given this background, this paper poses that threats could now becoming more serious: the advent of IoT devices equipped with low-cost SDR hardware offers an alternative to high-end spectrum analyzers to monitor the spectrum [15]. An overview is showed in Fig. 5.1. A network of IoT devices could extract and decode car's signals and perform analytics, e.g. identify car's pattern, that were not possible before.

Previous work has shown that these low-cost spectrum sensor devices can be of interest to stakeholders in the telecommunications sector as well as for practitioners in the field, due to the insights provided in the spectrum usage [15], modulation and technology used to communicate [70], and signal decoding [16], [112]. In particular, a cheap SDR dongle such as the RTL-SDR [37] can sample In-Phase (I) and Quadrature (Q) data at a speed of about 3 M samples per second, which is sufficient for most IoT protocols. This dongle combined with a Single-Board Computer (SBC) becomes an efficient sensing device that can decode signals such as FM, AM, ADS-B (aircraft), AIS (ships) and LTE cell identifiers using optimized open source decoders. Signals can then be sent to a central-

ized backend for further processing [16]. However, as SBCs can easily reconfigure their software, these new pervasive spectrum sensing platforms can be repurposed by malicious actors. In practice, an overview is depicted in Fig. 5.1, we showcase how can be used to decode and correlate any unprotected TPMS signal, infers peoples' routine, and pose a new threat to the privacy of car drivers.

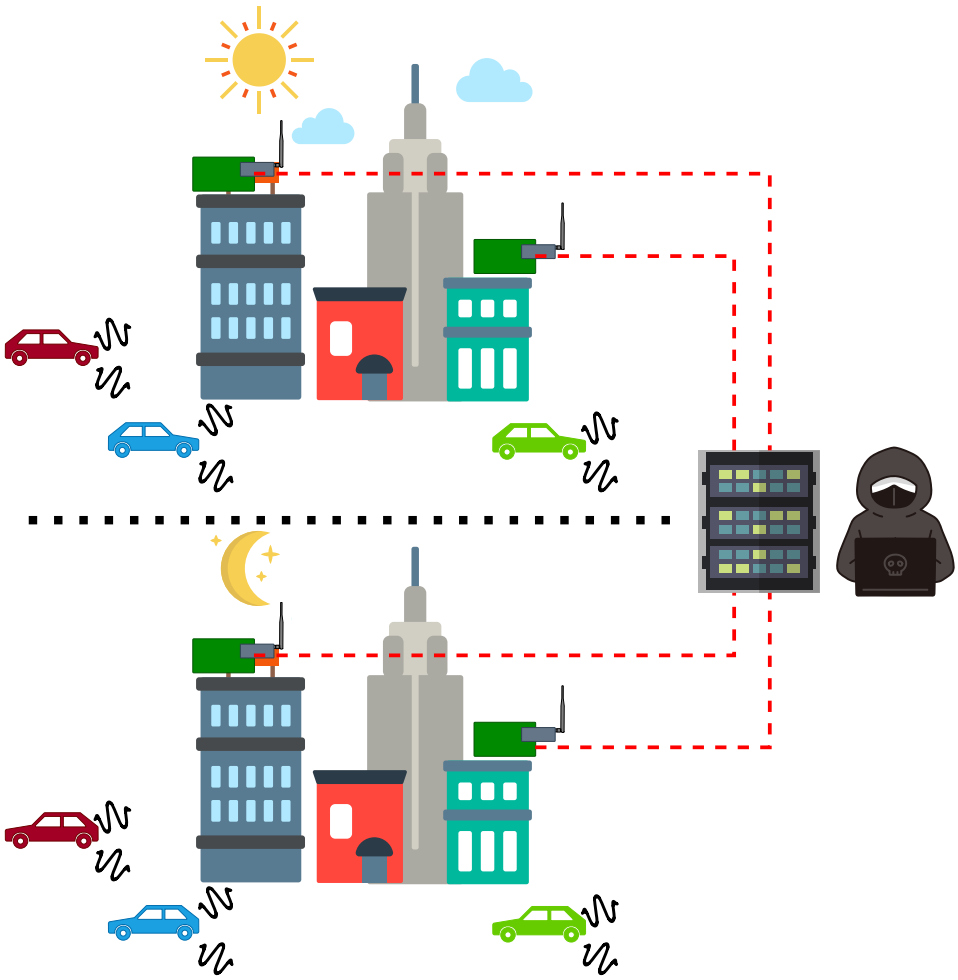


Figure 5.1: This chapter demonstrates how a network of low-cost spectrum receivers can capture and analyze the movement patterns of vehicles, inferring car owners' routines. A malicious actor might install sensors that capture car transmissions continuously, allowing them to track cars (and their drivers) over a region.

In this work, we empirically investigate how the recent development of networks of low-cost spectrum receivers affects the privacy of modern car drivers. We collect more than 75,000 TPMS transmissions from 12 verified users (each of them sole driver of one car) over a period of 5 weeks, and study the feasibility of inferring private information about driver patterns over time and space.

During the data collection and analysis process, we faced and solved the following challenges:

Data collection feasibility: Receivers must be in the wireless range of vehicles' TPMS transmissions, to properly receive and decode the signal. We did not know a prior how close they had to be for practical deployments, given that low-cost spectrum receivers are far less sensitive than more powerful hardware.

Dealing with non-robust decoders: Decoders are developed mainly by reverse engineering the TPMS sensors. Moreover, the lack of error detection mechanisms and the fact that adversary receivers are deployed meters away at low signal strength result in noisy data. Identifiers and other values can be incorrectly decoded, making identification and tracking more difficult.

Lack of protocol information: Apart from the operation modes, there is not much information on communication protocols used by these TPMS sensors, such as modulation or bitstream data representation. Although research studies have proposed more secure protocols [113]–[117], there is no knowledge of the practical impact of these solutions. Most manufacturers use 32-bit addresses to uniquely identify individual air pressure sensors. We rely on that value to track target vehicles.

TPMS insights over time and space: There is little knowledge about what information can be extracted from tire identifiers and subsequently from attackers collecting data over time and space, and what real threats can be inferred.

To this day, the impact of an attacker that collects and analyzes TPMS data from a network of low-cost spectrum receivers has not been empirically analyzed so far. The uptake of pervasive low-cost spectrum receivers causes unexplored threats because massive deployments could now become possible, and movement and presence patterns can now be detected at scale. Furthermore, correlating data from multiple tires in the car can provide further novel insights.

In summary, our key contributions are as follows:

- We deploy a total of 5 low-cost spectrum receivers in an area of $500\,m^2$, showing the feasibility of collecting TPMS messages for performing analysis despite of the sensitivity limitation of low-cost receivers, the deployment of receivers indoors to create a realistic deployment, and just using dipole antennas to limit the size and visibility of receivers.
- We introduce a detailed architecture for data collection, and show that we are able to collect more than $75,000$ transmissions from 12 cars in the span of 10 weeks. To the best of our knowledge, this is the first research piece to perform experiments at these timescales and space locations.
- We investigate how to assign TPMS ID from the various single tires of a car to individual cars with the objective of improving tracking accuracy and robustness, as signals from specific tires could be lost depending on the orientation of the car;
- We study pattern matching and data mining techniques to mine vehicle patterns from sparse sensor data, offering several key insights on user patterns, together with an analysis on the feasibility of a vehicle tracking infrastructure through TPMS sensors.

5.2 Related work

Cars are becoming more and more advanced in terms of technology. However, its increasing complexity has made them the target of cyber-attacks. New regulations are in place: starting from July 2022, 54 countries worldwide (including all EU countries and other OECD nations) approved new regulations on cybersecurity, requiring cars that are traded to have a cybersecurity certificate [118], [119]. This certification implements the United Nation Regulation No. 155 for cybersecurity management system of vehicles [120]. For example, the United Nations' regulation lists various cyberthreats to vehicle data, such as identity and location information. The TPMS is however not considered yet in this certification process. As we show in this work, we urge revising the regulation, considering the threats related to TPMS transmissions.

TPMS has been analyzed in previous works from several perspectives. In [109], authors analyzed TPMS properties regarding security and privacy. They demonstrated how spoofing attacks are possible. In this sense, TPMS packets were successfully decoded up to 40 meters away from the transmitter with a dedicated low noise amplifier. Besides, false measurements and pressure alerts were displayed in a real car just by sending packets with a false and spoofed ID. This study was limited to a single Ettus Research's USRP board, it did not perform ID correlation to identify actual cars or movement patterns but it rather focused on active attacks to manipulate pressure readings.

Hacker et al. [110] performed car identification from a traffic simulation tool and not from real TPMS sensors. They assumed the presence of a reverse lookup in truth vehicle data to score car candidates, unavailable in real environments. Other assumptions do not hold in practice, such as 1-second window associations, losing a great deal of information as many TPMS models can take up to 1 minute to transmit.

Sterne et al. [111] analyzed privacy concerns regarding eavesdropping attacks in different motion states of the transmitter and the receiver (static receiver and moving transmitter, both traveling in the same direction, both traveling in different directions and both static). The authors studied TPMS sensor transmission triggering and radiation patterns in different rotation angles. However, no identification or correlation experiment was performed. Here, most of the analyzed data were also extracted from simulations, using only 3 real cars for the experiments. To enhance privacy, Kilcoyne et al. [114] proposed an encryption method for TPMS sensors' IDs, based on linear-feedback shift register (LFSR)-based cryptography, enhancing system resilience against spoofing attacks. The work in [113] proposed to use the embedded accelerometer in the TPMS to estimate the best moment to send the data, when the tire sensor is closer to the receiving antenna, reducing the transmission power and, therefore, the distance from which the TPMS data can be externally collected.

Other research related to increasing privacy through encryption can be found in [115] and [117]. The former proposes software-based encryption for TPMS messages whereas

the latter presents a novel ID randomization scheme that requires the car and TPMS sensors to be configured with a shared secret key. Recently, there have been proposals for 'Smart' TPMS sensors such as the one from Daimi et al. [116] which aim at replacing current sensors altogether. However, these solutions have not been taken into manufacturer implementation to date, as our findings show that many brands still use TPMS sensors without any encryption. In fact, many TPMS programming devices still rely on this lack of encryption to obtain the tire pressure information [121]. Other works have instead looked at TPMS from different perspectives, including energy harvesting techniques for providing a maintenance-free and battery-less solution [122] or improving traffic monitoring [123].

In summary, although prior work has studied possible privacy and security threats in TPMS, their approaches present significant limitations, mainly related to the lack of real data and analytic studies. This work seeks to solve previous limitations, by deploying a network of low-cost spectrum receivers to collect real TPMS data, understanding the current status of deployment of more secure protocols, and analyzing our data in time and space to extract insights regarding driver security and privacy issues. Our study is timely, given the aforementioned ongoing regulations on cyberthreats at the United Nations level, and in a large number of Countries in the world.

5.3 Primer on TPMS systems

When measuring tire pressure, TPMS systems are divided into 2 categories: Indirect TPMS (iTPMS) and Direct TPMS (dTPMS). The former measures wheel pressure through speed sensors (typically from the ABS system) and tracks the linear and rotational speed variation over time. When the difference in one of the wheels exceeds some predetermined thresholds, the system reports a pressure loss. There exist a variety of methods to determine the pressure [124], but they are regarded as low precision because they do not directly measure the wheel's pressure. These systems are favored by companies in the Volkswagen Group (VW, Audi or Skoda).

On the other hand, dTPMS systems utilize sensors embedded in the rim of the wheel, or on the inner liner of the tire, to continuously to directly measure its pressure and temperature [122]. dTPMS makes use of battery-powered pressure sensors to acquire the tire pressure of a vehicle. As there is no open dTPMS standard, dTPMS communication is based on proprietary protocols and simple modulation schemes (for instance, ASK or FSK at 315 MHz or 433 MHz).

Each TPMS sensor transmits the information about tire pressure and identifier to the Electronic Control Unit (ECU), located inside the vehicle. Manufacturers like Toyota, Renault, Hyundai or Mercedes typically favor these systems over iTPMS.

The length of the message varies depending on the manufacturer but dTPMS messages usually are 100 bits in length with a symbol rate of 20 kbps, meaning that a full transmission takes 5 ms. A TPMS message contains the following fields:

- **Preamble**: With common hex structures like `0x55555556` or `0xaaaaaaa9`.
- **ID**: A hexadecimal 24 to 32 bit string with the dTPMS sensor identifier. For example: `0xb62aaf`.
- **Temperature**: The temperature of the wheel in 8 bits.
- **Pressure**: The pressure of the wheel in 8 bits.
- **Different flags and checksum**: Depending on the manufacturer, dTPMS sensors might add information like the battery or sensor health status. At the end of the message an 8 bit CRC checksum is added.

Although the operation of these devices vary among different manufacturers, most of them transmit pressure information when sudden changes in the tire pressure are detected or when the vehicle is moving. When the vehicle starts to move, motion sensors trigger the pressure sensor to start data transmission with a period of 30-120 seconds. Another way to trigger transmissions from a TPMS sensor is to send a pulse in the LF band (125 kHz), which is the operation principle of many TPMS monitoring tools used in repair shops, such as the one shown in [121]. A diagram showing a basic dTPMS sensor architecture is shown in Fig. 5.2.

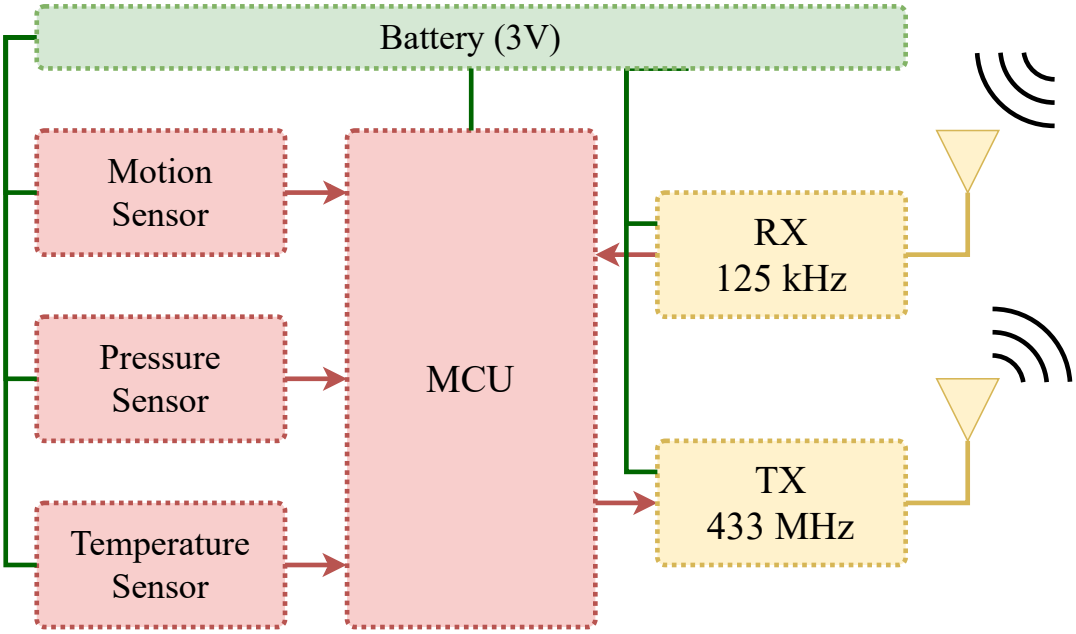


Figure 5.2: dTPMS Sensor components.

As an example, [125] shows the modes of operation of a dTPMS sensor from Schrader Electronics. It defines up to nine modes of operation for these sensors, but the ones we are more interested in are *Drive*, *Stationary* and *ID Response* modes. The first sends up to eight messages in 60 seconds (1 message every 7.5 seconds on average) while the car is moving. The second mode sends messages every hour when the car is stopped. The last one is a response to the trigger in the LF band that can be used to verify the sensor IDs of a car. As we will show in this work, this is not the case for all brands, as some transmit at high rates even when the car is stopped.

High-end car models use the 4-antenna configuration to receive the information from the tires. Each antenna is connected to the ECU and is located very close to the tire (in the arc of the wheel). Therefore, every pressure sensor can transmit at low power due to the vicinity of the antenna. However, the automotive industry always aims to reduce manufacturing costs, and hence prefers a 1-antenna model, which usually is located in the rear window of the vehicle (other cars have the antenna located close to the radio). The 1-antenna model reduces the costs, but pressure sensors need to transmit at high power. The latter implies that signals transmitted can escape from the car context and messages could be properly decoded by any receiver in the vicinity.

Because dTPMS send data wirelessly without encryption or obfuscation (other than by using proprietary, yet simple, communication protocols), we believe that they might pose several threats to driver privacy. Therefore, the focus of this chapter is on these systems, and we will use the terms TPMS and dTPMS interchangeably.

5.4 Threats of a TPMS surveillance system

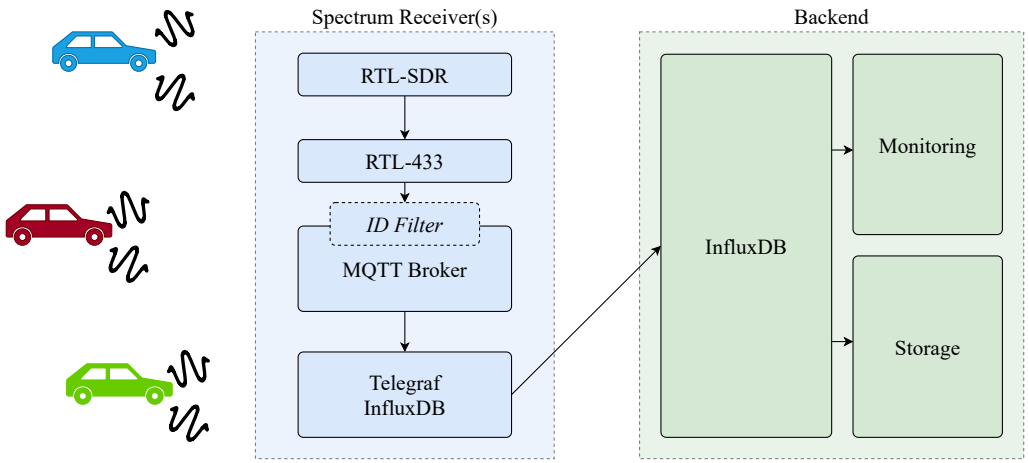


Figure 5.3: The software architecture used in this work is composed of low-cost RTL-SDR platform publicly available TPMS decoder. The data pipeline and monitoring interface are also composed of open-source software.

TPMS transmissions are sent without any encryption or secure mechanisms and include a unique identifier. This allows anyone with an affordable equipment like a low-cost spectrum receiver and a standard off-the-shelf antenna to capture and track them throughout time and space. In this section we explore how this could be exploited and what information could an attacker gain by gathering TPMS measurements.

5.4.1 Threat Model

For our analysis, we consider the following passive tracking system with low-cost spectrum receivers as a threat model. At a high level, an entity could set up an architecture like the one shown in Fig. 5.1, with receivers distributed over a geographical area, continuously capturing tire messages. These devices need not be costly, as RF dongles like the RTL-SDR [37] attached to a Raspberry Pi have enough capability to collect messages continuously. Apart from ID filtering, the rest of the components are open source and easily available online, making this kind of architecture easy to deploy and affordable to anyone. Once the architecture is in place, the threats are manifold.

Malicious users could deploy passive receivers at large scales and track citizens without them knowing. The advantage of such a system, over more traditional camera-based ones, is that no direct Line of Sight (LOS) is needed with the TPMS sensors and spectrum receivers could be placed in covert or hidden locations, making them harder to spot by victims. A data mining company could deploy receivers, gain insights on the types of traffic and routes taken, and then sell that data, all without the awareness of the users (the drivers).

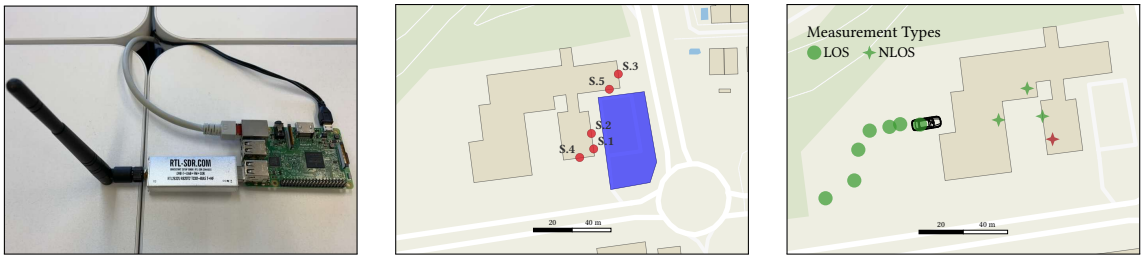
By establishing such a network of spectrum devices over a city, malicious users could track cars and infer behavioral patterns. Indeed, another type of attack stemming from passive surveillance could be for burglars in suburban residential areas. By tracking the vehicles of each household, they could infer the schedule and pattern of a particular household and take advantage of their absence. Finally, by combining passive monitoring with active spoofing, threat actors could track logistics trucks, spoof flat tire alerts to force stops, and then hijack the freight.

This work does not explore active attacks, but the scenario presented in Fig. 5.1 could be extended to also include spectrum transmitters targeting specific TPMS sensors obtained through passive monitoring, for instance with the objective of depleting the battery of TPMS sensors. However, this chapter uncovers the significant warnings associated with such wireless transmissions, including the potential threats in tracking vehicle patterns.

5.4.2 Tracking Car Owners

Capturing TPMS messages is relatively easy as they do not employ any security mechanism. By collecting and analyzing this data, potential attackers can yield insights that aggregate vehicle information to track users' movements. This work demonstrates a concrete use case, illustrated in in Fig. 5.1 where, through the application of our proposed algorithm detailed in Section 5.7, these insights enable the matching of tire IDs and the aggregation of tires to identify specific vehicles, inferring about workers' routines.

By matching the 4 TPMS sensors that comprise a car we can obtain a more robust pattern of the driver's profile, when they start and end their workday, but also can observe



(a) Spectrum receiver setup. (b) Receiver deployment. (c) TPMS range tests.

Figure 5.4: Selected receiver and deployment site. We prepare 5 spectrum receivers around the selected area (Fig. 5.4b). Receivers (red) are placed indoors, and the violet area represents the parking lot were most measurements are focused on. In Fig. 5.4c, we show the location of receiver positions used to test the range of TPMS messages. Marked circles indicate the position of spectrum receivers with LOS conditions. Marked stars indicate scenarios with NLOS conditions.

anomalies in their behavior. This could help threat actors to infer the location of the user at any time of the day, but also alert them in case an extraneous behavior was detected. This and other examples are further discussed in Section 5.8.

5.5 Deployment

5.5.1 Data collection setup

We position a total of 5 spectrum receivers along the perimeter of a workplace. The total area covered is the blue region of Fig. 5.4b and surroundings, which are about 100×50 m^2. The setup consists of an RTL-SDR connected to a Raspberry Pi, which is cost effective while having a small form factor (see Fig. 5.4a). The equipment necessary to collect these data can be as low as 100 dollars per spectrum receiver. Internet connectivity depends on the deployment location, with some places offering Ethernet or Wi-Fi, while mobile internet connections can be used in more remote locations. Moreover, if internet connectivity is unavailable, data can be recorded locally on the device and collected periodically.

In order to create a realistic deployment, we place the spectrum receivers indoors pointing at the parking and the nearby roads, and close to windows. The deployment is presented in Fig. 5.4b. The distances range from $10 - 50$ m, and we show later that it is sufficient to capture many transmissions.

We use the open source `rtl433` software for message decoding [126]. It includes around 250 different decoders operating at 433MHz, 28 of them related to the TPMS sensors of different manufacturers. `rtl433` provides various output formats such as JSON, MQTT, or InfluxDB. We opt for a combination of the MQTT and InfluxDB, so that we can reliably send data to our backend and for its flexibility in terms of managing discontinuous

data, tools to deal with aggregate data and large data sets, and optimization for time series data, which makes it ideal in our scenario. In order to support data collection, we set up a centralized InfluxDB collecting data from all receivers, where we create a database per each. We show a general overview of the architecture in Fig. 5.3.

