Real-Time Localization using Software Defined Radio

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Islam Alyafawi

von Jordanien

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik

Copyright Notice

REAL-TIME LOCALIZATION USING SOFTWARE DEFINED RADIO

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Islam Alyafawi

von Jordanien

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Der Dekan:
Prof. Dr. Gilberto Colangelo

Bern, 25.08.2015

Abstract

Service providers make use of cost-effective wireless solutions to identify, localize, and possibly track users using their carried MDs to support added services, such as geo-advertisement, security, and management. Indoor and outdoor hotspot areas play a significant role for such services. However, GPS does not work in many of these areas. To solve this problem, service providers leverage available indoor radio technologies, such as WiFi, GSM, and LTE, to identify and localize users. We focus our research on passive services provided by third parties, which are responsible for (i) data acquisition and (ii) processing, and network-based services, where (i) and (ii) are done inside the serving network.

For better understanding of parameters that affect indoor localization, we investigate several factors that affect indoor signal propagation for both Bluetooth and WiFi technologies. For GSM-based passive services, we developed first a data acquisition module: a GSM receiver that can overhear GSM uplink messages transmitted by MDs while being invisible. A set of optimizations were made for the receiver components to support wideband capturing of the GSM spectrum while operating in real-time. Processing the wide-spectrum of the GSM is possible using a proposed distributed processing approach over an IP network. Then, to overcome the lack of information about tracked devices' radio settings, we developed two novel localization algorithms that rely on proximity-based solutions to estimate in real environments devices' locations. Given the challenging indoor environment on radio signals, such as NLOS reception and multipath propagation, we developed an original algorithm to detect and remove contaminated radio signals before being fed to the localization algorithm. To improve the localization algorithm, we extended our work with a hybrid based approach that uses both WiFi and GSM interfaces to localize users. For network-based services, we used a software implementation of a LTE base station to develop our algorithms, which characterize the indoor environment before applying the localization algorithm. Experiments were conducted without any special hardware, any prior knowledge of the indoor layout or any offline calibration of the system.

Acknowledgement

First of all, I would like to express my sincere gratitude to my supervisor Prof. Dr. Torsten Braun for his excellent advising from my very first to my final steps in conducting the work leading to this thesis. Furthermore, I would like to address my deepest gratitude to my thesis examiner Prof. Dr. Francisco Barcelo-Arroyo for his valuable comments and suggestions and the president of jury Prof. Dr. Matthias Zwicker.

I wish to thank all members of the CDS group for many fruitful discussions and colloquiums. Special thanks go to Dr. Desislava Dimitrova, Dr. Eryk Schiller, and Dr. Eirina Bourtsoulatze for their kind helps throughout my doctoral study and to Daniela Schroth for all administrative work and help. I would also like to thank Prof. Dr. Navid Nikaen for arranging my visit to the Eurecom, France, and his valuable time, discussion and support.

Furthermore, I must acknowledge my best friends who support me to get through the tough times. Last but not least, I would also like to thank my parents, my sisters and brothers. They were always supporting me and encouraging me with their best wishes.