In every database, there is one table per TPMS decoder or manufacturer. We show the main manufacturers we can observe, as listed in Table 5.1a. Note that Schrader and Jansite are Original Equipment Manufacturer (OEM) suppliers used in many car brands. The main fields of interest along with data type are summarized in Table 5.1b. Time denotes when the data have been sensed, while ID is the identifier of the TMPS sensor. In order to visualize the data and to make preliminary analyses such as data quality checks, we create several Grafana dashboards, directly connected to the InfluxDB databases.

5.5.2 Feasibility

TPMS sensors are expected to transmit at low power, so most research assumes that these transmissions should be visible up to 10-20 m [127]. However, our observations contradict this, and transmissions can be observed at distances greater of up to 55 m.

To test and validate that our receiver deployment is capable of obtaining enough tire pressure information, we measure Car 3 (C3) transmissions at different distances and conditions. The scenario is illustrated in Fig. 5.4c. We trigger the transmission of all the tires of the car and measure the number of messages and the SNR at the 7 positions of the map (green circles), with spectrum receivers deployed outdoors, with LOS conditions with respect to the car's tires. The furthest position is around 55 m. In this scenario, in all positions, we can receive all messages with a SNR of at least 10.0 dB.

For the second part of the test, we set our measurements in NLOS conditions (we place the spectrum receiver inside a building, without any window nearby) and we are able to capture and decode TPMS measurements in 3 out of the 4 selected locations, with a SNR

Table 5.1: TPMS manufacturers (a) and fields (b).

(a)

Manufacturers	**Abbr.**
Ford	F
Jansite	J
Schrader	S
Toyota	T
Schrader-EG53MA4	S2
Hyundai-VDO	H
Renault	R

(b)

Fields of interest
Time (primary index)
ID (String)
SNR (Float)
RSSI (Float)
Center Frequency (Float)
Pressure (kPa/PSI) (Float)
Temperature (C/F) (Float)

of at least 8.0 dB. These results show that it is feasible to install receivers at relatively large distances from roads and still obtain many car tire measurements, in contradiction with the assumption that they are visible only up to 10 m. Therefore, we conclude that our deployment is suitable for measuring TPMS transmissions in the area shown in Fig. 5.4b as the distances involved are well within limits. Clearly, with a better antenna specifically designed for 433 MHz, it could be possible to reach even greater distances under both LOS or NLOS conditions.

5.6 Dataset

We collect the data for our experiments from the cars of 12 individuals. To verify the IDs of each TPMS sensor we use the Autel MaxiTPMS TS501 [121]. This device is able to transmit the required 125 Hz LF trigger and capture the response of the TPMS sensor back. At the same time, we also capture the response of each of the TPMS sensor with a laptop to verify that the IDs decoded by the software library `rtl433` coincide. Once we collect all TPMS IDs, we add them to a whitelist in each of our receivers so that only those are sent to the backend.

To protect the privacy of users, we will not show the real IDs of the cars used in this study. However, we will make all the software and hardware required to perform the data collection public, together with instructions, to ensure reproducibility of our results.

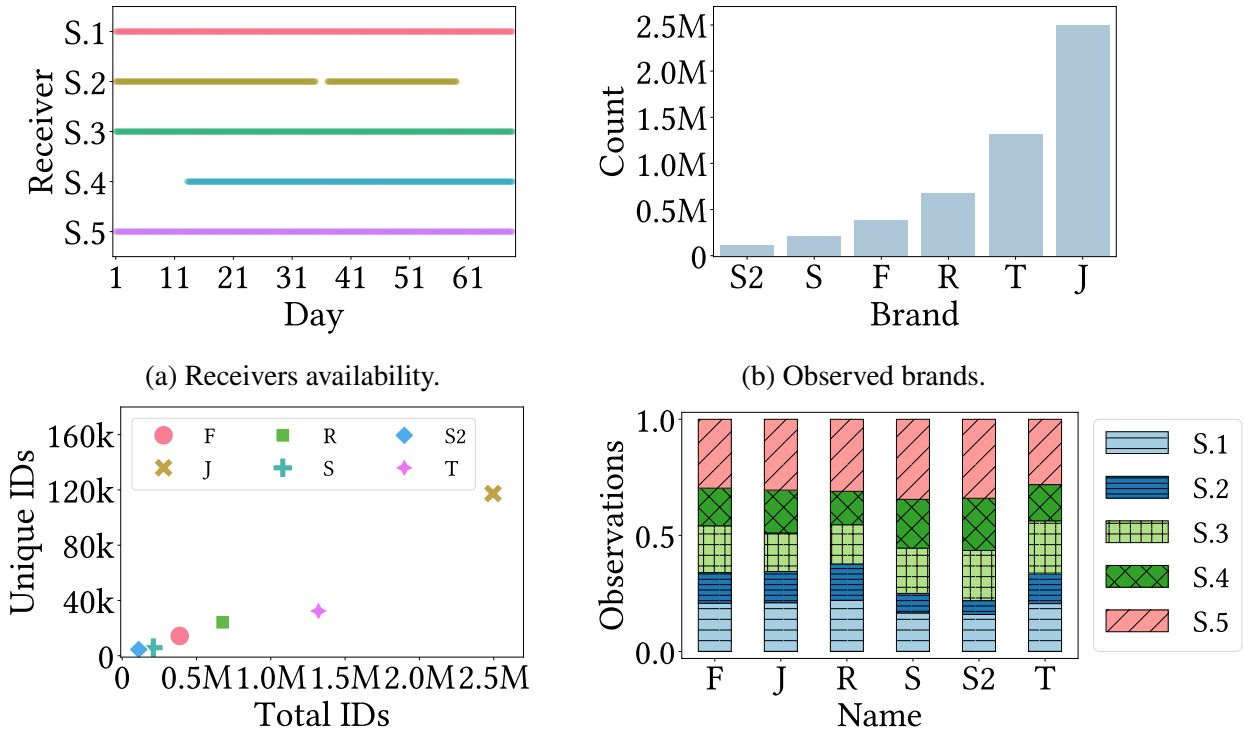
5.6.1 Receiver Sites

The availability of the spectrum receivers differs throughout the measurement campaign. Fig. 5.5a shows the data availability over the measurement campaign among receivers. Among the factors that can cause downtimes and lower data volumes are interruptions in the power supply, and the location of the antenna. This makes the deployment realistic, as these issues may occur also for an attacker.

We conduct our measurements over a time span of up to 9 weeks. Overall, the first measurements start with 4 receivers, which are used to build the data collection framework.

Table 5.2: Approximated receiver sites, their availability and generated data.

Site name	Recording Hours	Measured IDs	Recorded IDs
Receiver 1	655	476248	17621
Receiver 2	655	412905	22706
Receiver 3	655	514484	3230
Receiver 4	653	753327	7708
Receiver 5	362	356526	24123



(a) Receivers availability.

(b) Observed brands.

(c) Total vs Unique IDs.

(d) TPMS manufacturer distribution.

Figure 5.5: TPMS measurement characteristics for the selected location. Fig. 5.5a shows the availability of each receivers over time. Fig. 5.5b the number of measurements per TPMS sensor type, Fig. 5.5c the comparison between the total and unique IDs in the dataset and Fig. 5.5d how each car brand is distributed across receivers.

The network is then extended with 1 additional receiver starting after 12 days. A summary of the receivers sites, their availability, and generated data is shown in Table 5.2. The *Measured IDs* represent all the unique TPMS IDs each receiver has processed during the data collection campaign. However, as we are only interested in IDs of specific users, we use the term *Recorded IDs* to represent the subset of data points from those authorized users. With this we aim to provide insights into how many measurements it would be possible to collect had we gathered all data from the wild.

5.6.2 Recorded data

The data set consists of more than 110,000 data points that were recorded over more than 7,500 combined sensing hours. A data point is collected every time a receiver captures and decodes TPMS signals. The distribution of observations for each TPMS type is shown in Fig. 5.5b.

The distribution shows *Jansite* with a substantially high number of received signals, followed by *Toyota* and *Renault*, each with significantly more than other types of tire pressure monitoring. *Jansite* manufacturer are equipped in many brands, which is one of the

reasons why they are so prevalent in the data. Another is that, at of the moment of data collection, *Jansite* decoder from the `rtl433` [126] does not implement a checksum verification, thus it sometimes confuses other protocols. We note this happening with 2 cars of our dataset, so we substitute those IDs for the actual ones.

The reasons for such prevalence could be that cars equipped with these types of TPMS sensors are more common around deployed locations, or that they simply have higher transmission rates. To distinguish between these two cases, we also look at the transmissions of individual TPMS messages identified by their addresses. Each TPMS does pressure measurements for a single car tire. We note that these addresses might not be unique. Measurements also contain a CRC so that the decoder is able to discard erroneous messages. However, Fig. 5.5c shows a high correlation between the number of unique IDs and the number of observations for each TPMS sensor type, which indicates that the different systems use similar transmission rates.

Fig. 5.5d shows the percentages of observations by receiver site and TPMS sensor types. Due to the proximity of receivers to one another, distributions look very similar; the main difference among them is the amount of messages received. This can be explained because some receivers mostly point at roads or more sparse areas, whereas others point to car park areas and thus are able to capture more messages from stationary vehicles. Instead of using a single receiver, by utilizing all 5 we increase the likelihood of capturing transmissions from vehicles of interest.

5.6.3 Ethical Considerations

Each of the identifiers, tied to the brand of the vehicle and the location of the authors' workplace, could lead to the connection of the daily patterns shown in this investigation with specific individuals. Our research is committed to complying with ethical standards and all authors involved in the measurement and collection of data have obtained the necessary Institutional Review Board (IRB) approval.

In summary, the analysis provided in the following sections on driver patterns was done with tire measurements collected from individuals who gave explicit consent via an electronic form. In this paper, we will not show the real car brands or identifiers of the people involved in this study. Instead, we will refer to cars as Ci where i belongs to the 12 cars analyzed in this study. For the Front-Left, Front-Right, Rear-Right and Rear-Left tires, we will refer to them as FL, FR, RR and RL, respectively. Thus, when analyzing the Rear-Right Tire of our car number 7, we will refer to it as C7.RR.

We will make all the software images, architecture and data processing scripts available to the public, for other researchers to replicate and verify our findings.

5.7 Car inference from tires

This section details the data analysis approach for TPMS sensor correlation in order to infer cars and their patterns. The objective is to be able to reliably identify a car based on the transmissions of its four TPMS IDs (more than four when it is a truck or a bus). The idea behind this study is to correlate TPMS sensor IDs into more complex vehicle identifiers based on how they appear together in time. Moreover, by working with ”cars” instead of ”sensor IDs”, our analysis is expected to have a better resolution as transmission rates are low for power constraints. This approach also alleviates the problem that individual ID can be missed, depending on obstructions, like the car itself, between the tire sensor and the spectrum receiver. On the other hand, the probability that no message from four sensors is correctly received by a spectrum receiver is significantly lower, because of spatial channel diversity. In the next section, we will quantify experimentally the impact of performing car matching.

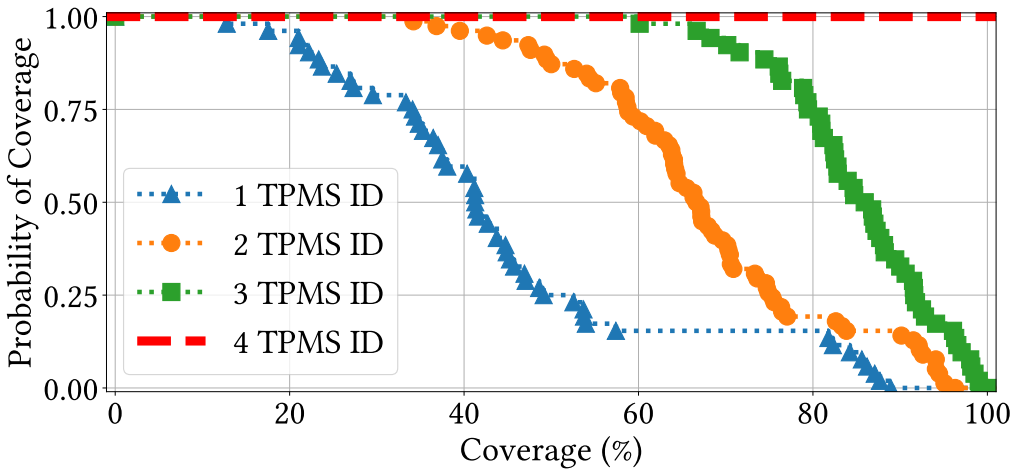


Figure 5.6: Probability of achieving the same time coverage for a car when using 1,2 or 3 TPMS sensors instead of all 4.

5.7.1 Why Car Matching

TPMS transmissions are sparse and the likelihood of capturing all 4 sensors from a car passing by a road decreases with the vehicle's speed. In Fig. 5.6 we show the amount of TPMS transmissions that would be lost if we only used one, two or three IDs for tracking instead of all four. To analyze this, we aggregate all measurements of our dataset in one-minute windows –$\tilde{3}$0-60s is the standard transmission frequency– and count the number of times each TPMS ID appears compared to the number of times the vehicle to which it belongs appears. We call this coverage, and is represented on the x-axis of Fig. 5.6.

A coverage of 100% would mean that a specific ID or group of IDS is always captured whenever a transmission from the vehicle they belong is captured. The y-axis represents

the probability of achieving the given coverage. When we use a single TPMS ID to track a vehicle, the median probability is slightly above 40%, meaning that for half the vehicles in our dataset, we are losing at least 60% of the total messages. When using 2 or 3 IDs the coverage increases, but even when using 3 TPMS IDs from a vehicle, we would still be losing at least 20% of the transmissions for 25% of the vehicles.

These figures are relatively low, and we also must note that most of these measurements were with parked cars, so the percentage should be higher than in moving conditions, where this proportions might reduce significantly. Our findings motivates that by performing car matching, collection and tracking improve significantly and gains might potentially be even higher in higher-mobility scenarios.

5.7.2 Aggregation based on cross-correlation

In this section, we describe a method based on correlation to aggregate sensor IDs belonging to the same car. Our intuition is that sensors belonging to the same car will have similar temporal series over the course of our sampling period.

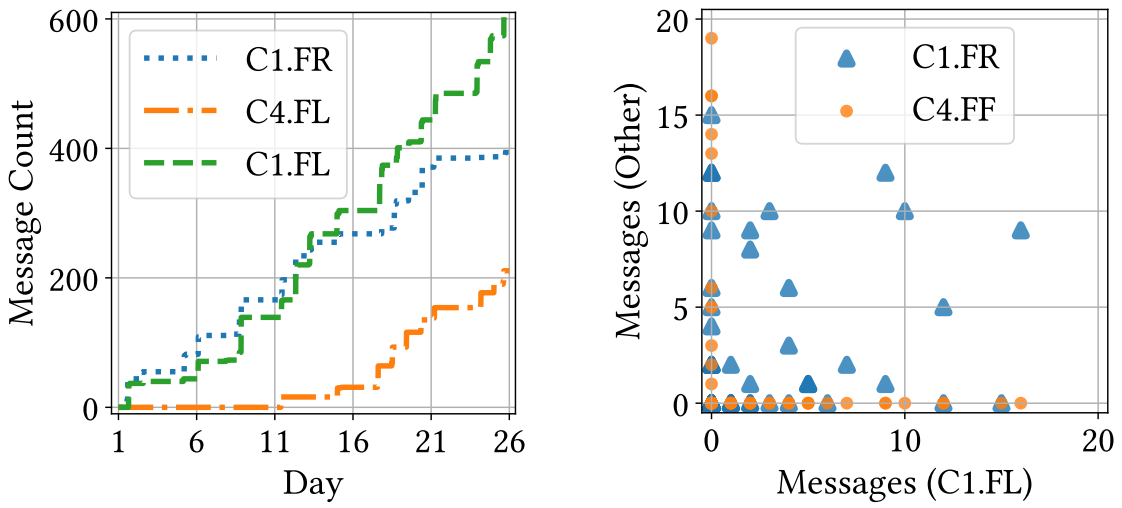
In the first step, we rank tire sensor IDs by their number of appearances and then analyze them in pairs. As an example, we take the Front-Left and Front-Right Tire sensors from car 1 `C1.FL` and `C1.FR`. We hypothesize that they belong to the same car as, looking at their time series we can observe that, generally, every time we receive measurements from one, we receive it from the other one as well. On the other hand, we also take sensor `C2.FL`'s time series. Its time distribution is independent of the other two. In Fig. 5.7a, we show those three sensors and their temporal series. The cumulative sum of the messages matches almost exactly with the IDs that are close, and on the other IDs time series diverges significantly.

If we plot the measurements of `C1.FR` and `C4.FL` against wheel sensor `C1.FL`, as shown in Fig. 5.7b, the correlation pattern is clearer. For a time window of one minute, the number of messages received from sensors `C1.FL` and `C1.FR` appears to be linearly correlated. However, between `C4.FL` and `C1.FL`, they are completely orthogonal (each time we receive a message from one, we do not receive anything from the other). We repeat this analysis for all pairs of sensors. From this study, we expect to extract a more accurate understanding of the number of cars that have passed close to the selected tire sensor.

To model the relationships of these temporal series, we resort to the well-known cross-correlation function [128], which is used to calculate the delay between 2 signals. Assuming 2 signals $x[n]$ and $y[n]$ with N points each, the cross-correlation can be calculated as:

$$\tau[l] = \sum_{n=0}^{N-1} \bar{x}[n] \cdot y[n+l],\ l = -N, \ldots, N \tag{5.1}$$

where $\tau[l]$ is the cross-correlation result, which provides the best time alignment between



(a) 3 IDs over time

(b) Correlation of 3 IDs

Figure 5.7: Correlation of messages between TPMS IDs `C1.FL`, `C1.FR` and `C4.FL`.

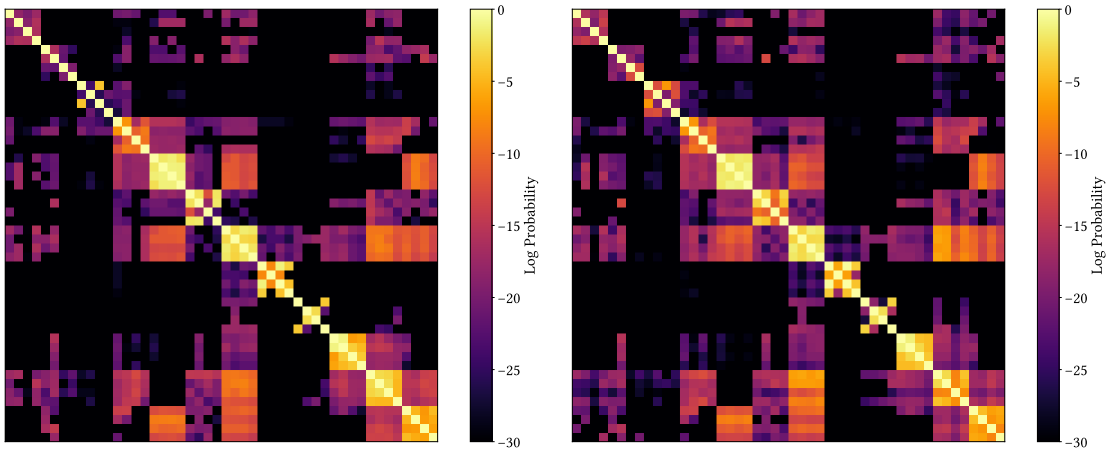
both signals $x[n]$ and $y[n]$.

However, this approach would produce extremely low correlation values if we were to use the entire message distribution for each sensor. This is because arrival times vary significantly even for sensors on the same car, making it impossible to group cars. To overcome this, we aggregate the measurements for each sensor into time buckets of different granularity and observe the results. However, it can occur that the larger the buckets, the more sensors will be highly correlated among each other. Therefore, we expect a trade-off for the time aggregation, and we look for the best operating point that returns results close to the actual number of sensors that could be aggregated.

Another improvement is that instead of cross-correlating the whole set of delays, we can compute the cross-correlation for a smaller set of lags (i.e., for $l = -1, 0, 1$). With this we convert the cross-correlation operation from $\mathcal{O}(n \log n)$, to a faster version with $\mathcal{O}(n)$. In fact, the computation of all sensor-pair cross-correlations takes < 1 second in the researchers' machines.

In Fig. 5.8 we show the correlation matrix for a time aggregation of 1 minute. We observe that some 4 wheel clusters start to appear. This is expected as most vehicles are probably cars that have 4 wheels. Reasons for seeing numbers less than 4 could be that the signal from the other wheels is blocked, and therefore in some cases we only manage to aggregate two or three tire sensors. Using all sensors increases the overall correlation value among sensors belonging to the same car.

We aggregate measurements from 5 seconds to 5 minutes and compute the car aggregations, using the algorithm we describe in Section 5.7.4. The results are shown in Fig. 5.9. Starting from 30 second aggregation windows, when we use 5 sensors we can correctly group all 12 cars of our dataset. When only using one sensor we need at least an aggregation window of 2 minutes to correctly capture all cars. These results match the



(a) Using one receiver.

(b) Using all receivers.

Figure 5.8: Correlation matrix for a time aggregation of 1 minute, using only 1 sensor vs using all sensors combined. We observe scenarios up to 4 tire sensors with high correlation.

intuition that we can obtain when looking at the correlation heatmaps shown in Fig. 5.8.

Although the cross-correlation method yields good results for this dataset, it may have some limitations as windows are fixed and cannot look into the previous or next buckets. In the next paragraphs we present a method to overcome this limitation.

5.7.3 Car aggregation based on Jaccard index

We look at the Jaccard index as a similarity metric between two sets of correlated sensors (*A,B*). This idea is inspired by the preliminary work presented in [110], with the objective of overcoming some of its limitations as it uses simulated and controlled data. Jaccard index is defined as follows:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{5.2}$$

In our case, $|A|, |B|$ are the number of times sensors A, B appear respectively, and $|A \cap B|$ is the number of times both sensors are seen together in the same time window. From our dataset, we extract the tire sensor ID and the timestamp of each packet retrieved. Then, we measure the number of occurrences of each one of the identifiers with the rest of the dataset, grouping together the most frequent sensors. In particular, for each one of the transmissions registered from the sensor being processed, we generate a list of the sensors that appear in the same one minute window ($\pm$ 30s). The amount of time required to compute Jaccard indices for all IDs is around 20 seconds with data coming from all sensors.

Table 5.3 shows the correlation of TPMS sensor with the ID belonging to the Front-

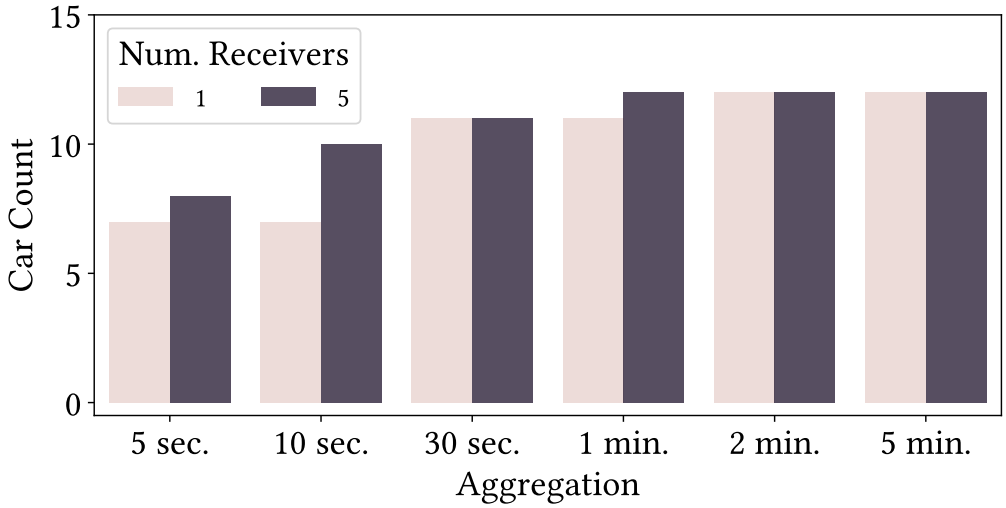


Figure 5.9: Detected cars when using 1 sensor vs 5.

Left tire of Car 4, `C4.FL`, with the rest of the sensors in the dataset. This sensor appears 1022 times in our dataset (aggregated by 1 minute time windows). `C1.FL` appears together with `C4.RR` a total of 476 times, but we can observe how the first three sensors appear more than 400 times in the same minute window, while the rest appear less than 50 times. This gives the first 3 sensors a Jaccard close to 0.30, while the rest are below 0.05. In addition, these three sensors use a very similar ID format (the first 12 bits are identical). Compared to ground truth, we can indeed verify that the other sensor IDs with the highest Jaccard index are, in fact, the other wheels of the car. If we apply this approach to all sensors from a car, we can obtain similar result as in Table 5.4 for Car 9.

We choose a one-minute window as this provides a clear pattern. In particular, Fig. 5.10 shows the distribution of the overlaps over all pairs considering the aforementioned one-minute window. With the Jaccard index method, we can see a clear improvement when identifying IDs belonging to the same car. Twelve clusters are visible (corresponding to the 12 verified cars), and many of them can be separated with relative ease. To obtain all the cars, we can take an iterative approach, where we select the most probable clusters and remove them from subsequent iterations. We discuss this approach more in depth in the

Table 5.3: Jaccard distance with the sensors related to the ID of C4.FL.

Target	**Other**	$\|A\|$	$\|B\|$	$\|A \cap B\|$	**Jaccard**(A, B)
`C4.FL`	`C4.RR`	1022	1017	474	0.3029
`C4.FL`	`C4.FR`	1022	1018	462	0.2928
`C4.FL`	`C1.FR`	1022	995	429	0.2701
`C4.FL`	`C12.FR`	1022	363	76	0.0581
`C4.FL`	`C12.RR`	1022	264	42	0.0338
...	...	...	...	...	...

Table 5.4: Jaccard index between the TPMS sensors forming **Car 9**.

Front-Left Tire Jaccard indexes

Target	**Other**	$\|A\|$	$\|B\|$	$\|A \cap B\|$	**Jaccard**(A, B)
C9.FL	C9.FR	155	160	141	0.8103
C9.FL	C9.RL	155	163	142	0.8068
C9.FL	C9.RR	155	150	135	0.7941

Front-Right Tire Jaccard indexes

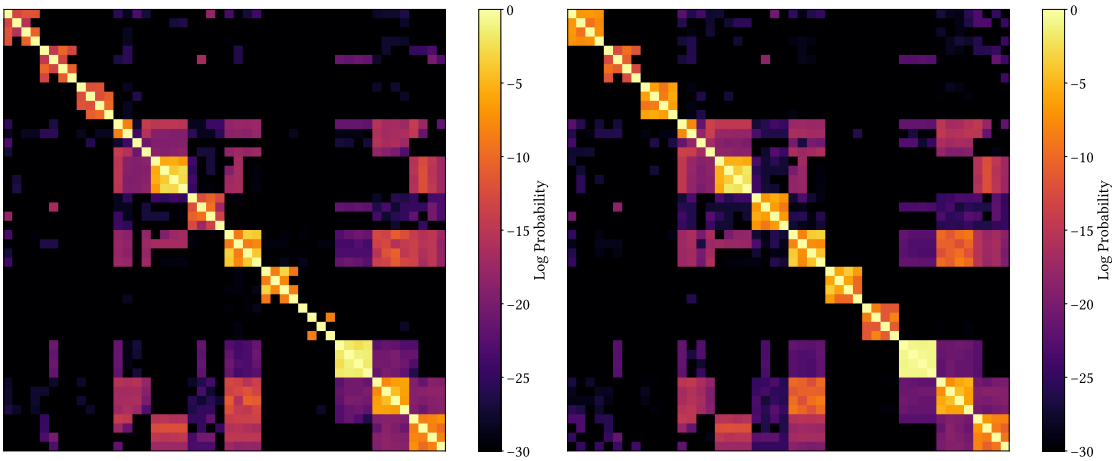
Target	**Other**	$\|A\|$	$\|B\|$	$\|A \cap B\|$	**Jaccard**(A, B)
C9.FR	C9.RR	160	150	143	0.8563
C9.FR	C9.RL	160	163	145	0.8146
C9.FR	C9.FL	160	155	141	0.8103

Rear-Left Tire Jaccard indexes

Target	**Other**	$\|A\|$	$\|B\|$	$\|A \cap B\|$	**Jaccard**(A, B)
C9.RL	C9.RR	163	150	141	0.8198
C9.RL	C9.FR	163	160	145	0.8146
C9.RL	C9.FL	163	155	142	0.8068

Rear-Right Tire Jaccard indexes

Target	**Other**	$\|A\|$	$\|B\|$	$\|A \cap B\|$	**Jaccard**(A, B)
C9.RR	C9.FR	160	150	143	0.8563
C9.RR	C9.RL	150	163	141	0.8198
C9.RR	C9.FL	150	155	135	0.7941



(a) One receiver. (b) All receivers.

Figure 5.10: Jaccard matrix for all verified IDs. Aggregation of 30 seconds.

next Section.

Apart from the fact that IDs belonging to the same car transmit at similar time windows, we also note that in most cases there are several bits in common. For our dataset, the number of common bytes ranges from 2-5, but we also observe that after the common bytes, the next octet is usually formed by consecutive numbers. This insight, combined with the probabilistic observations provided by the Jaccard index, can provide a clear identification of individual cars.

5.7.4 Robust Car Matching

If we only track one TPMS sensor, we might be losing many data points as that sensor might be blocked or not transmitting. For robust tracking, attackers might prefer to aggregate a number of IDs together (4 for cars, more for trucks or buses). Based on the findings of Table 5.4, a naive approach would be for each ID, to select the IDs with the highest Jaccard index over a predefined threshold. However, this approach has several problems: it could happen that for a certain ID, the TPMS sensor from another car is more correlated than to other from the same car (see Fig. 5.10, the 4th cluster). Another problem is that this approach does not take into account transitivity: if sensor A is correlated with B and not C, they will not be grouped together, even though B might be correlated with C.

To overcome these limitations we propose a modified version of this algorithm (Algorithm 1). After defining the aggregation methods, we proceed to describe the car matching algorithm (Algorithm 1). Starting with the measurements matrix C with M time points and N IDs, we can calculate the scoring matrix S, which contains the match score between 2 IDs. This score can be calculated with either cross-correlation or the Jaccard index method.

From there, we start the matching algorithm. We start with a high threshold value θ_0,

Algorithm 1: Car Matching

Data: C; S; θ_0; θ_l; θ_k
Result: D

$D \leftarrow \emptyset$; $\theta \leftarrow \theta_0$
while $\theta > \theta_l$ **and** $S \neq \emptyset$ **do**
  $D^{aux} \leftarrow \emptyset$ **for** R **in** $Rows(S)$ **do**
    $R_j \leftarrow IDs$ **where** $R > \theta$ **if** $Length(R_j) > 1$ **and** $Length(R_j) \leq 3$ **then**
      $k \leftarrow$ FindAny(D^{aux}, R_j)
      if k **is** $\emptyset$ **then**
        Append(D^{aux}, R_j)
      else
        UpdateDonRowk(D^{aux}_k, R_j)
        if $Length(D^{aux}_k) = 4$ **then**
          UpdateDandS(D^{aux}_k, D, S)
          RemoveRowk(D^{aux}, k)
        end
      end
    else if $Length(R_j) \geq 4$ **then**
      UpdateDandS(R_j, D, S)
    end
  end
  $\theta \leftarrow \theta - \theta_k$
end
Function UpdateDandS(R_j, D, S)**:**
  Sort(R_j)
  Append$(D, R_j[1 \rightarrow 4])$
  Remove$(S, R_j[1 \rightarrow 4])$

so that we can match the most prominent cars (In Fig. 5.10 would be the brightest clusters), and add them to the resulting matrix D. If we find a smaller cluster (2-3 IDs) we store them in an auxiliary matrix D^{aux}. In further iterations, if some of the IDs are present in D_{aux}, we append the new ones to the already existing row in the auxiliary matrix. This way we can support transitivity (e.g., if sensor A is strongly matched with B, and sensor B is strongly matched with C, then A,B,C should be matched). When one of the rows of D_{aux} is filled (4 sensors in our case) we add them to D and remove them from D_{aux}. Every time a new car is matched, we also remove those IDs from the scoring matrix. That way we can remove sources of ambiguity for other IDs. After iterating over all rows, we reduce the threshold and repeat the process again until we reach a limit of iterations or all possible cars have been matched.

When comparing both methods, we can observe some clear differences. The cross-correlation method is generally faster: even for low aggregation values, it can compute

the correlations in less than one second for the size of our dataset. In contrast, it is less accurate when identifying cars, compared to the Jaccard index method. With the latter, we are able to identify the 12 verified cars at the aggregation windows we tested at the expense of more computation time. The results of this comparison are summarized in Table 5.5. Note that the aim of this work is not to provide an optimized version for ID matching, but rather show the different techniques that could be exploited to detect and match TPMS IDs to single cars.

From this analysis, we conclude that with relatively simple hardware, data analysis, and correlation methods, it is possible to identify vehicles. The cross-correlation method shows notable results and can produce results at a fraction of the time of the Jaccard index method. However, looking at the heatmaps (Fig. 5.8 and Fig. 5.10) we can observe that the correlation values are 'cleaner' with the second case. In larger datasets it could be possible that the cross-correlation method fails to capture more ambigous results. In general, if looking for real-time speed one could opt for the correlation method, whereas those analysis aiming at more accuracy could opt for the Jaccard index method, at the expense of more computation time.

5.8 Privacy Analysis Results

In the next step of this analysis, we look at the privacy implications of the unencrypted usage of mostly unique TPMS IDs, aggregating them over different dimensions, and the availability of open-source software-defined radio implementations to monitor these signals.

The privacy leakage through the collection of TPMS messages can be achieved with

Table 5.5: Comparison of Cross-Correlation and Jaccard index methods for different time aggregations and naive vs robust matching.

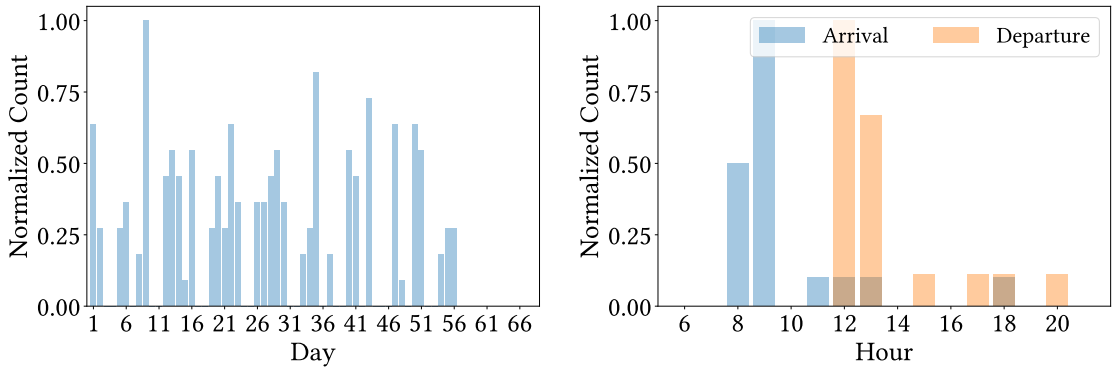
Method	Agg. (s)	Naive Match	Robust Match	Comp. Time (s)
Xcorr	5 sec.	5	9	2.0970
	10 sec.	6	10	1.5616
	30 sec.	8	11	1.1017
	1 min.	8	12	0.8708
	2 min.	10	12	0.6534
Jaccard	5 sec.	7	10	34.1570
	10 sec.	9	12	23.7702
	30 sec.	10	12	21.1729
	1 min.	11	12	19.1987
	2 min.	11	12	18.2871

different approaches. In this section we first explore our dataset and then perform a study to detect static patterns. Then, with correlation techniques, the possibility of grouping RSSI and IDs information in order to track cars.

5.8.1 Temporal Trends

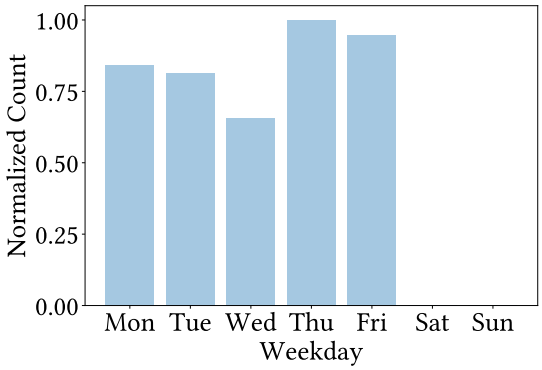
Since cars and, therefore, TPMS sensors are inherently associated with the mobility of their owners, the temporal dimension could reveal the mobility patterns of individuals. That information could be used to infer privacy-sensitive information such as people's work schedules, vacations, or sick leave.

Almost all cars in the data set are seen from Monday to Friday (like for Car 1 on Fig. 5.11c). As expected, an almost 0 trend is also visible on weekends. These observations match the reality of the deployment since the receivers are deployed in a workplace. However, a similar deployment in a residential area would probably reveal a different reality, such as car IDs appearing every day of the week, thus allowing anyone to track when drivers leave their homes.



(a) Messages per week of year.

(b) Selected ID Arrival-Departure.



(c) Daily Message count.

Figure 5.11: Temporal pattern analysis over time of year 5.11a, hour of day 5.11b and weekdays 5.11c.

5.8.2 Analysis of individual cars

Now, we look closer at the temporal dimension of individual cars that are observed in the data set. In addition, this section analyzes the appearance patterns of the most common cars in the dataset, demonstrating how a malicious actor could infer private information about owner movements and behavior. We stress that since we do not have the ground truth (real user identity), the behaviors and patterns described here are hypothetical. The following use cases properly address the challenge mentioned in Sec. 5.1, revealing the temporal insights obtainable from tire IDs, and the potential for attackers to harvest this data across time.

Patterns over weeks

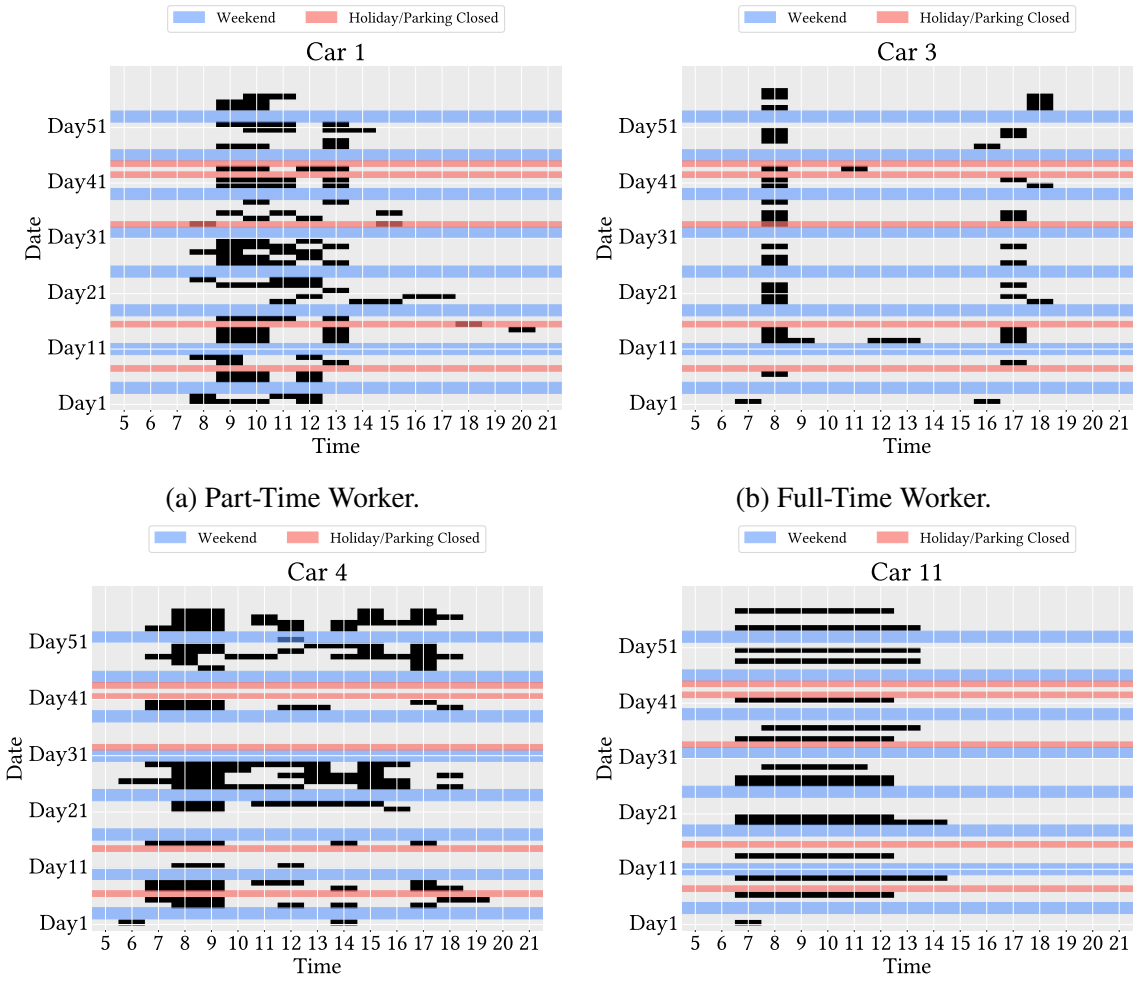
Once all the data is collected and we identify all the sensors that belong to a car, we can start analyzing more precise patterns per driver. In this first study, we look at the car appearance pattern per week, for our verified users.

From the set of Car 1's IDs, we then aggregate information in the time domain from one single car. We first quantify the information spread throughout the observation period. For this reason, we aggregate weekly messages and visualize them as shown in Fig. 5.11. We observe the movements for about 5 weeks. There are some clear patterns in this driver's behavior. This driver tends to come every day at around 8:00 am and stays until 12:00 pm and comes every week day, so it is likely to be a worker. There are also some gaps in the appearance, which could correlate to either the weekend or some other event.

Worker Profiles

We study the profile of four workers. The first case of a worker profile was already shown in Fig. 5.12b. This car is systematically seen at a fixed time at 8:00 am (except for a day) during the week and always leaves at 5:00 pm. Due to the working hours, we can deduce that this profile belongs to a full-time worker of the company. We can also extract more information from these patterns. On Day 12, they went outside to have lunch and we measured their car's IDs at a time around 12:00 pm to 1:00 pm, as it took around 1 hour to have lunch. Another insight we can extract is that their car is never seen on Fridays, because they usually work from home on those days. Thus, from TPMS transmissions, we can also infer the remote/in-person work patterns of individuals.

Fig. 5.12a shows the observations of Car 1 over the course of the measurement campaign. Some extra observations can be made from this temporal profile in addition to those made in Section 5.8.2. The pattern we describe is clearly visible, since the worker usually arrives at 8:00 am and leaves at 12:00 pm. However, by overlaying weekends and local holidays, we can further explain the gaps in the data. This driving pattern corresponds to a part-time worker, which is actually the case.



(a) Part-Time Worker.

(b) Full-Time Worker.

(c) External Worker.

(d) External Part-Time Worker.

Figure 5.12: Weekly patterns of different Worker profiles. X-axis represents the hour of the day, and the Y-Axis the Day. Each hour from each day that one of the 4 IDs of a car was seen, we paint that square black. Weekends and Holidays are shaded in blue and red respectively.

Apart from the regular working hours, we can also observe anomalies. First, on Day 14 the driver left at their usual time, but appeared later during the evening. This can be explained as that driver attended a university course in the vicinity and on their route back home, they passed through a road nearby the workplace. The next day, we can also see that the same driver attended another class and decided to go back through the same route. We see that, in the second day, the workplace was closed for holidays and yet we were still able to capture transmissions from the nearby road. This further enhances our intuition on the feasibility of deploying a car tracking infrastructure, since we are able to reliably capture transmissions from nearby roads and moving cars. These transmissions were captured by only one of the IDs of the car, thus reinforcing the car matching approach proposed in Section 5.7.4.

Fig. 5.12c has a similar pattern as Fig. 5.12b, but there are some key differences. First,

we see more transmissions during the day (the brands are different), so we can capture their car transmissions more often. The working pattern is quite stable as well; however, there are some days where we see that the car did not appear for a few consecutive days. This pattern actually matches that of an external worker/collaborator at the building. The gaps during some days are explained by the fact that this person was working elsewhere or on a trip. In the last week of the measuring campaign, that worker went on a longer trip and was not visible throughout the time. Apart from tracking individuals' patterns, the study shows that companies could use this information to track their employees without their consent or knowledge, as transmissions cannot be controlled by drivers.

The next case is of an external part-time worker, which comes a few days every week and does a shorter schedule than full-time workers. What is interesting is that we can capture TPMS transmissions every hour even when the vehicle was not moving. During our experiments we observe that each TPMS brand transmits with different strategies, as it can be seen in Fig. 5.12. We observe that TPMS sensors used by **Toyota** tend to transmit continuously, brands like **Ford** or **Nissan** do it less regularly, and brands like **Renault** only transmit when movement is detected on the wheels.

Insights into Pressure

Additional information can also be inferred from the data provided by the open source decoders. For example, the pressure of the tire could indicate the size and weight of a car. For example, larger SUVs or trucks tend to require higher pressures. In addition, it could provide information on the weight of the load of a truck (the higher the load, the higher the pressure on the tires), which could be used in combination with the previously described analysis and allow more precise targeting.

In Fig. 5.13, we show the pressure of the 4 tires for Car 1. From this, we can infer that it is a utilitarian sized vehicle as the maximum pressure is around 230 kPa (note that due to temperature, we can expect some variation in the pressure). Another insight is that during the course of our measuring campaign, the driver noticed pressure loss in the front wheels, and we can observe that, starting on the 13th day, they had inflated the tires.

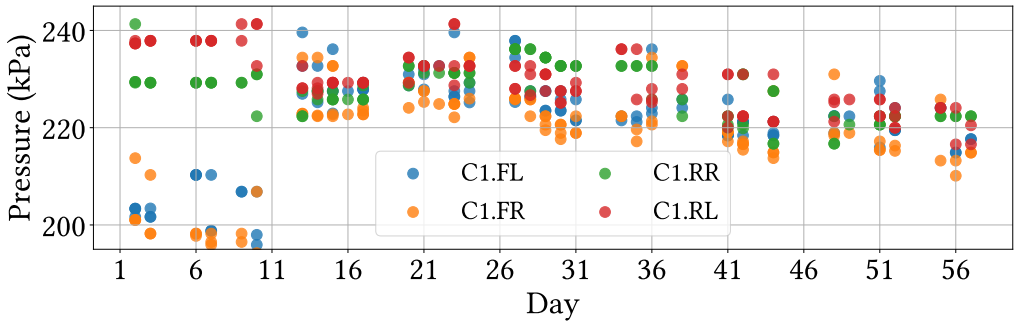


Figure 5.13: Tire pressure of the selected car.

5.9 Discussion

We have demonstrated the feasibility of deploying a low-cost receiver network that can capture tire broadcasts even at relatively large distances. Our findings show that by looking at common bits and time correlation techniques it is possible to identify individual cars and track them across different tires sensors.

When it comes to traffic surveillance, vision-based approaches have been widely popular [129], [130], as cameras are low-cost and already integrated in many vehicles and roads. However, camera-based systems require direct LOS on the road and its vehicles, and may not work well under challenging meteorological or low light conditions (for example, at night). Another issue these systems face is the higher computational cost required to process images in real time. TPMS tracking does not require LOS and is mostly unaffected by lighting or weather conditions. In this paper, we could capture tire messages within a radius of 55 m, which could be further increased with higher quality antennas or receivers specifically designed for the 433 MHz band.