Contents

Contents **ii**

List of Figures **vi**

List of Tables **x**

1 Introduction **1**

1.1 Motivation . . . 1

1.1.1 Connected Problems . . . 2

1.1.2 Research Questions . . . 4

1.2 Structure of the Thesis . . . 5

1.3 Summary and Contributions . . . 7

2 Background and Related Works **11**

2.1 GPS Technology and Indoor Challenges . . . 11

2.2 Bluetooth and WiFi Radio Technologies . . . 13

2.3 The Global System for Mobile Communications . . . 15

2.3.1 GSM Logical Channels . . . 16

2.3.2 Signaling between BTS and MDs . . . 18

2.3.3 Mobile Device Power-up Scenario . . . 21

2.4 The Long Term Evolution . . . 22

2.4.1 LTE PHY Layer . . . 25

2.4.2 Resource Allocation in LTE . . . 28

2.4.3 Radio Measurements in LTE . . . 29

2.4.4 Processing-Time Restrictions . . . 30

2.4.5 Mobile-Edge Computing . . . 31

2.5 Centralized-RAN (CRAN) . . . 32

2.6 Cloudification of CRAN (Cloud-RAN) . . . 33

2.7 Software Defined Radio Systems . . . 34

2.7.1 Hardware . . . 35

2.7.2 Software . . . 37

2.8 Active and Passive Localization . . . 44

2.8.1 Active Localization . . . 45

2.8.2 Passive Localization . . . 45

2.9 Localization Systems 46
2.9.1 Localization in Cellular Networks 46
2.9.2 Input Parameters to Localization Algorithms 47
2.9.3 Fingerprinting-based localization 48
2.9.4 Proximity-based Localization 50
2.9.5 Range-based localization 52
2.9.6 Hybrid Localization Techniques 54
2.9.7 Tracking using Kalman Filters 55
2.9.8 Accuracy Metrics 57
2.10 Channel Modelling 57
2.11 Outlier Detection and Mitigation 60
2.11.1 Indoor Narrowband Radio Propagation 60
2.11.2 Estimators of Radio Metrics 61
2.11.3 Outlier Detection 63

I Passive Localization 67

3 WiFi-Based Localization Systems 69
3.1 Introduction 69
3.2 Sensor Setup 70
3.3 Evaluation Parameters 71
3.3.1 Evaluation A: Technical Characteristics 71
3.3.2 Evaluation B: Manufacturing Discrepancies 76
3.3.3 Evaluation C: Direction-Specific Multipath 77
3.3.4 Evaluation D: Channel Reciprocity 80
3.4 Experimental Setup for LOS Localization 82
3.5 Experimental Results 82
3.6 Conclusions 84

4 GSM-Based Localization Systems 87
4.1 Introduction 87
4.2 GSM Single Frequency-Channel Receiver 88
4.2.1 Passive Receiver Requirements 89
4.2.2 Passive GSM Receiver 91
4.2.3 Experimental Setup 98
4.3 Wideband GSM Cannelizer 104
4.3.1 Optimized CPU-based PFB 107
4.3.2 GPU-based PFB 111
4.3.3 Experimental Setup 112
4.3.4 Conclusions 115
4.4 Distributed Wideband Receiver 116
4.4.1 Challenges in Wideband Capturing 117
4.4.2 Proposed Architecture for Distributed Signal Processing . 119

4.4.3 Real-Time Load Balancer 120
4.4.4 Signal Compression 122
4.4.5 Experimental Setup 122
4.4.6 Experimental Results 123
4.4.7 Compression Performance 125
4.4.8 Conclusions 126
4.5 Localization Algorithms 126
4.5.1 Proposed Solutions 126
4.5.2 Experimental Setup 131
4.5.3 Experimental Results 132
4.6 Robust Radio Estimators 135
4.6.1 Proposed Algorithm for Outlier Detection and Filtering . . 135
4.6.2 Experimental Setup 137
4.6.3 Experimental Results and Discussion 138
4.7 Conclusions 145

5 Hybrid-Network Localization Systems 147
5.1 Introduction 147
5.2 Characteristics of Captured Signals 148
5.3 Hybrid-Network Indoor Localization 149
5.3.1 Aggregation, Filtering and Interpolation 150
5.3.2 Hybrid Signal Preprocessing 151
5.3.3 Hybrid Location post-processing 154
5.4 Experimental Setup 155
5.5 Experimental Results 157
5.5.1 Signal Quality 157
5.5.2 Received Power vs. Distance 158
5.5.3 Hybrid Localization 158
5.6 Conclusions 163

II Network-Based Localization 165

6 Real-Time LTE RAN Validation for SDR 167
6.1 Introduction 167
6.2 Processing Budget in LTE Frequency Division Duplex (FDD) . . 168
6.2.1 Radio Access Network (RAN) on a PC 169
6.2.2 Cloud-based Long Term Evolution (LTE) RAN 169
6.3 Evaluation 170
6.3.1 Experiments 171
6.3.2 Results and Analysis 171
6.4 Conclusions 173

7 LTE-Based Localization Systems 175
7.1 Introduction 175
7.2 Architecture 177
7.2.1 Algorithm 178
7.2.2 Blind Channel Characterization 179
7.2.3 Positioning and Tracking Algorithms 181
7.3 Experimental Setup 181
7.4 Measurement Results 184
7.4.1 Channel Characterization 184
7.4.2 Static Positioning 187
7.4.3 Mobile Tracking 191
7.5 Discussion and Remarks 193
7.6 Conclusions 194

8 Experiences and Lessons Learned 197
8.1 Performance Measures 197
8.2 System Costs 199
8.3 Application Scenarios 199

9 Conclusions and Outlook 201

Bibliography 212

List of Figures

2.1 Availability of Cellular and WLAN signals indoors 13
2.2 Graphical representation of Wi-Fi channels in the 2.4 GHz band. . 14
2.3 General architecture of GSM cellular network 16
2.4 Downlink and uplink bands in GSM 16
2.5 Logical channels classification in GSM 17
2.6 GSM OSI model 18
2.7 GSM Layer 3 message structure 18
2.8 Burst structure in GSM 19
2.9 Burst structure in GSM 20
2.10 Different types of GSM TDMA downlink/uplink multiframes . . . 23
2.11 GSM message encoding 24
2.12 General architecture of LTE cellular network. 24
2.13 LTE Radio frame definition (short cyclic prefix). 26
2.14 Location based services in MEC. 31
2.15 Cloud RAN evolution. 32
2.16 SDR generalized functional architecture 35
2.17 SDR block diagram based on USRP hardware 36
2.18 ExpressMIMO2 architecture [72] 37
2.19 PFB channelizer. 40
2.20 The concept of wideband channelizer. 40
2.21 Airprobe system architecture [13] 42
2.22 OpenBTS system architecture [141] 42
2.23 OAI eNB architecture. 43
2.24 Architectural overview of ZMQ messaging library. 44
2.25 Fingerprinting localization algorithm. 49
2.26 Proximity-based localization: ANs pull the estimated location propotional to RSSI value. 50
2.27 Localization area provided by trilateration approach 53
2.28 Iterations in Kalman filter 56
2.29 Different types of radio signals in typical indoor environment. . . 58
2.30 An example of bivariate MCD estimator 65

3.1 WiFi and Bluetooth SN module: a) Gumstix board, b) TP link adapter 71
3.2 Experiment A: Setup. 72

3.3 RSSI time variation of three MDs. 72
3.4 Histogram of the Bluetooth RSSI levels for three MDs measured at the same SN; distances one and three meters. 74
3.5 Histogram of the WiFi RSSI levels for three MDs measured at the same SN; distances one and three meters. 74
3.6 Bluetooth boxplots of three MD, RSSI measured by the same SN; distances one and three meters. 75
3.7 WiFi boxplots of three MD, RSSI measured by the same SN; distances one and three meters. 75
3.8 Scenario B: setup 76
3.9 Bluetooth boxplots of three SNs measuring the RSSI values of the same MD; distances one and three meters. 78
3.10 WiFi boxplots of three SNs measuring the RSSI values of the same MD; distances one and three meters. 78
3.11 Scenario C: setup 79
3.12 CDFs in eight communication directions. 80
3.13 Evaluation setup for LOS localization using WiFi SNs. 81
3.14 Bi-directional radio measurements for indoor LOS communication 82
3.15 Network-based localization using 5 min aggregation. 83
3.16 User-assisted localization using 5 min aggregation. 83

4.1 General design of a passive receiver system 89
4.2 Time synchronization challenge in GSM uplink passive receiver . 91
4.3 Ideal solution for fast retuning between downlink and uplink channel. 92
4.4 Retunning the frequency of an USRP N210 device. 93
4.5 Algorithm for passive synchronization recovery for an uplink signal. 94
4.6 Normal burst structure 95
4.7 Detection algorithm for RACH bursts 97
4.8 A city map showing information about IAM Institute and a Sunrise BTS 99
4.9 Experimental setup for decoding GSM uplink messages 99
4.10 Signal, noise, and interference power levels in a GSM uplink channel100
4.11 Downlink evaluation experiment setup. 100
4.12 Distributions of decoded and non-decoded downlink messages. . . 101
4.13 Distributions of decoded and non-decoded uplink messages. . . . 102
4.14 Combined distributions of decoded and non-decoded uplink messages. 103
4.15 Power gain relation to the number of decoded messages per cell. . 104
4.16 Unencrypted messages distribution over a working day period. . . 105
4.17 Schematic diagram of a wideband GSM receiver. 106
4.18 USRP receiver front-end architecture. 107
4.19 Aligned filter taps in GR implementation. 109
4.20 GR FIR AVX dot product. 110
4.21 AVX unpack operation and multiplication. 111