About the feasibility of such deployments is that tire transmissions are non-deterministic and happen in time windows of around 30 seconds. At speeds of 30, 50, 80 and 120 km/h, cars travel 250, 400, 600 and 1000 m, respectively. If we take into account the 4 tires of a car, we can estimate that, on average, a car will then transmit every 60, 100, 150 and 250 m.

From these calculations, we can deduce that a TPMS tracking infrastructure would be more feasible in urban scenarios and close to intersections or traffic lights, where vehicles tend to slow down. Knowing that the range of capture is around 110 m, 1 receiver would be sufficient to cover a road up to 50 km/h and up to 3 would be needed on a highway with cars traveling at ˜120 km/h.

We can also estimate the probability of message collision based on the number of cars present in a road. In Section 5.4 we show that a typical TPMS transmission takes around 5 ms to be completed and we assume each tire sends a transmission every 30 seconds. The access to the medium can be modelled as unslotted Aloha, assume a throughput G of:

$$G = N \times \frac{L}{T} \tag{5.3}$$

where N is the number of TPMS sensors, L is the frame time (5 ms) and T the total window time 30 s. We can thus estimate the probability of collison by $P = e^{-2G}$. In Fig. 5.14 we show the probability that the spectrum receiver captures a particular TPMS transmission without collision. Even for a road with 150-200 cars (600-800 TPMS devices), the probability would be between 81-76%. We showed in Section 5.7.4 that it is not needed to track all 4 sensors of a car always, even with little overlap it could be possible to reconstruct the TPMS sensors of a single car. This shows that even in very congested roads, it would be feasible to capture and track vehicles.

During our work we also observed different transmission strategies. Brands like Toy-

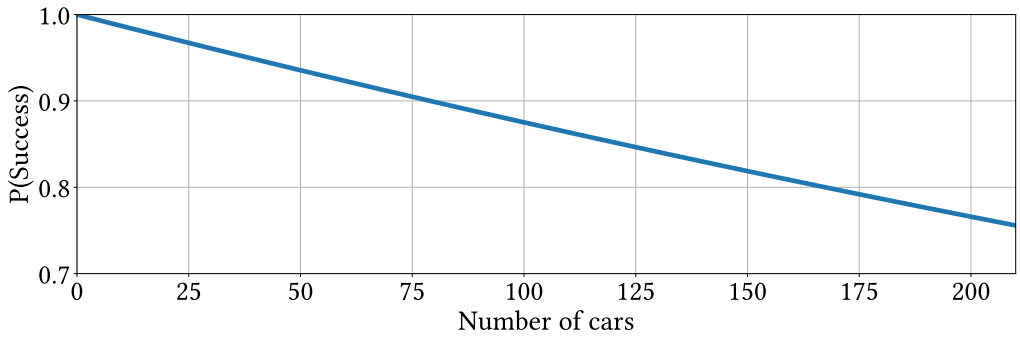


Figure 5.14: Probability of no message collision given a number of cars on the road.

ota tend to transmit at high power even when the car is parked, leading to higher probability for any of the vehicle's TPMS transmissions to be captured. We argue that this is unnecessary, as there is no risk for accident while the car is parked, and continuously sending might discharge the battery sooner and lead to information leaks as shown in this paper.

Unexplored aspects of this work include tracking the trajectory with more spatially distributed receivers. In such scenarios, in addition to temporal patterns, it would become feasible to uncover mobility patterns across cities. We believe that the techniques presented here could be transferable to such scenarios in the same manner as we sending spoofed messages, as shown in [109]. Combining this method with our work, an attacker could focus on specific targets with ease and disrupt individuals or traffic in a given area.

Besides, from July 2022, TPMS became mandatory in Europe also for buses, coaches, heavy trucks, and heavy trailers [119]. Larger vehicles tend to prefer dTPMS so as to maintain better control over their tires. However, this comes with the risk that a receiver network can capture transmitted TPMS IDs. By tracking IDs and pressure, it could be feasible to target larger vehicles like trucks, learn their daily routine, and send spoofed signals to the truck, causing them to stop and perhaps disrupt the logistics chain of companies.

With regards to solutions to avoid the privacy risks of dTPMS we envision 2 possible solutions: The former would be to add an encryption mechanism to the RF transmissions. The encryption keys could be burned in the memory of the sensor or use a rolling-key mechanism as is used in car key fobs, for instance. The other alternative would be to switch to iTPMS. Even though it is less accurate, we believe that for the majority of drivers, the risks attached to sending data in clear outweigh the benefits of accurate pressure and temperature readings for each tire.

5.10 Conclusion

We have shown that TPMS signals, originally introduced to improve road safety, can be misused to track vehicles and therefore their owners through a network of low-cost software-defined spectrum receivers. The lack of encryption and standardization is one of

the root causes, despite TPMS sensors being embedded in new cars for more than a decade. This information could be used by attackers to infer mobility patterns of individuals or to track people around cities if multiple sites are available for recording. Furthermore, we argue that it is rather easy in practice to link TPMS sensors with a specific person of interest. A TPMS signal receiver could be linked to a camera or, if the person of interests home address is known, a targeted recording at the persons home could reveal their unique TPMS sensor IDs. Once these IDs are known, the movements of that person could be tracked with cheap publicly available software-defined radios. Attackers could use this information to learn, predict, and exploit a persons movement, points of interest, and behavior patterns. Although there exist proposals to enhance privacy for TPMS sensors, we could not find any real deployment from manufacturers. Therefore, we urge legislators and manufacturers to take the necessary steps to improve the privacy and security of the pressure monitoring system.

PART IV

HIGH-END SDR IN LOW-LEVEL APPLICATIONS

6 5G Positioning with SDR radios

6.1 Introduction

The evolution of 5G technology, underpinned by the 3GPP's ongoing standardization efforts, marks a pivotal shift in the telecommunications landscape. Beginning with Release 16, the focus has been on enhancing the capabilities of 5G networks to support precise localization services, a critical component for a myriad of emerging applications [131], [132]. The integration of advanced localization features into 5G networks is not just a technological advancement; it represents a paradigm shift in how mobile networks can be leveraged, targeting high-accuracy and low-latency positioning through increased bandwidths and flexible numerology within the 5G physical layer.

The ambition to achieve centimeter-level accuracy, as outlined in Release 17, is a testament to the growing importance of 5G localization accuracy, low latency, network and device efficiency [133], for commercial use cases, Industrial IoT and especially for time-sensitive applications like remote control systems. Release 18, known as 5G Advanced, further enhances these capabilities, introducing improvements such as New Radio (NR) carrier phase measurements and sidelink positioning [134], [135] for out-of-range scenarios and Radio Access Technology (RAT)-dependent techniques for better accuracy under NLOS conditions [136]. With the initiation of Release 19, the exploration of wireless sensing technologies and mobile XR further emphasizes the strategic role of AI/ML in localization services in the future of wireless networks [137].

While the 3GPP has established a robust framework for 5G positioning with its comprehensive technical reports and specifications [133], [138], [139], the practical realization and evaluation of these technologies in real-world scenarios remains a challenge. This complexity is further accentuated by the diverse range of positioning use cases outlined in 3GPP documents [133], [139], each with distinct performance accuracy and Time to First Fix (TTFF) requirements. Simulations and preliminary experimental setups have been the primary methods for assessing 5G positioning performance to date [140]–[150].

Positioning using LTE signals has been explored in recent years, using them as sig-

nals of opportunity to enhance localization accuracy in other fields. To the best of our knowledge, only a few pioneering studies have begun to conduct experimental research with Over-The-Air (OTA) 5G signals [151], [152].

In this field, open-source platforms like OTA [153], srsRAN [17], and Aether [154], when paired with SDR devices such as Ettus USRP B210, HackRF One, LimeSDR, BladeRF, and the more advanced 5G-oriented Ettus USRPs X310, N310 and X410 [155], provide a robust framework for implementing and testing the 3GPP stack. These open-source platforms and commercial devices offer a unique opportunity for rapid prototyping and testing new algorithms and paradigms in real-world OTA conditions.

Despite these technologies' potential, their application in 5G localization and positioning demands significant effort in hardware and software setup and configuration.

In this chapter, we provide a deep understanding of 5G positioning in real-world conditions building on our preliminary insights [4], and stressing prior work over three dimensions: 1) We assess the impact of increasing the number of gNBs, a strategy aimed at enhancing the spatial diversity and accuracy of the localization system. 2) We explore the benefits of expanding the operational bandwidth up to 100 MHz, the largest bandwidth currently allowed by the 3GPP standard for communication below 6 GHz; larger bandwidth is expected to provide finer resolution in time-based measurements, leading to more precise localization. 3) We delve into oversampling and moving average signal processing strategies, examining the effects of sampling rates up to four times at 100 MHz and moving average window up to 50 measurements; these approaches are hypothesized to improve the accuracy of TOA measurements and reduce the effect of quantization errors, a critical factor in precise localization.

The results not only enhance the theoretical understanding of 5G localization but also provide practical insights and methodologies for application and experimentation in real-world settings that can be of interest both for the research community and for the telecommunication engineers that will build new 5G and beyond infrastructures. In fact, we demonstrate how strategic decisions in network infrastructure design and signal processing techniques can be effectively evaluated, offering valuable guidance to researchers and industry professionals seeking to enhance localization performance in contemporary 5G and beyond networks.

6.2 5G Positioning: State of the Art

This section introduces and describes the main elements of the 3GPP standardization of 5G localization and examines the role of existing open-source platforms in implementing location services within 5G mobile networks.

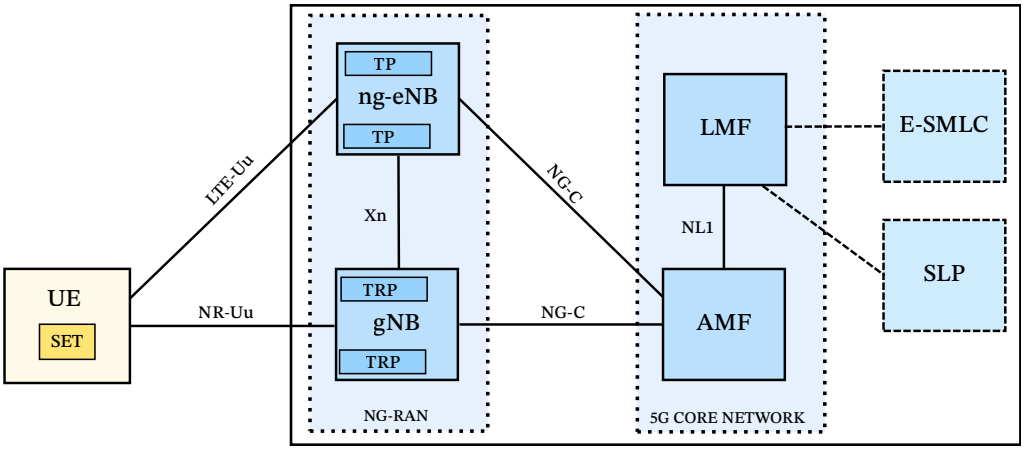


Figure 6.1: 3GPP 5G Positioning architecture [138].

6.2.1 5G Positioning Architecture

The implementation of 5G positioning, now advancing with the innovations of 3GPP Releases from 16 to 19, leverages an optimized architectural design for efficient transmission of location signals between gNB and UE, and for the seamless exchange of location measurements with the 5G Core Network. The evolving architectural specifications for 5G positioning, reflecting these technological strides, are detailed in [138].

As depicted in Fig. 6.1, the composition of the 5G positioning architecture encompasses the following components:

- UE that may carry out measurements of DL signals from the Next Generation Radio Access Network (NG-RAN), or may send Reference Signals in uplink for positioning;
- NG-RAN, comprised of gNBs and Next Generation evolved NodeB (NG-eNB) network elements, which may contribute through radio signal measurements and convey these measurements to the Location Management Function (LMF). A gNB can be composed of one or multiple Transmission-Reception Points (TRP), a set of geographically co-located antennas (e.g. antenna array). In our experimental studies with downlink positioning, TRPs are geographically separated, and therefore we consider one TRP per gNB;
- The Access and Mobility Management Function (AMF), which is responsible for handling connection and mobility management tasks, including positioning for a target UE, as per request for some location service associated or on behalf of a particular target UE;
- LMF, which is responsible for managing and supporting a range of location services for target UEs, including computation of user position and the delivery of assistance data.
- The Enhanced Serving Mobile Location Centre (E-SMLC) was first introduced in LTE network architecture. The LMF may communicate with E-SMLC with a proprietary interface to access location information from older networks;
- The SUPL Location Platform (SLP) is a server-based technology for providing secure and accurate location services to mobile devices on the user plane. The LMF may

communicate with the SLP with a proprietary interface.

The LMF plays a crucial role in receiving location requests from the AMF and interacting with the NG-RAN to obtain both uplink and downlink position measurements. The LMF may also directly engage with the target UE to provide requested assistance data or calculate a location estimate. The LMF determines the positioning methods to be employed based on various factors such as the LoCation Service (LCS) Client type, the desired Quality of Service (QoS) class [156], UE positioning capabilities, and NG-RAN positioning capabilities, and invokes these methods in the UE and serving NG-RAN.

5G positioning signals: 5G positioning signals have been introduced to achieve higher accuracy compared to LTE positioning [157]. These signals include the uplink SRS and an improved LTE PRS version for the downlink. In this work, we delve into the latter, as our focus is on downlink positioning methods. PRS signal provides superior accuracy through the use of a diagonal or staggered physical layer resource element scheme and superior coverage and interference suppression through the coordinated transmission of gNBs. This coordinated transmission avoids interference from adjacent cells and maximizes signal strength over multiple symbols throughout the NR bandwidth.

5G Positioning Schemes: As of Release 16, 3GPP has standardized the following 5G positioning schemes [138]: Downlink Time Difference of Arrival (DL-TDOA), Uplink Time Difference of Arrival (UL-TDOA), Downlink Angle of Departure (DL-AOD), Uplink Angle of Arrival (UL-AOA), Multi-Round Trip Time (Multi-RTT) and NR Enhanced Cell ID (NR E-CID). The recent releases have further refined these schemes, enhancing their accuracy and efficiency. In addition to the Radio Access Technology (RAT)-dependent solutions, another group of RAT-independent solutions can be used for positioning, such as GNSS, WLAN, Bluetooth, barometric pressure, and terrestrial beacon system (Terrestrial Beacon System (TBS)). In the following, we focus only on RAT-dependent schemes. The 5G network supports five positioning modes:

- Hybrid positioning: 5G network employs multiple positioning schemes.
- Standalone positioning: UE autonomously, without network assistance, uses one or more methods from the mentioned schemes.
- UE-based: UE computes its position estimate using measurements provided by the LMF. It is supported by DLpositioning schemes.
- UE-assisted/LMF-based: measurements are provided by the UE to the LMF to be used in the computation of a position estimate. It is supported by DLpositioning schemes.
- NG-RAN node-assisted/LMF-based: measurements are provided by the NG-RAN node to the LMF to be used in the computation of a position estimate. It is supported by UL positioning schemes.

As mentioned earlier, each gNB consists of one or multiple TRPs. Nevertheless, to simplify the description, we simply refer to gNBs as the element in the NG-RAN involved in the procedures of positioning in the rest of the chapter. In the DL-TDOA scheme, the UE calculates the Reference Signal Timing Difference (RSTD) or TDOA of the PRSs trans-

mitted by two distinct gNBs. The measured TDOA, is then communicated to the LMF, which uses these timing differences to estimate the UE's position. The UL-TDOA scheme is conceptually similar to DL-TDOA but uses uplink signals and reverses the signal direction. Here, the gNB is responsible for measuring the Relative Time of Arrival (RTOA) of multiple SRSs emitted by UEs. The gNB communicates these measurements to the LMF. In the DL-AOD scheme, the UE measures the received power of downlink reference signal beams from multiple gNBs. These power measurements are used to estimate the position angle of the UE and are communicated to the LMF. The LMF then solves the derived equations to determine the UE's position based on these multiple gNBs estimates. Mirroring DL-AOD, UL-AOA utilizes uplink signals. The gNBs measure the angle of arrival of the uplink signal from UE, which is communicated to the LMF, and the UE's position is determined from these multiple-angle multi-gNBs measurements.

In the Multi-RTT positioning scheme, both uplink and downlink signals are used. The UE and a minimum of three gNBs exchange reference signals (SRS and PRS) to measure the Round Trip Time (RTT), deducing the RX-TX time difference. The LMF synthesizes these measurements to calculate the distances between the UE and each gNB, estimating the UE's position at the intersection point of the circles formed with these distances as radii. The NR E-CID positioning technique involves the estimation of the UE position using information about its connected NG-eNB, gNB, and cell. This information can be acquired through paging, registration, or other similar methods. NR E-CID positioning approach takes this a step further by incorporating additional UE measurements, NG-RAN radio resources, and other metrics, significantly enhancing the accuracy of the UE location estimates.

6.2.2 Mobile Network Synchronization Solutions

In the context of 5G localization, synchronization is a critical aspect to guarantee precise RSTD and RTOA time measurements in downlink and uplink, respectively. A lack of high-precision network synchronization would inevitably affect the overall location accuracy, making the overall effort for 5G positioning ineffective.

Various existing solutions for mobile network Internet Service Provider synchronization are available to achieve this. The most popular solution is utilizing the GNSS signals, such as GPS, to synchronize in time and frequency. However, GNSS can be susceptible to environmental factors or disturbance attacks that affect its reliability; consequently, it cannot be used standalone as a Primary Reference Clock (PRC).

For providing a PRC signal to the network, the older ITU-T G.811 was suitable only for frequency synchronization. More recently, the G.8272.1 enhanced Primary Reference Time Clock (ePRTC) is a newer, more specialized solution standardized by the ITU-T for network synchronization, which requires the usage of cesium clocks for centralized grand master clocks and quartz/rubidium for distributed master clocks and boundary

clocks [158]. ePRTC offers sub-nanosecond accuracy, with time, phase, and frequency aligned and calibrated to the GNSS signal over time. This makes ePRTC well-suited for 5G positioning.

For the RAN domain, the main option would consist of using the Precision Time Protocol (PTP) and Synchronous Ethernet (SyncE) technologies to enable synchronization with high accuracy and stability between network nodes at the edge. Using this combined approach, PTP provides time-of-day information and phase, while SyncE provides highly stable frequency synchronization. ITU-T has defined a set of telecom standards for 5G networks that are built on top of PTP, namely G.8275.1 and G.8275.2. While G.8275.1 is the most accurate solution, G.8275.2 is more practical to implement. Both deliver highly accurate frequency synchronization, phase synchronization and time of day from the PRC signal. Nevertheless, G.8275.1 is recommended to meet the relevant time synchronization requirements [159]. Finally, in the RAN domain, over-the-air synchronization could provide a valid alternative as it is based on round-trip time measurement sent over the radio interface between neighboring base stations [159]. However, it lacks work in standardization.

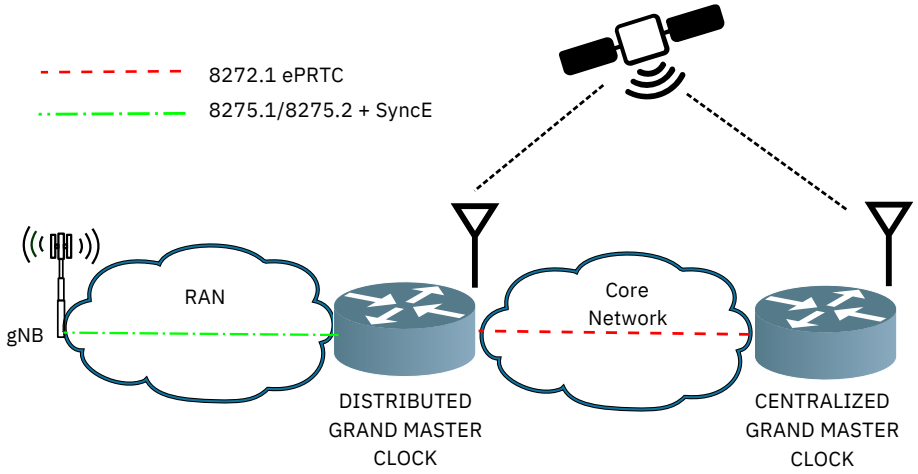


Figure 6.2: Mobile network synchronization for 5G networks.

In conclusion, and as summarized in Fig. 6.2, while each synchronization solution has its own advantages and disadvantages, the combination of ePRTC in the 5G Core Network (5GCN) with ITU G.8275.1/G.8275.2 in the RAN domain provides a promising solution for ensuring precise synchronization in 5G positioning and advanced mobile network applications due to its excellent performance, being also the most suitable approach for supporting positioning services based on very accurate timing information.

6.2.3 Open-source platforms

The main open-source software software platforms for 5G development are: **OpenAirInterface**, **srsRAN** and **Aether**. A description of their capabilities has already been presented in Chapter 1. The development of our testbed used **OpenAirInterface**.

6.3 System Model

This section introduces the main elements of our location system model and the associated signal processing methodologies. In the subsequent sections of the chapter, we assume a 2D location scenario, i.e., considering only the x and y coordinates while neglecting the z coordinate. Furthermore, assuming a 2D environment, we focus on horizontal positioning accuracy, which is often of primary interest in many practical localization applications. For clarity, it is worth noting that using 5G reference signals dedicated to positioning in a 3D localization scenario is possible.

We consider a network of neighboring N_g gNBs, which estimates the position of N_u users. The ith gNBs is at $\mathbf{p}_{\text{gNB}}^{(i)} = \left[x_{\text{gNB}}^{(i)}\ y_{\text{gNB}}^{(i)}\right]$, while the unknown position of the jth UE is at $\mathbf{p}_{\text{UE}}^{(j)} = \left[x_{\text{UE}}^{(j)}\ y_{\text{UE}}^{(j)}\right]$. A location-dependent time-based measurement is performed for each gNB. Fig. 6.3 provides a visual representation of a typical network deployment following our system model.

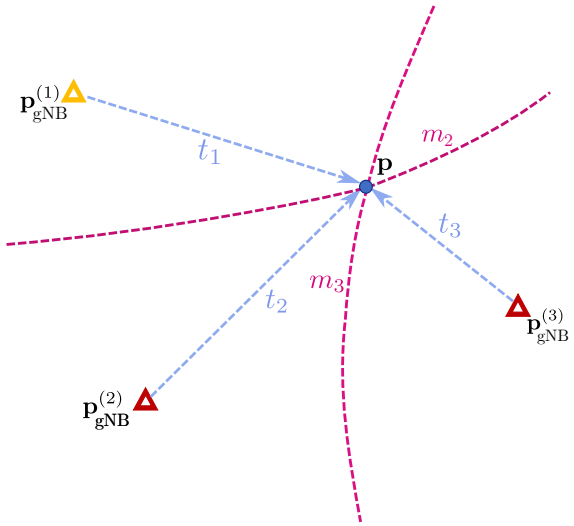


Figure 6.3: Example network deployment with $N_g = 3$ gNBs (triangles) and $N_u = 1$ UE (circle). The TDOA $m_2 = t_2 - t_1$ and $m_3 = t_3 - t_1$ are measured considering the first anchor as the reference (yellow triangle). The corresponding hyperbola is illustrated in magenta.

6.3.1 Comparative Analysis of PRS and SRS for Positioning

In the context of utilizing 5G reference signals for positioning, it is essential to distinguish between PRS and SRS. Both signals have unique characteristics and applications within the network, influencing their suitability for positioning tasks.

- Signal Design and Purpose: Specifically designed for positioning, PRS is optimized for time-based measurements, which are crucial for accurate localization. The structure of PRS minimizes ambiguity in detection, making it highly effective for positioning in diverse environments. Hence, SRS primarily aims to estimate uplink CSI, facilitating

network decisions regarding beamforming, power control, and user scheduling. Although SRS can be repurposed for positioning, its design is not inherently optimized for this application.