4.22 PFB Channelizer on GPU. 112
4.23 PFB channelizer benchmark overview. 113
4.24 PFB channelizer performance vs. number of channels. 114
4.25 PFB channelizer performance vs. number of taps. 115
4.26 CPU performance of a single machine using PFB GPU channelizer. 116
4.27 a) Analog and quantized signal in the time domain. b) Quantized noise in the time domain. c) Signal and quantized noise in the frequency domain. 117
4.28 High interference for low power signals. 118
4.29 Downlink captured power at of a wideband receiver. 119
4.30 Broadcast approach for a wideband receiver. 120
4.31 Unicast approach for a wideband receiver. 120
4.32 Client/Worker communication overview. 122
4.33 Compression scenario to increase link efficiency. 122
4.34 Experimental setup using distributed unicast approach. 123
4.35 CPU usage of distributed unicast approach with a real-time GSM data from a USRP. 124
4.36 TCP blocks throughput chart. 125
4.37 Operation of differential Received Signal Strength Indicator (RSSI) and LWC algorithms. 127
4.38 Operation of combined differential RSSI algorithm. 128
4.39 Operation of circumcenter algorithm. 129
4.40 Global System for Mobile Communications (GSM) SDR system. . 130
4.41 Indoor experiment environment setup. 131
4.42 Estimated locations using Weighted Circumcenter (WCC) algorithm: L9 in case of closed doors and L4 in case of open doors. . . 134
4.43 LOS and NLOS power relation during day and night. 135
4.44 Outlier detection and filtering algorithm. 136
4.45 GSM sensor processing chain. 137
4.46 GSM burst power. 139
4.47 Shadow mitigation using robust estimator. 139
4.48 Shadow mitigation using robust estimator. 141
4.49 Experiment setup. 142
4.50 Estimated location using different aggregated points. 143
4.51 Localization error mean and standard deviation with and without outlier detection and filtering techniques. 144
4.52 Signals residual of 4 GSM ANs using the Huber estimator 144
4.53 Signals residual of 4 GSM ANs using the MLE estimator 145

5.1 Signal overhearing setup 148
5.2 Simplified illustration of CDRSS algorithm. 149
5.3 a) Preprocessing for colocated and distributed antenna setups. b) Post-processing for both antenna and distributed antenna setups. . 150
5.4 Example of hybrid signal preprocessing of colocated antennas . . 152

5.5 Probabilities for WiFi and GSM $\mathcal{D}$ values 153
5.6 a) Virtual antenna setup. b) Probability calculation of RSSI measurements. c) RSSI and coordinate calculations for VANs. 154
5.7 a) colocated antennas setup. b) distributed antennas setup 155
5.8 Distance between WiFi and GSM measurements. 156
5.9 Comparing Signal Interference on colocated Antennas 157
5.10 Comparing WiFi and GSM path loss 159
5.11 Estimated location of hybrid localization system, a) colocated antenna setup, b) distributed antenna setup. 159
5.12 Comparison between hybrid and non-hybrid algorithms. 160

6.1 Proposed architecture for LTE network-based positioning. 168
6.2 Processing orders in OAI. 170
6.3 OpenStack management architecture. 170
6.4 eNB processing time for transmitting packets. 172
6.5 Processing time for transmitting (OFDMA) and receiving (SC-FDMA) packets at eNB given full PRB allocation at 20 MHz. 174

7.1 Network-based positioning in LTE. 175
7.2 System architecture of RTPoT. 178
7.3 Experimental setup 182
7.4 eNB setup with single Antenna 183
7.5 Fading in subcarriers. 185
7.6 NLOS power characterization. 186
7.7 NLOS timing advance measurements. 187
7.8 Robust estimation of PLE using MCD. 188
7.9 Positioning experiment setup. 190
7.10 Positioning experiment result with 30 sec aggregation. 190
7.11 Prosed outlier mitigation algorithm using different estimators. . . 191
7.12 Positioning experiment setup. 191
7.13 Matlab Processing time of total 2.4 sec. 192
7.14 Tracking experiment using double Kalman filters. 193

List of Tables

2.1 Different training sequences for a GSM NB. 21

3.1 Experiment 1: Response Rates 76
3.2 Experiment 2: Response Rates 77
3.3 Error comparison of localization error mean, standard deviation (meter) and RMS. 84

4.1 Success rate of passive receiver 101
4.2 End-to-end compression measurements. 125
4.3 Error comparison of Centroid and Circumcenter based approaches (meters). 133
4.4 Error comparison of WCC localization algorithm using raw and filtered based approaches (meters). 142

5.1 Performance evaluation of WiFi and GSM localization systems . . 161
5.2 Performance evaluation of the proposed hybrid-network localization 162

7.1 Positioning performance comparison. 189

Chapter 1

Introduction

An important requirement for most wireless communication systems is detecting the presence (or finding the location) of particular remote objects for both indoor and outdoor environments [79]. In the terminology of this dissertation, location (localization) and position (positioning) are synonym words. The number of location services is vast and rapidly growing. The Global Positioning System (GPS) initially addressed the outdoor localization problem globally. Other navigation systems have been also proposed, such as the Indian Regional Navigation Satellite System (IRNSS) and the European Global Satellite-Based Navigation System (Galileo) [190]. However, the performance of satellite-based positioning solutions is not reliable in urban and indoor environments because satellite radio signals do not penetrate well through walls of large buildings. In many real-world indoor scenarios, spanning businesses, social and personal desire, service providers are motivated to provide and control their own localization system. For example, in real-world markets, a shop owner may want to detect the presence, localize, or even navigate potential customers. This example can be extended to localize items, materials, and equipment in warehouses, hospitals, airports, and similar places. An application's Quality of Service (QoS) is in direct relation to its localization accuracy. Operators of wireless communication systems, such as GSM and LTE, utilize various localization methods to estimate the location of an object, expressing the distance of the target object from a set of reference points [134]. However, producing an accurate distance estimation in the presence of noise and interference of radio signals, especially in urban and indoor environments, remains a challenge and of our interest and motivation in this research [123].

1.1 Motivation

In recent years, wireless devices have become more powerful and pervasive. Furthermore, these devices often support more than one radio technology, such as WiFi, Bluetooth, GSM, and LTE. Information about users' location is of equal importance to network operators, users themselves, and third-party service providers, such as administrative people in hospitals and shopping malls. Development of detection and localization services contains two steps: data acquisition and processing. Depending on the application scenario, it might not be possible to rely on users or even network operators for data acquisition. Hence, third party service providers have to take this task on their own. For example, the number of international visitors in a certain area might be private data that neither the network operator nor end users are willing to reveal to a third party service provider. The focus of our research is to provide solutions for data acquisition for third party service providers and localization algorithms that show reliable performance in indoor environments. In theory, all indoor localization systems can be used outdoors. Nevertheless, dissimilarities in indoor environments are a particular challenge that cause performance to differ significantly among localization systems.

1.1.1 Connected Problems

In outdoor urban areas with a considerable density of buildings and surrounding obstacles or indoors, several parameters deviate the radio signal from its usual behavior and degrade the performance of localization systems.

- **Multipath propagation:** Line of Sight (LOS) propagation can behave drastically differently depending on the environment. For example in outdoor urban LOS and indoor LOS, the presence of severe multipath fading in harsh propagation environments may cause a significant impact on the performance of localization systems [77, 163, 16]. Multipath is the reception of multiple copies of the transmitted signal, where each is arriving from different propagation paths (reflections from the many scattered objects, walls, doors, and windows). Multipath signals combine in either a constructive or destructive manner, which distorts the received signal power and phase, well-known as the fast fading phenomenon. Hence, multipath signals arrive at the receiver downgraded, phase-shifted, and time-delayed. The unexpected behavior of the received radio signals due to multipath propagation *prevents* localization systems to estimate accurately locations of target objects and hence, degrades the localization performance. For example, the fast fading phenomenon disturbs the distance power relationship required for localization systems.