- Robustness and Accuracy: PRS exhibits superior robustness to multipath effects and NLOS conditions, critical attributes in urban and indoor environments where direct paths between the transmitter and receiver are often obstructed. SRS, instead, while useful for its intended purpose of channel estimation, is less robust in the face of multipath and NLOS conditions when repurposed for positioning, potentially leading to decreased accuracy.
- Network Resource Utilization: PRS utilizes dedicated resources in the frequency and time domain, ensuring that the signal integrity is preserved for the precise measurement requirements of positioning tasks. SRS instead, though flexible in configuration, using SRS for positioning could lead to increased resource consumption or necessitate trade-offs with its primary function of channel sounding. SRS time-domain and frequency-domain allocation is limited compared to PRS(i.e., max 4 OFDM symbols in 1 slot).

Given these considerations, while SRS presents a theoretically viable option for positioning, its performance, robustness, and efficiency for such applications are inherently constrained by its primary design for channel sounding. Conversely, PRS, with its positioning optimized design, offers significant advantages in accuracy, reliability, and compliance with positioning standards. Therefore, in our system model and subsequent analyses, we focus on PRS. This choice is motivated by our objective to explore the use of SDR platforms for 5G positioning, leveraging the inherent advantages of PRSdesigned specifically for this purpose.

6.3.2 TOA Estimation

The relative TOA for the signal transmitted by the ith gNB and received by the jth UE is expressed as:

$$t_{i,j}(\mathbf{p}_{\mathrm{UE}}) = \frac{\|\mathbf{p}_{\mathrm{gNB}}^{(i)} - \mathbf{p}_{\mathrm{UE}}^{(j)}\|}{c} \tag{6.1}$$

where c is the speed of light. In OAI, PRS -based TOA estimation is implemented through a series of well-defined steps in accordance with the 3GPP standards for 5G positioning. Initially, Pilots are generated and modulated, serving as references for subsequent processing. Subsequently, the Resource Element (RE) offset of the PRS is calculated, followed by channel estimation and interpolation. Next, SNR estimation is conducted, and PRS channel estimates are transformed into a format suitable for the Fast Fourier Transform (FFT). The procedure then calculates the CIR in the time domain with IDFT oversampling up to 16x. Finally, the TOA is estimated by finding the peak estimate of the CIR as the maximum identified absolute value.

6.3.3 TDOA-Based Positioning Algorithm

The DL-TDOA technique leverages the PRS signal transmitted by a set of neighboring gNBs and received by the UE to measure the RSTD, or TDOA, between each pair of gNBs. This measurement is then employed in the multilateration or trilateration calculation based on the principles of hyperbolic geometry. Conventionally, the gNB with index $i = 1$ is designated as the reference for RSTD measurements.

The RSTD, as described in [160] Section 5.1.29, represents the relative time difference between the ith neighboring gNB and the reference gNB and is calculated by the jth UE as $m_{i,j} = t_{i,j}(\mathbf{p}_{\text{UE}}^{(j)}) - t_{1,j}(\mathbf{p}_{\text{UE}}^{(j)})$. Therefore, the measurement model for the TDOA is as follows:

$$m_{i,j} = \tau_{i,j} \cdot c = \left\| \mathbf{p}_{\text{gNB}}^{(i)} - \mathbf{p}_{\text{UE}}^{(j)} \right\| - \left\| \mathbf{p}_{\text{gNB}}^{(1)} - \mathbf{p}_{\text{UE}}^{(j)} \right\| + n^{(i,j)}(\mathbf{p}_{\text{UE}}^{(j)}) \tag{6.2}$$

Measurement noise $n^{(i,j)}(\mathbf{p}_{\text{UE}}^{(j)})$ is inherently dependent on the UE position, the SNR of the received signal, and the LOS conditions. To mitigate the risk of synchronization errors in RSTD measurements, all gNBs transmit the PRS simultaneously, thereby ensuring synchronized transmission from all gNBs.

6.3.4 Optimization Model for Positioning

We can formulate the optimization problem as a minimization of the L2-norm of the measured and estimated TDOA (for the sake of brevity we remove the superscript (j)):

$$\mathbf{p}_{\text{UE}} = \underset{\mathbf{p}_{\text{UE}}}{\arg\min}\ S(\mathbf{p}_{\text{UE}}) = \frac{1}{2} \sum_{\forall i,j} (\hat{m}_{ij} - m_{ij}(\mathbf{p}_{\text{UE}}))^2 = \frac{1}{2} \sum_{\forall i,j} r_{ij}^2 \tag{6.3}$$

where $\hat{m_{ij}}$ is the measured TDOA between gNBs i and j; $m_{ij}(\mathbf{p}_{\text{UE}})$ is the evaluation of Eq. 6.2 at the UE position $\mathbf{p}_{\text{UE}}$ and r_{ij} denotes the residual error between them.

For this method, the solution lies at the point where the gradient of $S(\mathbf{p}_{\text{UE}})$ is 0, which can be computed per transmitter coordinate (x, y) as:

$$\begin{aligned} \frac{\partial S}{\partial p_{\text{UE},x}} &= \sum_{\forall i,j} r_{ij} \cdot \left(\frac{p_{\text{UE},x} - p_{\text{gNB}_i,x}}{d_i} - \frac{p_{\text{UE},x} - p_{\text{gNB}_j,x}}{d_j} \right) \\ \frac{\partial S}{\partial p_{\text{UE},y}} &= \sum_{\forall i,j} r_{ij} \cdot \left(\frac{p_{\text{UE},y} - p_{\text{gNB}_i,y}}{d_i} - \frac{p_{\text{UE},y} - p_{\text{gNB}_j,y}}{d_j} \right) \end{aligned} \tag{6.4}$$

The optimization problem for the TDOA equations is non-convex, and requires more complex methods to obtain reliable solutions. There also exist techniques to linearize these equations and obtain fast (but suboptimal) solutions. These approaches are further explained in Section 6.4.3.

6.3.5 Geometric Dilution of Precision

The GDOP quantitatively represents the impact of the geometric configuration of gNBs on the accuracy of UE localization. It is derived from the Jacobian matrix, which relates the TDOA measurements to the UE's position.

For clarity in our 5G localization system analysis, it is important to note that while we are addressing a two-dimensional positioning problem, we typically refer to the HDOP in such cases. However, for the sake of generality, we choose to use the term GDOP in the following discussion. Given a network of N_{g} gNBs, with ith position $\mathbf{p}_{\mathrm{gNB}}^{(i)}$, N_{u} UEs with jth position $\mathbf{p}_{\mathrm{UE}}^{(j)}$ and the TDOA measurement given by Eq. 6.2, since one gNB is used as a reference, we have $N_{\mathrm{g}} - 1$ independent TDOA measurements for each jth UE. The corresponding Jacobian matrix in two dimensions $\mathbf{J}$, representing the sensitivity of the TDOA measurements to the position considered for UE, is calculated as follows:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial m_2}{\partial x_{\mathrm{UE}}} & \frac{\partial m_2}{\partial y_{\mathrm{UE}}} \\ \vdots & \vdots \\ \frac{\partial m_{N_{\mathrm{g}}}}{\partial x_{\mathrm{UE}}} & \frac{\partial m N_{\mathrm{g}}}{\partial y_{\mathrm{UE}}} \end{bmatrix} \tag{6.5}$$

where each element of $\mathbf{J}$ is the partial derivative of the TDOA measurement $m_{i,j}$, based on the measurement model defined in Eq. 6.2, with respect to the jth UE's coordinates $x_{\mathrm{UE}}^{(j)}$ and $y_{\mathrm{UE}}^{(j)}$.

Finally, GDOP is calculated as the square root of the trace of the matrix product $(\mathbf{J}^T\mathbf{J})^{-1}$:

$$\mathrm{GDOP} = \sqrt{\mathrm{trace}((\mathbf{J}^T\mathbf{J})^{-1})} \tag{6.6}$$

A lower GDOP value implies a more favorable geometric configuration, leading to enhanced accuracy in the UE's position estimation. Conversely, a higher GDOP suggests a less ideal geometry, potentially decreasing the precision of location estimates. This underlines the importance of strategic placement of gNBs in our network. Our experimental setup, featuring four gNBs, aims to minimize the GDOP, thus ensuring a geometrically optimal arrangement for precise localization.

In addition to GDOP, the CRLB provides a more general theoretical benchmark for the minimum variance of an unbiased estimator [161]. The CRLB is defined based on the FIM for the positioning error, and a lower CRLB value indicates a higher potential accuracy in position estimation.

6.4 Experimental Setup

This section delineates the deployment of our 5G SDR-based experimental localization system, detailing both the software and hardware configurations, along with the network

configuration and signal processing methodologies employed.

6.4.1 Mobile Network Software

In our experimental setup, we use the OAI open-source framework to implement both RAN (gNB and 5G UE) and 5GCN for its more advanced development concerning reference signals for positioning. We decided to focus on the DL PRS signal to test 5G positioning; however, applying the same procedures using the UL SRS signal is possible. The OAI release tagged *2023.w38* is employed for our experiments. TOA measurement estimation for PRS is executed using the *peak_estimator* function inside the *nr_prs_channel_estimation* function, located within the *nr_dl_channel_estimation.c* file. To enable TOA measurement estimation for UL SRS in OAI, it would be necessary to integrate the *peak_estimator* function into the *nr_srs_channel_estimation* function, found in the *nr_ul_channel_estimation.c* file. The PRS signal is configured to be transmitted every 5 ms.

6.4.2 Hardware

The experimental testbed utilized in this study consists of two customized workstations, one powerful laptop, and five SDRs, which are utilized to emulate the operator network (5GCN and gNB) and the user terminal.

SDR: Five Ettus USRP X310 devices, equipped with UBX160 daughterboards [162], constitute the backbone of our SDR setup. Four are used as gNBs, while the fifth acts as the UE. These devices, covering a broad radio frequency spectrum from 10 MHz to 6 GHz, support all NR FR1 frequency bands with up to 160 MHz bandwidth. Using the high-speed 10 GbE interface, a sampling rate of 184.32 MS/s can be achieved, enabling a theoretical distance resolution of 1.62 m. We used NR-specific Linx ANT-5GWWS3 [163] SMA multiband antennas. While both the Ettus USRP N310 and X310 were evaluated in our experimental setup, we encountered significant interoperability issues due to the differences in their supported master clock rates. Specifically, the USRP N310 supports master clock rates of 122.88 MHz, 125.00 MHz, and 153.60 MHz, while the USRP X310 operates at higher clock rates of 184.32 MHz and 200.00 MHz. This discrepancy in clock rate compatibility presents a challenge for interoperability between the N310 and X310 in a unified experimental setup, as synchronized data acquisition and processing are necessary for communication and positioning.

Base station synchronizer: Precise synchronization between gNBs is achieved using the Ettus Octoclock [164], a high-precision clock generator and distribution unit. With its eight independent clock outputs, GPS receiver, phase-locked loop (PLL), and voltage-controlled crystal oscillator (VCXO) configuration, this device provides exceptional frequency stability and accuracy, essential for our mobile network localization testbed. Ac-

cording to the specifications, the accuracy of the 1 PPS of the built-in GPSDO is within 50 ns of the UTC RMS, with a frequency accuracy of 1 ppb after the GPS is locked. A relative frequency tolerance of 1 ppb means a maximum time offset of 1 nanosecond in one second. Note that in a larger scenario, the octoclock would not be a viable solution to distribute timing information. The aforementioned USRP X310 has GPS input and a GPSDO. It would be possible to use this as a method to synchronize all gNBs to GPS time and achieve a similar timing accuracy. It is however out of scope for this chapter.

Workstation: The computational backbone comprises two Intel Core i9 powered workstations, each with 18 cores clocked at 3 GHz, running Ubuntu 18.04 LTS Linux operating system with a low-latency kernel. We strategically utilized Ubuntu 18.04 LTS for its proven stability and extensive support for critical legacy hardware and software components. These workstations implement the 5GCN and four gNBs. To implement the UE, an HP “ZBook Power G9 Mobile Workstation" powered by Intel Core i7 CPUs with 14 cores (6 high-performance cores clocked at up to 4.8 GHz and 8 efficient cores clocked at up to 3.7 GHz) running Ubuntu OS 22.04 LTS is utilized. For the sake of clarity, initially, our experiments were conducted on a stable setup using Ubuntu 18.04 LTS, prioritizing research continuity over system updates due to time constraints. Recognizing the importance of staying current, we later upgraded the two workstations to Ubuntu 22.04. We validated our setup’s compatibility with this newer version, demonstrating our testbed’s robustness and adaptability to evolving software environments.

6.4.3 Signal Processing Techniques and Localization Algorithms

Our experimental approach to assessing how to improve 5G localization accuracy includes the study of advanced signal processing techniques.

Moving Average: Given the experimental challenges and environmental variabilities associated with 5G localization using open-source SDR-based frameworks, achieving consistent positioning performance is a complex task. To address this, we employ moving average techniques, which are instrumental in mitigating the impact of outlier measurements. By averaging out extreme values in time-based ranging data, the moving average method substantially improves the reliability and stability of our localization results, ensuring a more robust performance even in fluctuating outdoor conditions.

Oversampling Techniques: To counteract the presence of multipath interference, which leads to significant errors in the TOA estimates, we implement oversampling. Oversampling is a technique used to improve the accuracy of TOA estimation by increasing the number of samples collected. In the context of 5G technology, CIR oversampling can be crucial in obtaining finer TOA estimation. The CIR provides information about the time-varying characteristics of the radio channel, including reflections and multipath interference. When CIR is oversampled, the number of samples increases, providing a more detailed representation of the channel response. This results in a higher SNR, allowing for

a more accurate and robust estimate of the TOA. In OAI, oversampling is implemented using a technique called zero-padding, which involves inserting additional zeros on both sides of the baseband frequency-domain representation of the signal and then performing the IDFT, effectively increasing the sampling rate without adding any new information. By oversampling the CIR, it is possible to mitigate the effects of multipath interference, random noise, and other distortions that can affect the accuracy of TOA estimation. This leads to a more reliable and accurate estimate of the TOA, improving the performance of ranging measurements.

Localization Algorithms: To address the localization challenge in wireless networks, a variety of algorithms are employed, each offering distinct approaches and benefits. These include LLS, Non Linear Least Squares (NLLS), and the Brute-Force algorithm. Each method offers distinct methodologies for estimating the position of UE in a 5G network environment.

Linear LS simplifies the localization problem by linearizing the TDOA equations in a similar manner as [59]. This method assumes a linear relationship between the measured TDOAs and the unknown UE position. It is most effective in environments with minimal signal distortion but tends to be less accurate in scenarios with significant multipath effects or non-line-of-sight conditions.

NLLS directly addresses the non-linear nature of TDOA equations. It leverages iterative optimization routines based on gradient-based techniques to determine the direction of maximum descent and minimize the 2-norm difference between the predicted and actual TDOAs. Compared to the Brute-Force method, this method is more efficient and retains accuracy. However, the main drawback of NLLS methods is the initial point estimation, which can be overcome by performing an initial estimation using the coarse grid defined with the Brute-Force method. Implementations of successive points can then be achieved through methods such as Levenberg-Marquardt [45], BFGS [165], or Gradient-Descent and its variants [166], with similar results.

Lastly, the Brute-Force Algorithm takes a comprehensive approach, discretizing the area of interest into a grid and evaluating a cost function at each grid point. Since the estimated position is identified as the point that minimizes the minimum 2-norm discrepancy between the ideal predicted and estimated TDOA values, accuracy depends on the grid size, and the tradeoff between accuracy and processing time became crucial. However, this method, especially for finer grids, offers high accuracy due to its exhaustive search of the solution space.

Each algorithm's suitability varies depending on the specific environmental conditions and system requirements, making it essential to select the appropriate method based on the context of the localization task.

As a benchmark, we define the 'Ideal' case as the theoretical best-case scenario with minimal error influence, utilizing NLLS, chosen because brute force and NLLS yielded nearly equal results in our experiments. For comparative purposes, this scenario is char-

acterized by only process noise and without the constant sampling time offset. We define this 'Ideal' case as we note that measurements in each capture are affected by a constant offset (besides the expected measurement noise). The probable cause are hardware effects that cause small differences in TOA measurements. Due to lack of other mechanisms to estimate this offset behorehand, we assume it to be the 'Ideal' scenario. The offset can later be observed in Fig. 6.10.

6.4.4 Network configuration

Our methodology for enhancing 5G localization precision entailed a detailed exploration of various network configurations, specifically examining the impact of different numbers of gNBs and bandwidth parameters.

Network Deployment with Four gNBs: In a TDOA-based system, the localization accuracy heavily depends on the geometry of the gNBs relative to the UE. With three gNBs, TDOA localization is typically reliant on the intersection of hyperbolae formed by the time difference measurements between pairs of gNBs. While effective, this approach can be limited in precision due to factors like signal multipath, non-line-of-sight conditions, and synchronization errors. The addition of a fourth gNB not only provides an extra set of TDOA measurements but also enhances the geometric robustness of the system, introducing new hyperbolic intersections and significantly refining the localization process. The configuration of four gNBs also mitigates the effect of GDOP. By adding the fourth gNB, we lead to a lower GDOP value and, thus, higher accuracy in the calculated position of the UE.

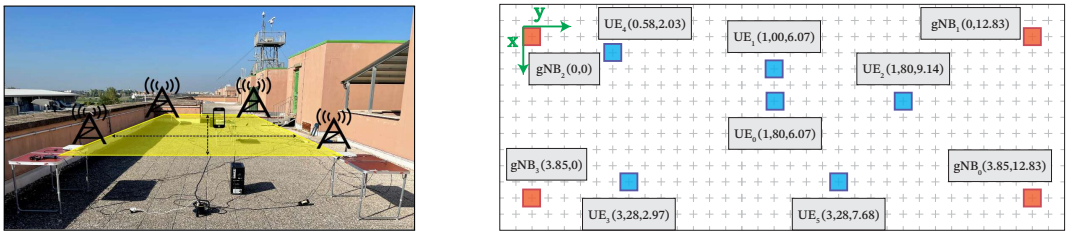
Bandwidth Configurations: We explore the impact of varying bandwidths - 40, 80, and 100 MHz - on the accuracy of localization. Higher bandwidths result in shorter impulse responses in the time domain, improving the precision of TDOA measurements due to more defined signal autocorrelation functions, offering sharper peaks and, consequently, more precise timing estimates. Essentially, increased bandwidth translates to improved time resolution, which leads to enhancing the accuracy of TDOA-based localization systems. This enhancement in time resolution positively affects the FIM, leading to more informative TDOA measurements. Consequently, a larger bandwidth not only enriches the FIM but also reduces the CRLB, indicating less variance in position estimates and thus increasing the overall localization accuracy.

6.5 Experimental validation

In this section, we present the deployment scenario and the experimentation settings, and discuss the experimental results.

6.5.1 Deployment Scenario

As illustrated in Fig. 6.4a, experiments were performed on an outdoor setting, a roof of the University of Rome, Tor Vergata. Experiments are conducted outdoors with clear GPS satellite visibility to ensure proper synchronization. The experimental testbed area is shown in Fig. 6.4b. Our testbed covers a surface of more than 50 m^2. As reported in Fig. 6.4b, we use six different selected UE locations to test our experimental positioning system.



(a) Testbed Setup.

(b) gNB and UE positions.

Figure 6.4: Experiment setup overview. The experimental area is shaded in yellow (Fig. 6.4a) with four gNBs and UE. Fig. 6.4b shows the positions of the gNBs and the 6 locations for the UE.

To prevent any disruption to other users, all tests are conducted with the caution of stopping the experiments when people approach the testbed. Furthermore, the output power of the SDR radios used in the experiments is thoroughly verified to ensure that the emitted 5G NR signal did not cause any interference outside the designated testing area.

The 5G NR TDD channel has been configured in the N78 frequency band (3500 MHz) for the experimental localization testbed. The channel has a variable instantaneous bandwidth of 40, 80 and 100 MHz (respectively, 106, 217 and 273 RB) and a sub-carrier spacing of 30 kHz.

NLOS conditions adversely affect TOA measurements for PRS in 5G networks by causing multiple propagation and an alteration of TOA estimates by delayed copies of the signal, which can be misinterpreted as direct signals, leading to incorrect distance estimates. In the following subsections, we will focus on the LOS scenario because our SDR testbed, constrained by low transmission power, is inherently suited for LOS scenarios and lacks the capability to manage the challenges posed by NLOS conditions effectively. Therefore, since we cannot handle NLOS scenarios with the hardware at our disposal, our study does not cover NLOS scenarios, emphasizing an area for future research with more capable hardware.

6.5.2 Moving Average Experimentation

In this subsection, we examine the influence of applying a moving average filter to TDOA measurements on position estimation accuracy. This analysis is supported by Fig. 6.5, which graphically illustrates the correlation between increasing average window sizes and the resultant position estimation errors.

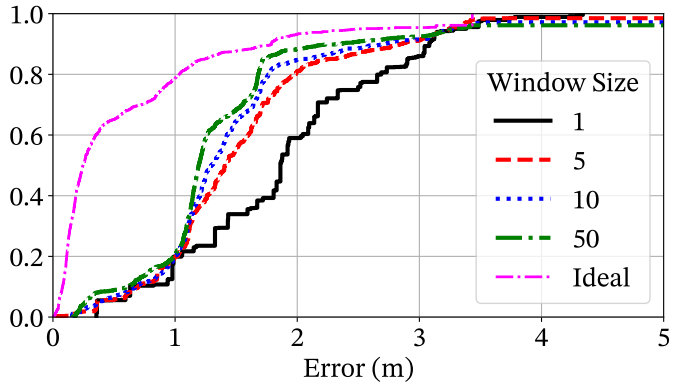


Figure 6.5: Empirical cumulative distribution function of the distance error using a Non-Linear Least Squares localization algorithm, four gNBs, no oversampling and 100 MHz BW with varying moving average window sizes applied to TDOA measurements.

As demonstrated in Fig. 6.5, implementing a moving average filter with a window size of up to 50 significantly reduces the localization error. Indeed, by using a Brute Force localization algorithm, four gNBs and 100 MHz BW, it is possible to achieve a localization error of 1.7 meters in 80% of the cases, an improvement of about 37.04% over the case without moving average. This reduction in error demonstrates the filter's efficacy in enhancing the accuracy of position estimations by smoothing out random noise and measurement fluctuations in the TDOA data. The filter effectively smooths out short-term fluctuations and random noise inherent in signal measurements by averaging TDOA measurements over a defined number of samples (the window size).

However, it is important to consider the trade-off between noise reduction and the potential delay introduced by larger window sizes. Especially in dynamic time-critical scenarios, it is crucial to optimize the window size to ensure that the filter does not overly smooth out the TDOA data, which could lead to inaccuracies in triangulation.

6.5.3 Oversampling improvement

To effectively show the impact of oversampling on location estimation accuracy, we present Fig. 6.6, which illustrates localization errors by implementing different oversampling factors. The oversampling analysis strategically focuses on UE's most representative positions 0, 4, and 5, to illustrate the system's performance across diverse scenarios: position 0 as a baseline, position 4 to assess performance under conditions close to a receiver, and

position 5 to evaluate robustness in a challenging environment at a corner and further from the receivers.

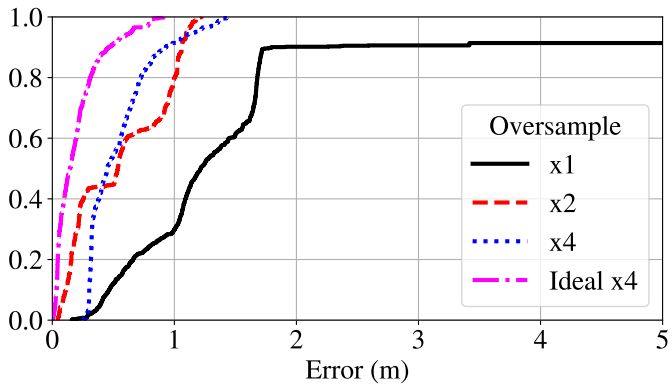
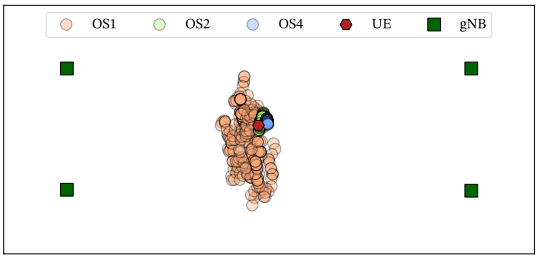
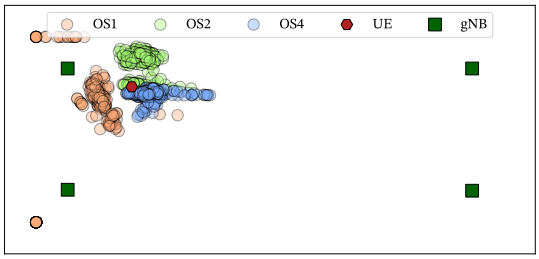


Figure 6.6: Empirical cumulative distribution function of the error using Non-Linear Least Squares localization algorithm, 100 MHz of BW, moving average window size 50 and varying the oversampling factor. We only consider positions 0, 4 and 5.

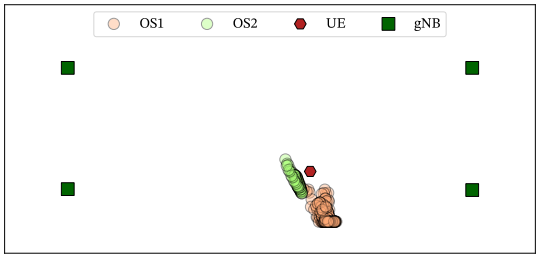
As shown in Fig. 6.6, by implementing a 4x oversampling rate, we can reduce the localization error to 0.7 meters in 80% of the cases, compared to 1.7 meters without oversampling, which results in a localization performance improvement of 58.82%. This notable improvement is attributed to the *ability of oversampling to generate more accurate TOA values, which are indispensable in location estimation algorithms*.



(a) UE position 0.



(b) UE position 4.



(c) UE position 5.

Figure 6.7: Impact of oversampling (OS) on the variance of three selected UE position estimates, with OS1, OS2, and OS4 corresponding to factors 1, 2, and 4, respectively.

Fig. 6.7 demonstrates the beneficial impact of oversampling on UE position estimations. In fact, the experimental results confirm what was argued in Section 6.4.3, showing

how effectively oversampling enhances the resolution of signal processing, allowing for a more detailed signal representation and, thus, a more precise extraction of timing information. As a result, the variance in position estimations is reduced, thereby improving the overall accuracy of the localization system.

These results collectively establish the efficacy of oversampling as a methodology to increase location estimation accuracy in wireless communication systems.

6.5.4 Network Configuration Experimentation

This subsection delves into the impact of varying the network configuration, particularly the number of gNBs, on the precision of position estimation in a 5G network. Fig. 6.8 presents a comparative analysis of positioning errors in three and four gNBs scenarios.

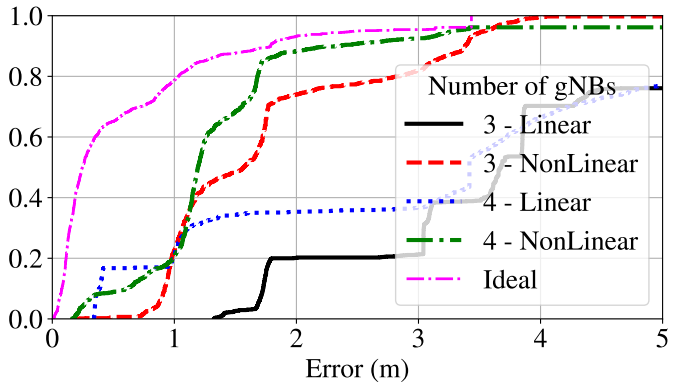
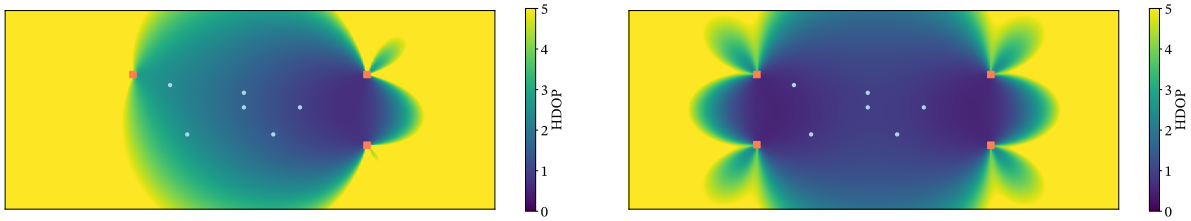


Figure 6.8: Empirical cumulative distribution function of the distance error using Non-Linear (i.e., Brute Force and NLLS) and Linear (i.e., LS) localization algorithms, with no oversampling 100 MHz of BW, moving average window size 50 and with network configurations of three and four gNBs.

Fig. 6.8 shows that the use of four gNBs leads to a reduced localization error of 1.2 meters, an improvement of 29.41% compared to the three gNB setup. When an extra gNB is added, the improvement in positioning accuracy can be attributed to improved network geometry, as previously discussed in GDOP subsection 6.3.5.

To better understand the impact of the placement of the gNB on the localization accuracy, Fig. 6.9 shows the theoretical HDOP in our experimental setup for 3 and 4 gNBs. Incorporating an additional gNB enhances the spatial distribution of signal sources around the UE, and the system benefits from additional reference points, which helps to refine the triangulation process and reduce the potential for positioning errors.

The findings of this experiment reinforce the concept that network configuration, specifically the number of gNBs and their spatial location, is a critical factor in achieving high accuracy in 5G positioning systems.



(a) HDOP with 3 gNBs.

(b) HDOP with 4 gNBs.

Figure 6.9: Theoretical HDOP in the experimental setup showing the spatial arrangement of UEs and 3-4 gNBs.

6.5.5 Localization Algorithms

This subsection evaluates the accuracy of localization using various algorithms, namely LLS, NLLS, and Brute-Force. Each of these algorithms offers a different approach to resolving the localization problem, with varying degrees of complexity and precision.

Since the Brute-Force algorithm discretizes the 2D space into a uniformly distributed grid and evaluates the cost function at each point, a finer grid leads to better accuracy but requires more processing time. To speed up the algorithm, a technique used in this paper is to begin with a coarser grid and refine the blocks with the lowest value. The space is initially divided into a $15 \times 7\,\mathrm{m}^2$ grid with a step size of 0.5m, and at each iteration, the grid point with the lowest objective function value and its neighbors are selected, and the step size is reduced to maintain a constant number of grid points. We set the maximum number of iterations to 10 and the tolerance to 1×10^{-4}.

Fig. 6.8 demonstrates the performance of these algorithms in terms of localization accuracy. The results indicate that both NLLS and Brute-Force algorithms outperform the Linear LS algorithm, providing more precise location estimates with both. Furthermore, the performance of NLLS and Brute-Force algorithms is found to be comparable, with both achieving a high level of accuracy in positioning. Notably, with configurations of 4 and 3 gNBs, compared with the case of linear algorithms, the localization error is reduced to 1.3 meters (with a remarkable 65.8% improvement) and 1.7 meters (with a significant 55.3% improvement) in 80 percent of the cases, respectively.

The superior performance of NLLS and Brute-Force over Linear LS can be attributed to their ability to accommodate real-world signal propagation complexities better. These algorithms effectively average the errors of individual measurements, leading to more accurate position estimations.

Our experimentation highlights the effectiveness of NLLS and Brute-Force algorithms in achieving high-precision localization in a 5G network, with both methods showing considerable advantages over the simpler Linear LS approach. This analysis underscores the importance of algorithm selection in designing robust and accurate localization systems for wireless communications.

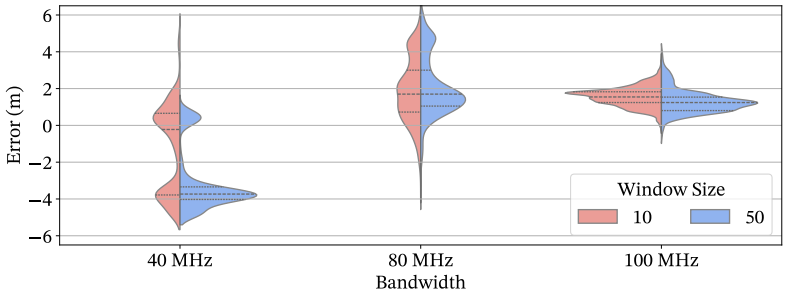


Figure 6.10: Distribution of error between measured and theoretical TDOAs for gNBs 2/3 at different bandwidths and window sizes.

To show the influence of bandwidth configuration on location estimation accuracy, in Fig. 6.10, we show the evolution of the error between the measured TDOA and the theoretical TDOA for UE positions 0, 4 and 5 and gNBs 2-3 by varying the bandwidth. When we increase the bandwidth, the improvements are two-fold: on the one hand, the estimations are closer to the theoretical value; on the other, the estimations are packed more densely towards the mean of the distribution, confirming the theoretical principles discussed in Sec. 6.4.4.

6.5.6 LOS vs NLOS scenarios

This paragraph provides an overview of what would happen in situations where NLOS was present. There could be several scenarios if our system were affected by NLOS. In the first scenario, one or more paths become blocked, and no PRS is received at the UE. If only one is blocked, the localization performance would revert to the 3 gNB scenario. If 2 or more PRS are blocked, localization would not be possible.

A second possible scenario is that the PRS reached the UE by another path, thus adding a delay to that particular TOA measurement. Since the OAI software is designed to select the highest correlation peak, the UE might latch onto a path different from the LOS one, thus reducing the performance of the localization algorithm. To simulate this with the 4 gNB scenario, we artificially introduce different delays in the TOA belonging to one of the gNBs and show the results in Fig. 6.11.

The TOA delay was increased in our simulations by introducing artificial delays to the TOA that were measured in the LOS/direct signal path. Specifically, we modeled a situation where the signal takes an indirect path to reach the UE, similar to what would happen with a single wall bounce. For instance, a signal reflecting off a surface at a 30° angle would incur a different delay compared to one reflecting at a 40° angle. These artificial delays emulate the effect of the signal taking a longer, indirect route, mimicking real-world NLOS conditions. In the end, since we are dealing with samples, the NLOS effect can be collapsed into a sample delay, simplifying the analysis and reducing the complexity of our simulations.

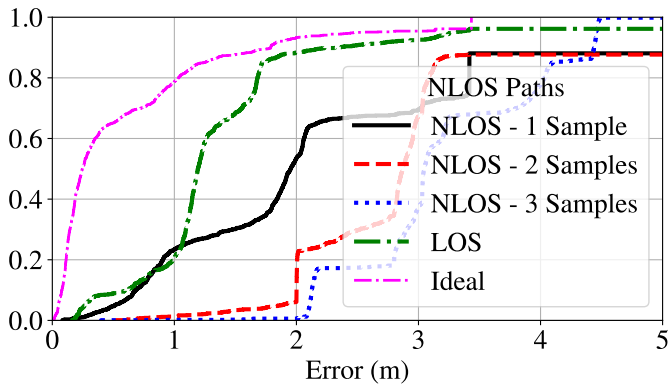


Figure 6.11: Empirical cumulative distribution function of the error when adding different delays to one of the TOA. Bandwidth is 100 MHz and no oversampling is performed.

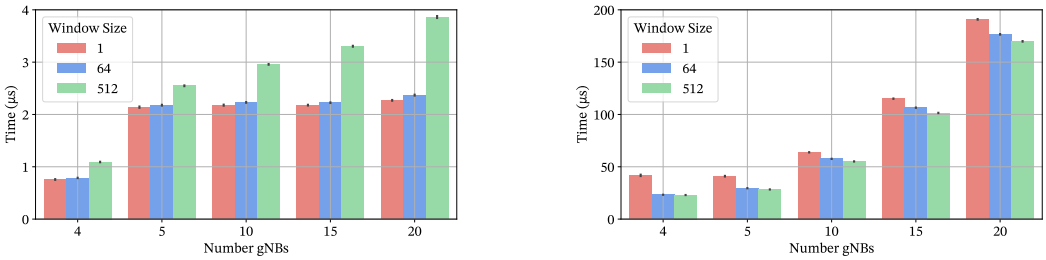
As expected, the performance degradation is notable even with 1 sample delay. Currently, the OAI software does not provide reliable tools to detect whether the received PRS comes from an LOS or NLOS path, so performance would be severely degraded.

6.5.7 Computational cost and Time to First Fix

This subsection evaluates the computational aspects of positioning in 5G networks, particularly focusing on the TTFF as defined by 3GPP technical specifications [133], [139]. In the most restrictive positioning cases, the enhanced positioning use case, the TTFF must be provided with a 90% horizontal error of less than 1 meter in less than 1 second. Given enough bandwidth and oversampling levels, we have shown that it is feasible to reach the accuracy goal. In this last section, we evaluate the time it would take to obtain the TTFF.

To assess the time required to obtain the position estimates with different window sizes, we developed a C++ TDOA library. We then generate 10k experiments and vary the number of gNBs from 4 to 20. We test this approach on a Macbook Pro M1 Pro with 10 cores. The results, illustrated in Fig. 6.12b, show the computation time for position estimates when varying the window size. Remarkably, the computation time remains relatively minimal, averaging around $\sim$200 μs even with 20 gNBs, which is practically negligible for most scenarios. Notably, when employing Linear least squares, this computation time further reduces to approximately $\sim$2 μs for up to 20 gNBs, as shown in Fig. 6.12a. This suggests that averaging more TOA measurements for enhanced accuracy, as seen in Fig. 6.5, incurs minimal time cost, except for the time it takes to accumulate enough data to fill the averaging window.

From these results, we can deduce that even for large window sizes, the computational time is negligible and could be suitable for most localization use cases specified in [139]. In our setup, the PRS was sent every 5 ms, enabling window sizes up to 128 and still meeting the requirements for a TTFF of less than 1 second. In other deployments, the PRS transmission frequency might vary, and our study indicates that the critical factor in



(a) Linear method

(b) Non-Linear method

Figure 6.12: Time taken to estimate position using (6.12a) Linear and (6.12b) Non-Linear methods when varying the window size and the number of gNBs.

determining TTFF is the PRS frequency, not the computational time for position estimation.

6.6 Conclusion

In this chapter, we thoroughly explored the key components and the evolving landscape of 5G positioning, leveraging the capabilities of open-source platforms supported by SDR technology. Our investigation delved into various aspects of 5G positioning, including the application and evaluation of different location algorithms, signal processing techniques such as oversampling and moving averages, practical impairments, and the implications of varying network configuration (i.e., number of base stations and bandwidth).

Overcoming the inherent challenges associated with open-source solutions and SDR configurations, this research has demonstrated the versatility and robustness of such an approach in dissecting the multifaceted nature of 5G positioning in a real-world environment and meeting the positioning accuracy and stringent TTFF requirements outlined by industry standards. Experimenting with different configurations and methodologies yielded valuable insights, especially in understanding how the interplay between network geometry, algorithmic choices, and signal processing strategies influence and enhance localization accuracy within 5G networks.

Our findings reveal that employing the NLLS or Brute-Force localization algorithm, combined with an averaging window size of 50 and an oversampling factor of 4, enables us to achieve a localization error as low as 0.7 meters in 80% of cases. Additionally, we have attained a TTFF of approximately 25 microseconds, further illustrating the efficacy of our approach. It is important to highlight that while the localization performances and operational efficiency achieved in our testbed are intriguing, the primary objective of this research was not to showcase the ultimate capabilities of the testbed in 5G localization. Instead, the chapter aimed to demonstrate the testbed's potential as a valuable tool for the research community and industry. This platform enables the experimentation and validation of algorithms and research findings through real over-the-air transmitted 5G posi-

tioning signals. In doing so, it accounts for standardized localization procedures and the complexities of real-world operating conditions. This research emphasizes the potential of leveraging open-source, SDR-based platforms as effective practical tools for innovation and exploration, thus paving the way for continued advancements in 5G positioning technology.

PART V

LAST REMARKS

7 Conclusions

In this thesis, we prsent how flexible SDR devices are, allowing localization and monitoring in many scenarios. Their ability to be easily configured in runtime allows for better RF spectrum monitoring at all levels and with any type of hardware quality. We identify several research questions that can be answered by taking advantage of such dynamic devices.

In Part II we explore how low-cost SDR devices enable diverse positioning applications. In Chapter 3 we show that low-cost receivers can be used to build a transmitter localization network that operates on the city scale and achieves notable accuracy. Although these devices are prone to hardware impairments, we demonstrate that it is feasible to estimate and correct these defects to achieve the desired results. In Chapter 4 we build a mixed network with sensors collecting metadata from aircraft broadcasts, which are then publicly made available to other receivers. Again, considering hardware defects and the geometry of the selected aircraft, we can achieve notable positioning accuracy, even for sensors with low-frequency precision.

In Part III we utilize similar low-cost sensors in RF signal analytics. In this case, in Chapter 5 we design a low-cost receiver network that is capable of collecting tire pressure transmissions at relatively large distances given the low power nature of these signals. In the backend, we design an algorithm that can robustly match all 4 tires from the same car to enable robust car tracking. By collecting data for several months, we can extract driver patterns, such as workplace habits or anomalous behaviors, when a transmission appears on unexpected days or hours. With this work, we also want to highlight the need for more privacy aware protocols for tires, as malicious actors could use these transmissions to track individuals without their consent.

In the last part, Part IV, we present a 5G localization framework by building a deployment using a popular open-source 5G communications framework. We perform modifications to allow the extraction of timing information to perform TDOA localization. We do an extensive evaluation of the positioning accuracy by modifying the number of transmitters, the positions, LOS/NLOS conditions, different optimization algorithms and several other parameter smoothening techniques. We also do extensive computational per-

formance tests, and verify that our approach can work well within the timing constraints defined by the standards.

However, these research lines are not closed and several improvements could be proposed in each of the parts. An aspect that is not explored in Part II is merging the results of both chapters. Using aircraft signals, it is not possible to localize but also synchronize a sensor network without the need of GPS or any other signal of oppportunity. Another line of research concerns data bandwidth, specifically for Chapter 3. We rely on sending full IQ data to the backend, which utilizes a considerable amount of bandwidth. In future work, we could explore techniques to reduce these bandwidth requirements without losing the accurate timing information that can be achieved with raw data.

For Part III, our system so far has been passive and is deployed in a relatively small region. Future work could look at extending the sensor network all over a city and track more complex spatio-temporal patterns. With bigger networks, it could be possible to infer routes from drivers' homes to their workplaces and provide malicious sniffers with information into when someone is out of their house. Tire pressure sensors are also vulnerable to spoofed transmissions, so active attacks on these systems could be targeted at logistic companies to disrupt their operations.

Last, in Part IV we could extend our deployment in several forms. We could increase the size of the deployment and add more UEs as well as gNBs. We could add obstacles to recreate NLOS conditions and test more sophisticated approaches to block signals and evaluate the effect on positioning accuracy. Moreover, we could implement these algorithms so that they are integrated within OAI and can perform localization in real-time.

In summary, in this thesis we have presented how thanks to the flexibility and variety of SDR devices we can build many useful applications. Low-cost receivers can be used to create spectrum sensing networks that can perform tasks that usually require high precision, like localization, but those same sensors can also be used to monitor specific signals at city levels and track them over long periods of time. With higher-grade SDR receivers we can build complex 5G networks and then use them to perform tasks like localization with remarkable accuracy and low computational cost.

Bibliography

[1] Y. Lizarribar, R. Calvo-Palomino, A. Scalingi, G. Santaromita, G. Bovet, and D. Giustiniano, "ORAN-Sense: Localizing Non-cooperative Transmitters with Spectrum Sensing and 5G O-RAN," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, May 2024, pp. 1870–1879. DOI: 10.1109/INFOCOM52122.2024.10621298. (visited on 09/15/2024).

[2] Y. Lizarribar, D. Giustiniano, G. Bovet, and V. Lenders, "SkyPos: Real-World Evaluation of Self-Positioning With Aircraft Signals for IoT Devices," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 1, pp. 134–145, Jan. 2024. DOI: 10.1109/JSAC.2023.3322829. (visited on 06/11/2024).

[3] Y. Lizarribar *et al.*, "Can't Hide Your Stride: Identifying driver patterns from a network of low-cost spectrum Recievers," in *Under Submission*, 2024.

[4] I. Palamà *et al.*, "From Experiments to Insights: A Journey in 5G New Radio Localization," in *2023 21st Mediterranean Communication and Computer Networking Conference (MedComNet)*, Jun. 2023, pp. 74–82. DOI: 10.1109/MedComNet58619.2023.10168856.

[5] I. Palamà *et al.*, "5G positioning with software-defined radios," *Computer Networks*, p. 110 595, Jun. 2024. DOI: 10.1016/j.comnet.2024.110595. (visited on 06/11/2024).

[6] Y. Lizarribar, "Signal transmitter localization using low-cost SDR receivers," M.S. thesis, Universidad Carlos III de Madrid, 2020.

[7] B. Reynders *et al.*, "SkySense: Terrestrial and aerial spectrum use analysed using lightweight sensing technology with weather balloons," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '20, New York, NY, USA: Association for Computing Machinery, Jun. 2020, pp. 352–363. DOI: 10.1145/3386901.3389026. (visited on 07/20/2023).

[8] M. Rea, D. Giustiniano, P. J. Mateo, Y. Lizarribar, and J. Widmer, "Beam searching for mmWave networks with sub-6GHz WiFi and inertial sensors inputs: An experimental study," *Computer Networks*, vol. 198, p. 108 344, Oct. 2021. DOI: 10.1016/j.comnet.2021.108344. (visited on 08/10/2023).

[9] International Telecommunication Union (ITU), *Radio regulations - article 1: Terms and definitions*, https://www.itu.int/hub/publication/r-reg-rr-2024/, 2024.

[10] Ericsson, *Ericsson Mobility Report November 2024*, https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/june-2024, 2024. (visited on 11/19/2024).

[11] NTIA, *United States Frequency Allocation Chart | National Telecommunications and Information Administration*, https://www.ntia.gov/page/united-states-frequency-allocation-chart, 2024. (visited on 06/09/2024).

[12] G. Lee *et al.*, "This is Your President Speaking: Spoofing Alerts in 4G LTE Networks," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '19, New York, NY, USA: Association for Computing Machinery, Jun. 2019, pp. 404–416. DOI: 10.1145/3307334.3326082. (visited on 06/09/2024).

[13] D. Moser, P. Leu, V. Lenders, A. Ranganathan, F. Ricciato, and S. Capkun, "Investigation of multi-device location spoofing attacks on air traffic control and possible countermeasures," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '16, New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 375–386. DOI: 10.1145/2973750.2973763. (visited on 01/12/2023).

[14] I. ComSoc, *Software-defined radio | IEEE Communications Society*, https://www.comsoc.org/node/19691. (visited on 09/14/2024).

[15] S. Rajendran *et al.*, "Electrosense: Open and Big Spectrum Data," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 210–217, Jan. 2018. DOI: 10.1109/MCOM.2017.1700200.

[16] R. Calvo-Palomino *et al.*, "Electrosense+: Crowdsourcing radio spectrum decoding using IoT receivers," *Comput. Networks*, vol. 174, p. 107 231, 2020. DOI: 10.1016/j.comnet.2020.107231.

[17] Software Radio Systems, *srsRAN 4G: Open source SDR 4G software suite from Software Radio Systems (SRS)*, srsRAN, Dec. 2023. (visited on 12/07/2023).

[18] Software Radio Systems, *srsRAN: Open source O-RAN 5G CU/DU solution from Software Radio Systems (SRS)*, srsRAN, Dec. 2023. (visited on 12/07/2023).

[19] Open Networking Foundation (ONF), *SD-RAN*, 2023.

[20] Open Networking Foundation (ONF), *SD-CORE*, 2023.