- **NLOS reception:** An intuitive definition of Non Line of Sight (NLOS) reception is when an obstruction is in the direct path between the transmitter and receiver. However, this definition is not comprehensive. The viability of NLOS propagation varies considerably based on the operating frequency, but works well with lower frequency ranges [158]. High-frequency signals (between 3 and 30 MHz) can penetrate buildings quite easily. As the operating frequency increases, radio signals can penetrate buildings also, but to a lesser extent depending on the type of obstruction and size (in comparison to the signal wavelength) [158]. Note that GPS and most cellular networks operate in the ultra-high-frequency range between 300 MHz and 3 GHz.

Above 3 GHz, such as in Ultra Wide Band (UWB) systems, NLOS propagation becomes inefficient. Obstructions, such as concrete walls or metal cupboards, between the transmitter and the receiver, will block the signal. Moreover, when obstructions completely block the direct signal path, the first non-direct path component may result in a significant bias that corrupts the propagate time from the transmitter to the receiver $\triangle t$ [158] or the angle of arrival θ and ultimately the position estimate. NLOS reception makes the distance–power relationship less predictable with the variety of obstructions, making it difficult to estimate the distance accurately [51, 16]. It is important to mention that indoor environments often undergo multipath propagation *and* NLOS reception simultaneously.

- **Radio Interference:** Cellular networks, such as GSM and LTE, operate in a controlled and a managed licensed spectrum, where spectrum resources, such as frequency and time, are well optimized to avoid as much signal interference as possible. However, other technologies, such as WiFi and Bluetooth, operate in the unlicensed Industrial, Scientific and Medical (ISM) band. In the unlicensed band, transmitted packets from nearby devices might collide, which results in a higher variation in collected radio measurements for localization, especially received signal strength. The amount of interference depends on the number of nearby active devices operating at the same frequency.

Due to the nondeterministic nature of these three parameters, the instantaneous behavior of indoor radio signals and corresponding estimated locations will be inconsistent. This means that repeating a radio measurement within the same indoor settings might generate different localization results.

Two other main challenges related to all localization systems in general (outdoor and indoor) are the real-time operation and cost [187, 121]. A localization system has to process radio signals and apply the localization algorithm in real-time or quasi- real-time. In GPS-based applications, the localization process is applied at the user side. Hence, the response time of the system is agnostic to the number of simultaneously served users. However, if the localization process is applied to a central unit, such as in cellular networks, the complexity of the localization system and overall response time increase with the number of served users. Another challenge for indoor localization systems is the deployment cost. From an economic point of view, the implementation cost per user should be maintained as low as possible. With these challenges, indoor localization requires hardware implementation at the building level (or even at the floor level). Sophisticated processing units might support a large number of served users with a short response time. However, this comes with the high cost of these units.

1.1.2 Research Questions

The practical usefulness of a localization system depends on the context in which it is used. In this dissertation, the context is the indoor environment. More specifically, the context is defined by a use case scenario to localize users in office buildings, exhibition centers, shopping malls or hospitals. More precisely, our target is to localize people carrying wireless devices, which emit radio signals in individual operational scenarios. To evaluate if a localization system is useful in a unique context, we define a set of system metrics, namely, accuracy, response time, deployment and calibration costs, adaptiveness and user costs. The desired accuracy of the localization system in this context is finding people with an accuracy of several meters. For example, people working in an office building will not be moving much, so the response time of the target system is not a significant issue for localization. However, the system should consider the response time for tracking scenarios, such as in the case of hospital environment. Deployment and calibration costs have an inverse relationship with the system adaptiveness. During the operational time of the localization system, the environment is subject to change, which may negatively affect the performance without additional calibration. Ideally, we would like the system to be agnostic to environmental changes and to adapt automatically without calibration. Finally, it is required to have comparable localization results agnostic to the user devices' manufacturer and with a minimum participation of users in the localization process. We believe that minimum participation of users can be achieved in *passive* localization systems without collaboration with end users or network operators (if possible) [54].