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

[22] J. Wei and C. Yu, "Performance analysis of TDoA based localization using SDRs," in *2013 Australian Control Conference*, Fremantle, WA, Australia: IEEE, Nov. 2013, pp. 91–92. DOI: 10.1109/AUCC.2013.6697253.

[23] X. Huang, K. Yan, H.-C. Wu, and Y. Wu, "Unmanned Aerial Vehicle Hub Detection Using Software-Defined Radio," in *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jeju, South Korea: IEEE, Jun. 2019, pp. 1–6. DOI: 10.1109/BMSB47279.2019.8971851.

[24] C. Zhan, M. Ghaderibaneh, P. Sahu, and H. Gupta, "DeepMTL Pro: Deep Learning Based Multiple Transmitter Localization and Power Estimation," *Pervasive and Mobile Computing*, vol. 82, p. 101 582, Jun. 2022. DOI: 10.1016/j.pmcj.2022.101582.

[25] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, Jun. 2005. DOI: 10.1109/MSP.2005.1458284.

[26] A. Zanella, "Best practice in RSS measurements and ranging," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, pp. 2662–2686, Apr. 2016. DOI: 10.1109/COMST.2016.2553452.

[27] H. Nurminen, M. Dashti, and R. Piché, "A Survey on Wireless Transmitter Localization Using Signal Strength Measurements," *Wireless Communications and Mobile Computing*, vol. 2017, e2569645, Feb. 2017. DOI: 10.1155/2017/2569645. (visited on 02/20/2023).

[28] M. Khaledi *et al.*, "Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, vol. Part F1312, pp. 235–247, Oct. 2017. DOI: 10.1145/3117811.3117845.

[29] C. Y. Kim, D. Song, Y. Xu, J. Yi, and X. Wu, "Cooperative search of multiple unknown transient radio sources using multiple paired mobile robots," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1161–1173, Jul. 2014. DOI: 10.1109/TRO.2014.2333097.

[30] C. Perkins *et al.*, "Distance sensing for mini-robots: RSSI vs. TDOA," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, Rio de Janeiro, Brazil: IEEE, May 2011, pp. 1984–1987. DOI: 10.1109/ISCAS.2011.5937980.

[31] M. Hartmann, T. Nowak, L. Patino-Studencki, J. Robert, A. Heuberger, and J. Thielecke, "A Low-Cost RSSI-Based Localization System for Wildlife Tracking," *IOP Conference Series: Materials Science and Engineering*, vol. 120, no. 1, Mar. 2016. DOI: 10.1088/1757-899X/120/1/012004. (visited on 02/20/2023).

[32] N. A. Azmi, S. Samsul, Y. Yamada, M. F. M. Yakub, M. I. M. Ismail, and R. A. Dziyauddin, "A Survey of Localization using RSSI and TDoA Techniques in Wireless Sensor Network: System Architecture," *2018 2nd International Conference on Telematics and Future Generation Networks, TAFGEN 2018*, pp. 131–136, Jul. 2018. DOI: 10.1109/TAFGEN.2018.8580464.

[33] A. Bhattacharya, C. Zhan, A. Maji, H. Gupta, S. R. Das, and P. M. Djuric, "Selection of Sensors for Efficient Transmitter Localization," *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, pp. 107–119, Feb. 2022. DOI: 10.1109/TNET.2021.3104000.

[34] Y. Wang and K. C. Ho, "Unified Near-Field and Far-Field Localization for AOA and Hybrid AOA-TDOA Positionings," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 1242–1254, Feb. 2018. DOI: 10.1109/TWC.2017.2777457.

[35] J. Yin, Q. Wan, S. Yang, and K. C. Ho, "A simple and accurate TDOA-AOA localization method using two stations," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 144–148, Dec. 2016. DOI: 10.1109/LSP.2015.2505138.

[36] D. Kim, S. Jeong, K. E. Lee, and J.-H. Kang, "Performance analysis of AoA-based localization with software defined radio," in *Ignss 2015*, International Global Navigation Satellite Systems, 2015.

[37] rtlsdrblog, *Rtlsdrblog/rtl-sdr-blog*, Jul. 2023. (visited on 07/26/2023).

[38] R. Kaune, "Accuracy studies for TDOA and TOA localization," in *2012 15th International Conference on Information Fusion*, Singapore: IEEE, Jul. 2012, pp. 408–415.

[39] M. Rea, A. Fakhreddine, D. Giustiniano, and V. Lenders, "Filtering Noisy 802.11 Time-of-Flight Ranging Measurements From Commoditized WiFi Radios," *IEEE ACM Transactions on Networking*, vol. 25, no. 4, pp. 2514–2527, Aug. 2017. DOI: 10.1109/TNET.2017.2700430.

[40] M. Eichelberger, K. Luchsinger, S. Tanner, and R. Wattenhofer, "Indoor Localization with Aircraft Signals," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '17, New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1–14. DOI: 10.1145/3131672.3131698. (visited on 02/20/2023).

[41] Y. Wang and Y. Wu, "An efficient semidefinite relaxation algorithm for moving source localization using TDOA and FDOA measurements," *IEEE Communications Letters*, vol. 21, no. 1, pp. 80–83, Oct. 2017. DOI: 10.1109/LCOMM.2016.2614936.

[42] F. Ma, F. Guo, and L. Yang, "Low-complexity TDOA and FDOA localization: A compromise between two-step and DPD methods," *Digital Signal Processing: A Review Journal*, vol. 96, p. 102 600, Jan. 2020. DOI: 10.1016/j.dsp.2019.102600.

[43] M. Gucciardo, I. Tinnirello, G. M. Dell'Aera, and M. Caretti, "A flexible 4G/5G control platform for fingerprint-based indoor localization," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, Sep. 2019, pp. 744–749.

[44] B. T. Fang, "Simple solutions for hyperbolic and related position fixes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 5, pp. 748–753, Sep. 1990.

[45] H. P. Gavin, "The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems," *Department of Civil and Environmental Engineering, Duke University*, vol. 19, no. Department of Civil and Environmental Engineering, Duke University, 2019.

[46] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., 1993.

[47] H. Sallouha, M. M. Azari, A. Chiumento, and S. Pollin, "Aerial Anchors Positioning for Reliable RSS-Based Outdoor Localization in Urban Environments," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 376–379, Jun. 2018. DOI: 10.1109/LWC.2017.2778723.

[48] L. Ceci, *Mobile Internet Usage Worldwide-Statistics & Facts*, https://www.statista-.com/topics/779/mobile-internet/, 2023. (visited on 07/13/2023).

[49] S. Roy *et al.*, "CityScape: A Metro-Area Spectrum Observatory," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, Vancouver, BC, Canada: IEEE, Jul. 2017, pp. 1–9. DOI: 10.1109/ICCCN.2017.8038427. (visited on 12/08/2023).

[50] F. Al-Turjman, E. Ever, and H. Zahmatkesh, "Small Cells in the Forthcoming 5G/IoT: Traffic Modelling and Deployment Overview," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 28–65, Aug. 2019. DOI: 10.1109/COMST.2018.2864779.

[51] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead," *Computer Networks*, vol. 182, p. 107 516, Dec. 2020. DOI: 10.1016/j.comnet.2020.107516. (visited on 07/24/2023).

[52] J. Breen *et al.*, "POWDER: Platform for Open Wireless Data-driven Experimental Research," in *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, ser. WiNTECH'20, New York, NY, USA: Association for Computing Machinery, Sep. 2020, pp. 17–24. DOI: 10.1145/3411276.3412204. (visited on 07/24/2023).

[53] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023. DOI: 10.1109/COMST.2023.3239220.

[54] *O-RAN ALLIANCE e.V*, https://www.o-ran.org/. (visited on 07/24/2023).

[55] S. Bartoletti *et al.*, "Location-Based Analytics in 5G and Beyond," *IEEE Communications Magazine*, vol. 59, no. 7, pp. 38–43, Jul. 2021. DOI: 10.1109/MCOM.001.2001096.

[56] A. Bourdoux *et al.*, *6G White Paper on Localization and Sensing*, Jun. 2020. DOI: 10.48550/arXiv.2006.01779. arXiv: 2006.01779 [cs, eess]. (visited on 07/23/2023).

[57] J. Schmitz, M. Hernandez, and R. Mathar, "Demonstration Abstract: Real-Time Indoor Localization with TDOA and Distributed Software Defined Radio," in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Vienna, Austria: IEEE, Apr. 2016, pp. 1–2. DOI: 10.1109/IPSN.2016.7460671. (visited on 06/16/2022).

[58] F. ture, T. Morong, P. Ková, and P. Purier, "High Performance SDR for Monitoring System for GNSS Jamming Localization," in *2019 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco: IEEE, Oct. 2019, pp. 1–5. DOI: 10.1109/WINCOM47513.2019.8942521.

[59] H. Sallouha, A. Chiumento, and S. Pollin, "Aerial Vehicles Tracking Using Non-coherent Crowdsourced Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 780–10 791, Oct. 2021. DOI: 10.1109/TVT.2021.3103315. arXiv: 2108.00922 [cs, eess]. (visited on 06/02/2022).

[60] R. Schreiber and J. Bajer, "Time difference measurement algorithm for TDOA positioning system using RTL-SDR," in *2017 International Conference on Military Technologies (ICMT)*, Brno, Czech Republic: IEEE, May 2017, pp. 608–612. DOI: 10.1109/MILTECHS.2017.7988829.

[61] Y. Zeng, R. Calvo-Palomino, D. Giustiniano, G. Bovet, and S. Banerjee, "Adaptive Uplink Data Compression in Spectrum Crowdsensing Systems," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 1–15, Jan. 2023. DOI: 10.1109/TNET.2023.3239378.

[62] J. Schmitz, R. Mathar, and D. Dorsch, "Compressed time difference of arrival based emitter localization," in *2015 3rd International Workshop on Compressed Sensing Theory and Its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*, Pisa: IEEE, Jun. 2015, pp. 263–267. DOI: 10.1109/CoSeRa.2015.7330305. (visited on 06/16/2022).

[63] A. Canals, P. Josephy, S. Tanner, and R. Wattenhofer, "Robust indoor localization with ADS-B," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '21, New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 505–516. DOI: 10.1145/3447993.3483257. (visited on 02/20/2023).

[64] 3GPP, "Study on new radio access technology: Radio access architecture and interfaces," 38.801, Apr. 2017.

[65] O-RAN Alliance, *WG1: Use cases and overall architecture workgrou*, Specification, 2023.

[66] O-RAN Alliance, *WG2: Non-real-time RAN intelligent controller and A1 interface workgroup*, Specification, 2023.

[67] O-RAN Alliance, *WG10: OAM for O-RAN*, Specification, 2023.

[68] O-RAN Alliance, *WG3: Near-real-time RIC and E2 interface workgroup*, Specification, 2023.

[69] R. Calvo-Palomino, F. Ricciato, D. Giustiniano, and V. Lenders, "LTESS-track: A Precise and Fast Frequency Offset Estimation for low-cost SDR Platforms," in *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, ser. WiNTECH '17, New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 51–58. DOI: 10.1145/3131473.3131481. (visited on 02/20/2023).

[70] A. Scalingi, D. Giustiniano, R. Calvo-Palomino, N. Apostolakis, and G. Bovet, "A Framework for Wireless Technology Classification using Crowdsensing Platforms," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, New York, USA: IEEE, May 2023, pp. 1–10. DOI: 10.1109/INFOCOM53939.2023.10228867. (visited on 11/28/2023).

[71] E. R. Brand a National Instruments, *Board Mounted GPSDO (OCXO)*, https://www.ettus.com/all-products/gpsdo-mini/. (visited on 11/19/2024).

[72] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11, New York, NY, USA: Association for Computing Machinery, Oct. 2011, pp. 75–86. DOI: 10.1145/2046707.2046719. (visited on 01/12/2023).

[73] J. Noh *et al.*, "Tractor Beam: Safe-hijacking of Consumer Drones with Adaptive GPS Spoofing," *ACM Transactions on Privacy and Security*, vol. 22, no. 2, 12:1–12:26, Apr. 2019. DOI: 10.1145/3309735. (visited on 01/12/2023).

[74] S. Bartoletti *et al.*, "Positioning and Sensing for Vehicular Safety Applications in 5G and Beyond," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 15–21, Nov. 2021. DOI: 10.1109/MCOM.011.2100339.

[75] R. Keating, M. Säily, J. Hulkkonen, and J. Karjalainen, "Overview of Positioning in 5G New Radio," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, Aug. 2019, pp. 320–324. DOI: 10.1109/ISWCS.2019.8877160.

[76] Q. Liu *et al.*, "Management of Positioning Functions in Cellular Networks for Time-Sensitive Transportation Applications," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, Jan. 2023. DOI: 10.1109/TITS.2023.3234532.

[77] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up OpenSky: A large-scale ADS-B sensor network for research," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, Apr. 2014, pp. 83–94. DOI: 10.1109/IPSN.2014.6846743.

[78] M. Eichelberger, K. Luchsinger, S. Tanner, and R. Wattenhofer, "Indoor Localization with Aircraft Signals," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '17, New York, NY, USA: Association for Computing Machinery, Nov. 2017, pp. 1–14. DOI: 10.1145/3131672.3131698. (visited on 01/12/2023).

[79] S. Zhu, X. Zheng, L. Liu, and H. Ma, "AirSync: Time Synchronization for Large-Scale IoT Networks Using Aircraft Signals," *IEEE Transactions on Mobile Computing*, vol. 22, no. 1, pp. 69–83, Jan. 2023. DOI: 10.1109/TMC.2021.3070644.

[80] S. Sciancalepore, S. Alhazbi, and R. Di Pietro, "Receivers location privacy in avionic crowdsourced networks: Issues and countermeasures," *Journal of Network and Computer Applications*, vol. 174, p. 102 892, Jan. 2021. DOI: 10.1016/j.jnca.2020.102892. (visited on 01/12/2023).

[81] J. Kouba and P. Héroux, "Precise Point Positioning Using IGS Orbit and Clock Products," *GPS Solutions*, vol. 5, no. 2, pp. 12–28, Oct. 2001. DOI: 10.1007/PL00012883. (visited on 01/12/2023).

[82] H. Sallouha, M. M. Azari, and S. Pollin, *Energy-Constrained UAV Trajectory Design for Ground Node Localization*, Jun. 2018. arXiv: 1806.02055 [cs]. (visited on 03/23/2023).

[83] Q. Liu, R. Liu, Z. Wang, and J. S. Thompson, "UAV Swarm-Enabled Localization in Isolated Region: A Rigidity-Constrained Deployment Perspective," *IEEE Wireless Communications Letters*, vol. 10, no. 9, pp. 2032–2036, Sep. 2021. DOI: 10.1109/LWC.2021.3091215.

[84] Q. Liu, R. Liu, Z. Wang, L. Han, and J. S. Thompson, "A V2X-Integrated Positioning Methodology in Ultradense Networks," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 17 014–17 028, Dec. 2021. DOI: 10.1109/JIOT.2021.3075532.

[85] R. Calvo-Palomino, F. Ricciato, B. Repas, D. Giustiniano, and V. Lenders, "Nanosecond-Precision Time-of-Arrival Estimation for Aircraft Signals with Low-Cost SDR Receivers," in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2018, pp. 272–277. DOI: 10.1109/IPSN.2018.00055.

[86] S. Sanfilippo, *Dump1090 README*, May 2023. (visited on 05/30/2023).

[87] M. Schäfer, M. Strohmeier, M. Leonardi, and V. Lenders, "LocaRDS: A Localization Reference Data Set," *Sensors*, vol. 21, no. 16, p. 5516, Jan. 2021. DOI: 10.3390/s21165516. (visited on 01/12/2023).

[88] M. Schafer *et al.*, "OpenSky report 2016: Facts and figures on SSR mode S and ADS-B usage," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sep. 2016, pp. 1–9. DOI: 10.1109/DASC.2016.7778030.

[89] J. Sun, *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*. TU Delft OPEN, 2021. (visited on 01/12/2023).

[90] C. Fernández-Prades *et al.*, *GNSS-SDR*, Apr. 2022. (visited on 02/14/2023).

[91] R. Kaune, C. Steffes, S. Rau, W. Konle, and J. Pagel, "Wide area multilateration using ADS-B transponder signals," in *2012 15th International Conference on Information Fusion*, Jul. 2012, pp. 727–734.

[92] G. Galati, M. Leonardi, I. A. Mantilla-Gaviria, and M. Tosti, "Lower bounds of accuracy for enhanced mode-S distributed sensor networks," *IET Radar, Sonar & Navigation*, vol. 6, no. 3, pp. 190–201, Mar. 2012. DOI: 10.1049/iet-rsn.2011.0197. (visited on 01/12/2023).

[93] I. A. Mantilla-Gaviria, M. Leonardi, G. Galati, and J. V. Balbastre-Tejedor, "Localization algorithms for multilateration (MLAT) systems in airport surface surveillance," *Signal, Image and Video Processing*, vol. 9, no. 7, pp. 1549–1558, Oct. 2015. DOI: 10.1007/s11760-013-0608-1. (visited on 01/12/2023).

[94] C. A. Floudas and P. M. Pardalos, *Encyclopedia of Optimization*. Springer Science & Business Media, 2008.

[95] S. Ruder, *An overview of gradient descent optimization algorithms*, Jun. 2017. arXiv: 1609.04747 [cs]. (visited on 02/10/2023).

[96] R. Santerre, A. Geiger, and S. Banville, "Geometry of GPS dilution of precision: Revisited," *GPS Solutions*, vol. 21, no. 4, pp. 1747–1763, Oct. 2017. DOI: 10.1007/s10291-017-0649-y. (visited on 01/09/2023).

[97] J. Bard and F. Ham, "Time difference of arrival dilution of precision and applications," *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 521–523, Feb. 1999. DOI: 10.1109/78.740135.

[98] M. Wang, Z. Chen, Z. Zhou, J. Fu, and H. Qiu, "Analysis of the Applicability of Dilution of Precision in the Base Station Configuration Optimization of Ultrawideband Indoor TDOA Positioning System," *IEEE Access*, vol. 8, pp. 225 076–225 087, 2020. DOI: 10.1109/ACCESS.2020.3045189.

[99] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics - Theory and Methods*, vol. 6, no. 9, pp. 813–827, Jan. 1977. DOI: 10.1080/03610927708827533. (visited on 01/12/2023).

[100] Radarcape, *ADS-B Receiver MLAT, Radarcape, ASTERIX, Dataports JSON, Binary*, http://radarcape.com/. (visited on 05/25/2023).

[101] R.-S. blog, *A Datasheet for our RTL-SDR.com V3 Dongles*, https://www.rtl-sdr.com/a-datasheet-for-our-rtl-sdr-com-v3-dongles/, Jun. 2017. (visited on 05/25/2023).

[102] *Flask-RESTful Flask-RESTful 0.3.8 documentation*, https://flask-restful.readthedocs.io/en/latest/. (visited on 01/12/2023).

[103] P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020. DOI: 10.1038/s41592-019-0686-2. (visited on 01/12/2023).

[104] *Redis*, https://redis.io/. (visited on 01/12/2023).

[105] *RediSearch*, https://redis.io/docs/stack/search/. (visited on 01/12/2023).

[106] F. Shang, B. Wang, F. Yan, and T. Li, "Multidevice False Data Injection Attack Models of ADS-B Multilateration Systems," *Security and Communication Networks*, vol. 2019, e8936784, Mar. 2019. DOI: 10.1155/2019/8936784. (visited on 01/12/2023).

[107] S. Velupillai and L. Guvenc, "Tire pressure monitoring [applications of control]," *IEEE Control systems magazine*, vol. 27, no. 6, pp. 22–25, 2007.

[108] J. Caban, P. Drodziel, D. Barta, and . Liák, "Vehicle tire pressure monitoring systems," *Diagnostyka*, vol. 15, no. 3, pp. 11–14, 2014.

[109] I. Rouf *et al.*, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *19th USENIX Security Symposium*, Washington, DC, USA: USENIX Association, 2010, pp. 323–338.

[110] K. Hacker, S. Graham, and S. Dunlap, "Vehicle Identification and Route Reconstruction via TPMS Data Leakage," in *Critical Infrastructure Protection XIII*, J. Staggs and S. Shenoi, Eds., ser. IFIP Advances in Information and Communication Technology, Cham: Springer International Publishing, 2019, pp. 123–136. DOI: 10.1007/978-3-030-34647-8_7.

[111] K. T. Sterne, J. M. Ernst, D. K. Kilcoyne, A. J. Michaels, and G. Moy, "Tire pressure monitoring system sensor radio frequency measurements for privacy concerns," *Transportation Research Record*, vol. 2643, no. 1, pp. 34–44, 2017.

[112] R. Calvo-Palomino, H. Cordobés, F. Ricciato, D. Giustiniano, and V. Lenders, "Collaborative wideband signal decoding using non-coherent receivers," in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, ser. IPSN '19, New York, NY, USA: Association for Computing Machinery, Apr. 2019, pp. 37–48. DOI: 10.1145/3302506.3310387. (visited on 11/17/2023).

[113] A. Kolodgie *et al.*, "Enhanced TPMS security through acceleration timed transmissions," in *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, Baltimore, MD, USA: IEEE, Oct. 2017, pp. 35–39. DOI: 10.1109/MILCOM.2017.8170841. (visited on 11/17/2023).

[114] D. K. Kilcoyne, S. Bendelac, J. M. Ernst, and A. J. Michaels, "Tire Pressure Monitoring System encryption to improve vehicular security," in *MILCOM 2016 - 2016 IEEE Military Communications Conference*, Baltimore, MD, USA: IEEE, Nov. 2016, pp. 1219–1224. DOI: 10.1109/MILCOM.2016.7795497. (visited on 11/17/2023).

[115] C. Solomon and B. Groza, "LiMon - Lightweight Authentication for Tire Pressure Monitoring Sensors," in *Security of Industrial Control Systems and Cyber Physical Systems*, A. Bécue, N. Cuppens-Boulahia, F. Cuppens, S. Katsikas, and C. Lambrinoudakis, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 95–111. DOI: 10.1007/978-3-319-40385-4_7.

[116] K. Daimi and M. Saed, "Securing tire pressure monitoring system," in *Proceedings of the 2018 the 14th Advanced International Conference on Telecommunications Conference, Barcelona, Spain*, Barcelona, Spain: IARIA, Jul. 2018, pp. 22–26.

[117] M. Vaszary, A. Slovacek, Y. Zhuang, and S.-Y. Chang, "Securing Tire Pressure Monitoring System for Vehicular Privacy," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA: IEEE, Jan. 2021, pp. 1–6. DOI: 10.1109/CCNC49032.2021.9369576. (visited on 11/17/2023).

[118] E. United Nations and S. Council, *ECE/TRANS/WP.29/2020/79: Proposal for a new UN Regulation on uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system*, 2020.

[119] Unece, *UN Regulations on Cybersecurity and Software Updates to pave the way for mass roll out of connected vehicles*, 2020.

[120] E. Union, *Regulation (EU) 2019/2144 of the European Parliament and of the Council on type-approval requirements for motor vehicles and their trailers, and systems, components and separate technical units intended for such vehicles*, 2019.

[121] Autel, *MaxiTPMS TS501 | Autel*, https://www.maxitpms.com/product/582.jhtml, 2023. (visited on 11/06/2023).

[122] C. R. Bowen and M. H. Arafa, "Energy Harvesting Technologies for Tire Pressure Monitoring Systems," *Advanced Energy Materials*, vol. 5, no. 7, p. 1 401 787, 2015. DOI: 10.1002/aenm.201401787. (visited on 12/11/2023).

[123] J. Schulz and M. Junghans, "Deployment of Tire Pressure Monitoring Systems for Traffic Monitoring and Safety Purposes," in *ITSWC 2011*, Orlando (FL), USA: ITSWC, Oct. 2011, pp. 1–10. (visited on 11/17/2023).