Based on the scenario discussed above, we define the following main research questions:

1. ***Which radio metrics are best suited for indoor localization?*** The acquired radio metrics depend mainly on hardware capabilities of the receiving device. Devices with an array of antennas are capable of estimating the signal's direction of arrival. The accuracy of this method increases with the array size, which is proportional to the overall system cost. Almost all devices that are capable of capturing the intended radio signal are also capable of performing time and power measurements. On the one side, time measurements require tight synchronization between transmitters and receivers. Moreover, sufficient timestamps resolution for indoor environments is required. On the other side, power measurements do not require any special hardware or synchronization and are mainly used in our research.

2. ***Which parameters affect received radio signals?*** Considering power metrics, such as RSSI, as a result of the first question, there is a set of *uncontrolled* parameters that affect the quality of measurements at the receiver side. Technologies like WiFi, which operate in the unlicensed band, are highly affected by interference because WiFi devices are transmitting on a

certain band without coordination. General parameters, such as multipath propagation, NLOS reception, moving objects and device orientation, are degrading the quality of radio signals. This degradation appears as unpredictable fluctuations of a radio metric at the receiver.

3. ***How to enhance the quality of collected indoor radio metrics?*** The quality of collected radio measurements is in direct relation to the localization accuracy. Due to the dynamic nature of the indoor environment, offline calibration in the area of interest is not a valid solution because the environment might be changed at the time of the online measurement step. However, it might also be possible to detect contaminated radio signals by the indoor environments during the online measurement step using statistical outliers detection and mitigation algorithms. The effectiveness of this approach is in direct relation to the number of collected measurements. However, it increases the response time of the localization system.

4. ***How to localize a device?*** Estimating the location of a target device (x,y coordinates in the 2D problem) can be done using multiple methods. In fact, a large number of localization methods (with a particular focus on power-based ones) exist in the literature. Most power-based localization methods require radio measurements at a set of known location Anchor Nodes (ANs). However, each method has its own characteristics and may not perform equally under different operational conditions. We target localization algorithms, such as proximity-based algorithms that fulfill our requirements in terms of ease of deployment, reliable localization performance, and most relevant, being operationally agnostic to specifications of target devices. The localization accuracy of such algorithms depends highly on the number of used ANs and number of collected radio measurements.

1.2 Structure of the Thesis

After providing the introduction in this chapter, where we motivated the need for indoor localization and introduced a set of connected problems and open research questions, the remainder of this dissertation is organized into eight chapters.

Chapter 2: Background and Related Works presents first the challenges that are facing localization and tracking applications using GPS in indoor and urban environments. As alternative solutions, we present later an overview of some radio technologies available indoors, such as WiFi, GSM, and LTE, and can be used to tackle the problem of indoor localization. Since service providers operate on a large scale, we present the evolution of networks Base Transceiver Station (BTS) towards cloud-RAN and the advantages that service providers gain in terms of reduction in cost, maintenance, and power consumption. The simplest definition of cloud-RAN is a software implementation of the RAN running in the cloud. Hence, we present possible hardware and software platforms that can be used for cloud-RAN implementations and possibly radio measurements than can be extracted and used for localization. Given our requirements for a minimum impact on target users and minimal calibration work, we present a set of localization algorithms, their advantages and challenges. The biggest challenge facing indoor localization algorithms is the multipath propagation and NLOS reception, which creates contaminated signals at the receiver (outliers). Hence, we present state-of-the-art work for outlier detection and mitigation that works with a single variable, such as RSSI, or with two variables, such as RSSI and distance. Having these sets of measurements without outliers, would allow us to characterize the indoor environment using well-known channel models presented in Section 2.11.