[124] R. Suender, G. Prokop, and T. Roscher, "Comparative Analysis of Tire Evaluation Methods for an indirect Tire Pressure Monitoring System (iTPMS)," *SAE International Journal of Passenger Cars - Mechanical Systems*, vol. 8, no. 2015-01-1519, pp. 110–118, Apr. 2015. DOI: 10.4271/2015-01-1519. (visited on 11/06/2023).

[125] FCC, *OET List Exhibits Report*, https://apps.fcc.gov/oetcf/eas/reports/ViewExhibitReport.cfm?mode=Exhibits&RequestTimeout=500&calledFromFrame=N&application_id=FBrst07d7dfvg4f9lmrR4A%3D%3D&fcc_id=MRXFG2R4MA, 2011. (visited on 11/25/2023).

[126] B. Larsson, *Rtl_433: Program to decode radio transmissions from devices on the ISM bands (and other frequencies)*, 2022.

[127] A. E. Kubba and K. Jiang, "A comprehensive study on technologies of tyre monitoring systems and possible energy solutions," *Sensors*, vol. 14, no. 6, pp. 10 306–10 345, 2014.

[128] J.-C. Yoo and T. H. Han, "Fast Normalized Cross-Correlation," *Circuits, Systems and Signal Processing*, vol. 28, no. 6, pp. 819–843, Dec. 2009. DOI: 10.1007/s00034-009-9130-7. (visited on 11/25/2023).

[129] C. Caraffi, T. Vojí, J. Trefný, J. ochman, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Anchorage, AK, USA: IEEE, Sep. 2012, pp. 975–982. DOI: 10.1109/ITSC.2012.6338748. (visited on 11/26/2023).

[130] S. Jung, Y. Kim, and E. Hwang, "Real-time car tracking system based on surveillance videos," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, p. 133, Nov. 2018. DOI: 10.1186/s13640-018-0374-7. (visited on 11/26/2023).

[131] 3GPP, "Study on NR positioning support," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 33.855, Jan. 2023.

[132] 3GPP, "Study on NR positioning enhancements," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.857, Mar. 2021.

[133] 3GPP, "Service requirements for the 5G system," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 22.261, Sep. 2023.

[134] 3GPP, "Service requirements for cyber-physical control applications in vertical domains," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 22.104, Sep. 2023.

[135] 3GPP, "Architectural enhancements to support ranging based services and sidelink positioning," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.586, Sep. 2023.

[136] 3GPP, "Study on expanded and improved NR positioning," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.859, Jan. 2023.

[137] 3GPP, "Study on integrated sensing and communication," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 22.837, Sep. 2023.

[138] 3GPP, "NG radio access network (NG-RAN); stage 2 functional specification of user equipment (UE) positioning in NG-RAN," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.305, Sep. 2023.

[139] 3GPP, "Study on positioning use cases," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 22.872, Sep. 2023.

[140] M. Cui *et al.*, "A novel iterative positioning method based on difference RSS model with 5G field experiments," ser. IEEE Sensors Journal, vol. 24, 2024, pp. 5466–5475. DOI: 10.1109/JSEN.2023.3277510.

[141] Z. Li, F. Jiang, H. Wymeersch, and F. Wen, "An iterative 5G positioning and synchronization algorithm in NLOS environments with multi-bounce paths," ser. IEEE Wireless Communications Letters, vol. 12, 2023, pp. 804–808. DOI: 10.1109/LWC.2023.3244575.

[142] P. A. D. N. Jayawardana, H. Obaid, T. Yesilyurt, B. Tan, and E. S. Lohan, "Machine-learning-based LOS detection for 5G signals with applications in airport environments," ser. Sensors, vol. 23, 2023. DOI: 10.3390/s23031470.

[143] X. Jia, P. Liu, S. Liu, X. Li, and W. Qi, "Link-level simulator for 5G localization," in *2022 IEEE Globecom Workshops (GC Wkshps)*, Jan. 2022, pp. 401–406. DOI: 10.1109/GCWkshps56602.2022.10008769.

[144] S. Dwivedi *et al.*, "Positioning in 5G networks," ser. IEEE Communications Magazine, vol. 59, Dec. 2021, pp. 38–44.

[145] G. Yammine *et al.*, "Experimental investigation of 5G positioning performance using a mmWave measurement setup," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Jan. 2021. DOI: 10.1109/IPIN51156.2021.9662535.

[146] A. Xhafa, J. A. del Peral-Rosado, J. A. López-Salcedo, and G. Seco-Granados, "Evaluation of 5G positioning performance based on UTDoA, AoA and base-station selective exclusion," ser. Sensors, vol. 22, 2022. DOI: 10.3390/s22010101.

[147] S. Huang, H.-M. Chen, B. Wang, J. Chai, X. Wu, and F. Li, "Positioning performance evaluation for 5G positioning reference signal," in *2022 2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT)*, Nov. 2022, pp. 497–504. DOI: 10.1109/ICFEICT57213.2022.00093.

[148] J. Shi, G. Zhang, Y. Lin, F. Li, and C. Shen, "Positioning of high-speed trains based on PRS," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, 2022, pp. 578–583. DOI: 10.1109/IWCMC55113.2022.9824177.

[149] J. Zhang and H. Wang, "Online learning based NLOS ranging error mitigation in 5G positioning," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, Jan. 2022, pp. 6487–6492. DOI: 10.1109/GLOBECOM48099.2022.10000807.

[150] C. Jin, I. Bajaj, K. Zhao, W. P. Tay, and K. V. Ling, "5G positioning using code-phase timing recovery," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, May 2021. DOI: 10.1109/WCNC49053.2021.9417556.

[151] Z. Liu, L. Chen, X. Zhou, Z. Jiao, G. Guo, and R. Chen, "Machine learning for time-of-arrival estimation with 5G signals in indoor positioning," ser. IEEE Internet of Things Journal, vol. 10, Jan. 2023, pp. 9782–9795. DOI: 10.1109/JIOT.2023.3234123.

[152] I. Palamà, S. Bartoletti, G. Bianchi, and N. B. Melazzi, "5G positioning with SDR-based open-source platforms: Where do we stand?" In *2022 IEEE 11th IFIP International Conference on Performance Evaluation and Modeling in Wireless and Wired Networks (PEMWN)*, Dec. 2022. DOI: 10.23919/PEMWN56085.2022.9963853.

[153] OpenAirInterface Software Alliance (OSA), *OpenAirInterface, 5G software alliance for democratising wireless innovation*, 2023.

[154] Open Networking Foundation (ONF), *Introduction — Aether Docs 2.2.0-dev documentation*, 2023. (visited on 12/07/2023).

[155] Ettus Research, National Instruments, *USRP software defined radio (SDR) online catalog*, 2024.

[156] Ericsson, "Support for Multiple QoS Class in deferred location requests," 3rd Generation Partnership Project (3GPP), Change Request (CR) 0491 of Technical Specification (TS) 29.122, 2023.

[157] 3GPP, "NR; Physical channels and modulation," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.211, Sep. 2023.

[158] ITU-T, ""Timing characteristics of enhanced primary reference time clocks"," International Telecommunication Union, Recommendation G.8272.1/Y.1367.1, Nov. 2016.

[159] S. Ruffini, M. Johansson, B. Pohlman, and M. Sandgren, "5G synchronization requirements and solutions," ser. Ericsson Technology Review, vol. 2021, Jan. 2021, pp. 2–13. DOI: 10.23919/ETR.2021.9904655.

[160] 3GPP, "NR; physical layer measurements," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.215, 2017.

[161] M. Z. Win, Y. Shen, and W. Dai, "A theoretical foundation of network localization and navigation," ser. Proceedings of the IEEE, vol. 106, Jul. 2018, pp. 1136–1165. DOI: 10.1109/JPROC.2018.2844553.

[162] Ettus Research, National Instruments, *UBX 10-6000 MHz Rx/Tx (160 MHz, X series only)*, 2023.

[163] TE Connectivity, *ANT-5GWWS3-SMA*, 2023.

[164] Ettus Research, National Instruments, *OctoClock-G CDA-2990*, 2023.

[165] C. A. Floudas and P. M. Pardalos, *Encyclopedia of Optimization*. Springer Science & Business Media, 2008.

[166] S. Ruder, "An overview of gradient descent optimization algorithms," ser. arXiv Preprint arXiv:1609.04747, 2016. arXiv: 1609.04747.

APPENDICES

A Communication Activities

While doing the research work needed for this thesis, and apart from the regular scientific conference attendance, I also had the opportunity to present my ongoing research at diverse events and workshops. Attending these events allowed me to explain my scientific activities to people from different backgrounds and receive useful feedback from them.

Cyber Alp Retreat (2020-2023)

The Cyber Alp Retreat is an annual event organized by the Cyber-Defense Campus (CYD), a research laboratory part of armasuisse Science and Technology in Switzerland. This event aims to bring together researchers, partners and external collaborators that are working together with the CYD. The background of the audience covers cybersecurity, data science, and aviation among others. I was able to participate in this scientific retreat 4 years and present my research ideas and results.

3rd Electrosense Workshop (April 2022)

The third Electrosense workshop[9] was hold in Rey Juan Carlos University (Spain), the 27th of April 2022 and was organized by the Electrosense Association. The workshop was targeted at anyone interested in research applications for crowdsourced spectrum monitoring. The workshop served as a forum to discuss new ideas for the Electrosense project. In total there were 15 attendees. I was one of the organizers of the event, in particular the publicity chair, and also participated in two talks in which I described different research topics related to Electrosense but also tied to this thesis, in particular some of the results shown in Chapter 3.

[9] https://workshop.electrosense.org/

RESTART tech camp (September 2023)

The 1st RESTART tech camp[10] was an event to promote wireless network experimentation using software-defined 5G deployments. It took place in Rome from September 13 to 15 2023. The even hosted around 40 PhD students with Telecommunications Engineering background. The main talks were about SDR, custom 5G deployments with SDR devices, O-RAN, virtualized 5G networks, and the applications that can be run with them. Apart from attending, I was invited to give a talk about the paper described in Chapter 4.

[10] https://5g-tech-camp.fondazione-restart.it/past-editions/2023.html

B IRB Protocol

For Chapter 5, since the data that we collected contained personal identifiers that could pose risks on the privacy of drivers. For that, we developed a protocol to ensure that we only collected data from participants that had given us explicit consent (via an electronic form) and we discarded the rest of the data. The process was briefly described in Chapter 5, but I will use the following pages to provide the full protocol to the curious reader.

institute imdea networks

Request for Approval

Human Subjects Research

1. Administrative Information

Application Type ☐ Initial ☐ Renewal ☒ Revision

Will the output of this project be published in scientific journals/conferences?

☒ Yes ☐ No

1.1. Project/Paper title.

IDENTIFYING DRIVER PATTERNS FROM LOW-COST AND DISTRIBUTED SENSOR NETWORKS

1.2. Investigator in charge.

Name	Domenico Giustiniano
E-mail Address	domenico.giustiniano@imdea.org
Phone	-

1.3. Co-PIs and members of the research team.

Title, Name, Surname	**Institution/Country**	**E-mail**
-, Yago, Lizarribar	IMDEA Networks/Spain	yago.lizarribar@imdea.org
-, Alessio, Scalingi	IMDEA Networks/Spain	alessio.scalingi@imdea.org
Prof., Roberto, Calvo	URJC/Spain	roberto.calvo@urjc.es
Dr., Matthias, Schaeffer	SeRo Systems/Germany	schaeffer@sero-systems.de
-, Markus, Fuchs	SeRo Systems/Germany	fuchs@sero-systems.de
-, Pedro Miguel, Sanchez	University of Murcia/Spain	pedromiguel.sanchez@um.es
Dr., Sreeraj, Rajendran	Sirris/Belgium	rsreeraj@gmail.com
Prof., Sofie, Pollin	KU Leuven/Belgium	sofie.pollin@gmail.com
Dr., Gérôme, Bovet	armasuisse/Switzerland	gerome.bovet@armasuisse.ch
Prof., Vincent, Lenders	armasuisse/Switzerland	vincent@lenders.ch

institute imdea networks

1.4. Funding Information.

Is this research being funded by an External Funding Agency?

☐ No
☒ Yes (Specify)
Funding Agency: FPU and armasuisse

institute imdea networks

2. Study Design, Methods and Procedures

2.1. Type of Project/Study.

Please select ALL the categories of work that apply to this proposed project.

☒ Active collection of data (no human biological materials or biomedical data)

☐ Active collection of human biological materials or biomedical data

☐ Use of existing data (not human biological materials)

☐ Use of existing human biological materials

☒ Military/Defense applications

The main risk with regards to military/defense applications would be that a defense organization might use TPMS data to build a civilian tracking network without consent. We also understand that part of this research is funded by a defense organization. However, armasuisse has renounced to access any data collected in this study.

Data will be collected, stored, and processed by the IMDEA Networks researchers mentioned in this document. **NO** access will be given to other researchers from IMDEA Networks, collaborators mentioned in this document not working at IMDEA Networks or third parties.

2.2. Please provide a lay summary of the study, including the purpose, research questions, and hypotheses in non-technical language. Provide supporting background information from prior studies.

The goal of this study is to assess the risks of tire pressure systems in cars. These sensors are sending their data in clear and any person with basic RF knowledge can capture these transmissions without the vehicle owners' consent. By having multiple stations distributed over a city, for example, one could track individual vehicles and patterns with ease.

This privacy risk was analysed several years ago, using high-end receivers in a controlled scenario [1]. Since then, several proposals have been submitted to increase the security of these sensors. However, today, millions of cars are still equipped with sensors without any security measures, and it is feasible to capture those transmissions with much cheaper hardware and easy-to-install software. With this study we want to assess how many cars are still in the wild using these protocols and show how with simple methods and low-cost devices, it would be possible to build a vehicle tracking network. We want to urge legislators to start considering TPMS sensors as a risk and start pushing manufacturers to change their design to use more secure protocols.

institute IMDEA networks

[1] Kenneth Hacker, Scott Graham, and Stephen Dunlap. 2019. Vehicle identification and route reconstruction via tpms data leakage. In International Conference on Critical Infrastructure Protection. Springer, 123–136.

2.3. How this study will contribute to existing knowledge in the field.

The main contributions that we expect from this work:

- Show that many cars today are sending TPMS data without any ID randomization or encryption.
- Show that with low-cost devices it is possible to build a vehicle tracking network.
- Urge legislators to take the steps to enforce more security in the design of these sensors to manufacturers.

3. Participants, Recruitment

3.1. Does your study involve interaction with or observation of human participants (e.g., interviews, surveys, Focus groups, shadowing, etc.)?

☐ No	☒ Yes
Sample Population: Participants that opt-in in our study	
Sample Size: in the order of ~1-10s	

3.2. Participants will be included in your study.

For this study, we will ask for the consent of participants to allow us to collect tire pressure measurements from their cars. We only want to collect IDs belonging to the cars of users that consent to this study. Thus, we will go with each participant to the parking at the back of IMDEA, which is sufficiently isolated from other roads (> 40 m from all roads) so that we only measure the IDs sent by the tires of their car, and then set the sensors so that only those IDs are stored in the database. Afterwards we will discard any ID not belonging to those user that gave us consent via the electronic form.

3.3. Special Protections.

Vulnerable populations require special protections. Explain the measures that you will take to obtain informed consent to protect confidentiality and prevent undue coercion in all the above populations that you have selected.

We will only collect data from people that gave us previous consent via a form that we will send in various IMDEA channels. We will not share data with any other institution or third party.

3.4. Recruitment.

3.4.1. How are subjects recruited? What inducement is offered? Will participants be compensated for their participation? (Append a copy of the letter, advertisement, poster, or recruitment text for online posting, if any)

For one part of the study, we will send a consent form via mail to the people from the institute. We do not expect to provide any compensation.

3.4.2. Describe the inclusion or exclusion criteria for participants in this study.

We will be looking for people that own a car, bring it to IMDEA Networks on a regular basis.

3.5. Location.

State the actual location of the research. For observational studies be as specific as possible (e.g., Main Library)

Data collection will be carried out at IMDEA networks.

3.6. Study Procedures.

In the box below provide a detailed description of the study including all the procedures to be performed (preferably in sequential order). Be sure to specify which procedures are experimental (i.e., testing a new intervention for psychiatric illness) versus which procedures are standard of care. Be sure to include the following information:

- *Methods of data collection*
- *Details regarding experimental interventions*
- *Number, frequency and duration and types of subject contacts (visits, phone calls, internet surveys, mailings, etc.)*
- *Duration of participation for single subject*
- *Anticipated duration of the entire study (up to and including data analysis)*

The plan for this set of measurements will be similar but all data collection and storage will happen at IMDEA Networks. We provide the details of the study below.

Hardware used:

- Raspberry Pi 3
- RTL-SDR version 3
- Antenna (designed for 433 MHz)
- Power/Ethernet cables.

Software:

- Custom Raspberry Pi OS image
- rtl_433 decoder
- InfluxDB (for data collection)

Sensors will be distributed in an area around IMDEA Networks (we expect to add 3-4 sensors) and will only store data belonging to participants that have agreed to their data being collected. We will conduct these experiments for around 2 months.

6

institute imdea networks

4. Risks and Benefits

4.1. From the list below, please select ALL the potential risks that are involved in your study.

☐ Use of deceptive techniques

☐ Use of private records (such as educational or medical records)

☐ Manipulation of psychological or social state such as sensory deprivation, social isolation, psychological stress

☐ Probing for personal or sensitive information in surveys or interviews (e.g.: private behaviours, employer assessments)

☐ Presentation of materials which some participants may consider sensitive, offensive, threatening or degrading

☐ Possible invasion of privacy of subject or subject's family social or economic risk (reputational, cultural, employability, etc)

☐ Identification of child, spousal, or elder abuse

☐ Identification of illegal activity

☐ Risk of injury or bodily harm

☒ Other risks (please specify below)

☐ There are no risks of any kind to any participants enrolled in this study.

For risks that are not listed, please provide an explanation here:

The risks that we identify with the ID collection is that a threat actor could match the pattern of a specific set of IDs and car brand and tie it to a specific person to discover their schedule or driving patterns, for example.

4.2. Describe the nature and degree of the risks or harms selected above. All the risk's harms must be disclosed in the consent form.

The threat model would be as follows: someone could group several IDs to a single car, then would have to guess the car brand and check which dealer could have sold it. From there, one could check which specific car was fitted the tires with those sensor IDs and then obtain the vehicle ID, and then the person that bought it.

Another possibility would be someone that lives in the proximity of the sensor, might recognize the car brand and the patterns and tie it to someone they know.

Based on our experience, through appropriate removal of several bits of the ID of each of the TPMS sensors, the reconstruction process will become more cumbersome for an attacker.

4.3. Explain what steps will be taken to minimize risks or harms and to protect subjects' welfare. If the study will include protected populations, please identify each group and provide an explanatory paragraph for each group.

We will not share this data with any collaborator or third party. However, at the time of publication we will not disclose the full TPMS id. The approach to occlude the id is explained in the following lines.

One of the properties that we observe TPMS sensors have is that the IDs tend to be close to each other when the sensors belong to the same car. For that we need to do the analysis using part of the original ID. Car brands do not reserve specific bits to distinguish themselves from others. So, in general, the obtained IDs for different cars look like:

```
| ID (32 bit)  | Car # | Brand  |
| XXXXXAB3     | Car 1 | Toyota |
| XXXXXF01     | Car 1 | Toyota |
| YYYYY008     | Car 2 | Toyota |
```

We need all ID bits and brand to uniquely distinguish cars from each other. However, when we share, we plan to remove the first 16-24 bits, as well as the car brand, so that very little information is given away. We believe that recovering the ID will be sufficiently complex so that privacy of participants is respected.

4.4. Describe any benefits that individuals may reasonably expect from participation. If there are none, state "None".

We expect this research to stir debate into the reasons of using direct TPMS measurements and sending them wirelessly with no authentication. We want to show how easy building a vehicle tracking network would be.

We want to urge both legislators and manufacturers to take TPMS security seriously and either switch to other methods to monitor wheel pressure or add encryption to the TPMS devices before sending data on the clear.

5. Privacy and Confidentiality

5.1. Will you or any member of your research team collect or have access to any of the personal identifiers listed below? Select ALL that apply.

☐ Name	☐ Date of Birth
☐ Mailing or email address	☐ Phone or Fax numbers
☐ Citizenship/Social Security number	☐ Medical records
☒ License, certificate or Vehicle ID	☐ IP Address
☐ Signatures, handwriting samples	☐ Photos/images/audio recording
☐ Biometric identifiers	☒ Any unique identifier not mentioned

☐ No member of the research team will have access to any personal identifiers

For identifiers not mentioned above, please specify in the box below:

To be more precise, it is **the wheel pressure sensor ID**, not the vehicle ID. We will also be able to collect **the car brand**, as each brand embeds their name in the TPMS fields. To tie it to a specific person, one would need to figure the dealer that sold the car, check which vehicle was fitted which tires, and then verify which person bought the car with the dealer's database.

5.2. Describe why each identifier is required.

By having the ID of each of the tire sensors, it is possible to group them by car (assuming most cars have 4 sensors), and then these grouped sensors can be tracked across the measurements of the different receivers that we set up.

5.3. Describe how will the information be stored and secured, including the types of devices used to store the information. Check all that apply.

5.3.1. How will research data be recorded?

☐ Data Entry Sheet

☐ Computer

☒ Database

☐ Other (specify):

5.3.2. How will data be stored?

☒ Computer

☐ Locked File Cabinet

☐ Locked Office

☐ Other (specify):

5.3.3. How will patient confidentiality be protected?

☐ Coding System

☐ Limiting access to data/specimens

☒ Password protected.

☒ Other (specify):

No access will be given to any researcher outside the researchers from IMDEA Networks mentioned in this document.

5.4. Who will have access to the identifiers? Please describe how you have ensured that non-authorized personnel do not have access to the identifier data.

For the study, all data will be gathered and processed by the researchers from IMDEA Networks (and part of the study), and NO access will be given to any collaborator or third party institution.

To minimize these risks, we will:

- Only involved researchers in the projects from IMDEA will have access to the real identifiers.
- Data will be stored for one year (for integrity purposes) and then it will be deleted.

5.5. What will be done with the identifiers after the study is completed? If the data will be destroyed, please describe when and how.

After one year, we will delete the database. That database will only be accessible to the researchers from IMDEA Networks carrying this study. We will thoroughly remove all files and cached data.

5.6. Will names or other identifiers be used in publications?

☐ No

☒ Yes

Which identifiers and why? (Specify):

We will only use the reduced version of the TPMS IDs (after removing the first 16-24 bits). Nothing else from any other participant will be shared.

6. Consent Procedures

Please attach a copy of the consent form to be used.

6.1. Check ALL that apply

☐ Signed consent will be obtained from subjects and/or parents (if subjects are minors).

☐ Verbal consent**** will be obtained from subjects, using an information sheet (attach) or Script (attach).

☒ Electronic Consent

6.2. Describe in detail in the box below your plans for obtaining informed consent from subjects. Be sure to include the following information.

Who (specifically) will obtain informed consent (these persons must be listed in Part 1.3 as investigators). How long will subjects have to consider whether they wish to participate in the study? When and how will consent be obtained (in person, by telephone, by mail, by inter-net, etc.)

We will send an electronic consent form asking participants to give us permission to collect their TPMS sensor IDs, as well as their license plate (to have the ground truth). We will also give an internal talk to IMDEA workers to explain the study and the consent that we will need to obtain in order to perform the study.

Note that the license plate will not be used anywhere in the publication, nor it will be shared with any other external party. We will need license plate to check the time of entry and exit from IMDEA Networks, so that we have a ground truth for when specific IDs appear in the dataset.

12

institute imdea networks

7. Conflict of Interest

Do you and/or any other investigators associated with the project described in this application have, or appear to have, any act ual or potential conflict of interest with respect to this research?

☒ No ☐ Yes

If yes, explain:

Signature(s)

Investigator

I certify that the information I provide in this application is correct and complete. I also pledge that I will not change any of the procedures, forms, or protocols used in this study without the approval of the Ethics Committee.

Name, Surname: Domenico Giustiniano Date: 11/10/2023

Signature:

13