Chapter 3: WiFi-Based Localization Systems studies a set of parameters that affect the behavior of RSSI measurements and consequently the performance of localization systems. Some of these parameters are the variation in Mobile Devices (MDs) manufactures, the orientation of a MD, manufacturing discrepancies among sensor nodes. To explore the impact of such parameters in a real indoor environment, we conducted a set of experiments using mainly WiFi technology. Furthermore, a set of localization experiments in indoor LOS environments were conducted.

Chapter 4: GSM-Based Localization Systems introduces first the challenges facing a Software Defined Radio (SDR) system to overhear uplink transmission of GSM MDs. Based on our understanding of the GSM signal processing of the Physical layer (PHY), we present our solution for a passive GSM receiver. The capturing rate performance of the proposed receiver was compared with a benchmark on a downlink channel. However, since GSM MDs in a certain area may connect to multiple GSM BTSs operating at different frequencies, we support our passive receiver with wideband capturing capability. To do so, we present the two required steps of wideband signal channelizing and channelized streams processing. For the first step, we evaluate first the performance of a CPU-based implementation of a wideband signal channelizer, which we improve later using advanced Intel-CPU instructions. Then, we present our solution using GPU-based channelizer and show the processing advantages over the CPU-based implementation. For the second step, we propose a distributed processing approach over an IP network to a set of servers dedicated to stream processing. After a complete setup for data acquisition in GSM is established, we present the challenges to localize GSM MDs in a passive way and illustrate our proposed solutions to overcome these challenges. A set of experiments that show the impact of the indoor environment and performance of the proposed algorithm were conducted using a real indoor setup. To improve the quality of radio measurements, such as RSSI, before being fed to the localization algorithm, we present last our proposed algorithm to detect and remove outliers in such measurements.

Chapter 5: Hybrid-Network Localization Systems motivates the problem of developing hybrid localization algorithms using multiple radio interfaces actively used at one MD. A discussion about different characteristic, in terms of operating frequency, transmitted power, and the data rate was presented with particular focus on WiFi and GSM radio technologies. Given the flexibility of antenna deployment of our SNs, it is possible to have distributed and colocated antenna for WiFi and GSM interfaces. Moreover, we presented two solutions for processing radio measurements of captured radio interfaces before the localization algorithm or after. Finally, a set of real-indoor experiments was conducted to show the performance of the proposed solutions.

Chapter 6: Real-Time LTE RAN Validation for SDR presents first the importance of operating a LTE RAN on the cloud for localization algorithms, especially network-based positioning. Then, we present the OpenAirInterface as an open source candidate for operating a LTE RAN on the cloud. A set of experiments were conducted to show the processing requirements for different modules inside the RAN implementation. Moreover, we modeled the minimum amount of required CPU for each operational mode in the LTE-RAN.

Chapter 7: LTE-based Localization Systems presents first the overall architecture of network-based positioning using LTE eNB software implementations running on a cloud environment. Detailed explanation of data acquisition and localization parameters were presented. To overcome the challenge posed by the indoor environment, a novel iterative approach is presented to characterize the indoor environment (estimate the path-loss exponent) online without any prior calibration process. To validate the proposed solution, a set of real-indoor experiments were conducted for both static and moving users. Finally, discussion and remarks about the proposed solutions and possible work extensions were presented.

Chapter 8: Learned Lessons presents a summary of the whole thesis, results achieved, challenges faced, and lessons learned for real-time radio-based localization.

Chapter 9: Conclusion and Outlook summarizes the main contributions of the dissertation and answers the main research question.

1.3 Summary and Contributions

The main contribution of this dissertation is divided three main parts:

- **Evaluation of real indoor setups:** While most work on localization provides a theoretical basis for controlled environments, this dissertation assesses the consequence if there is no control over certain parameters, to identify which factors we should consider and which we can disregard in the design of a positioning system. In real indoor environments, we performed our research using well-known and highly adopted technologies, such as WiFi, GSM, and LTE. We analyze the usability of four signal metrics, namely, instantaneous values, probability distribution, median and percentiles, mean and standard deviation, as well as the signal's detection rate.

  - For WiFi-based localization systems, we investigate the impact of (i) device's technical characteristics, (ii) manufacturing discrepancies, (iii) direction-specific multipath propagation, and (iv) interference versus the amount of active nearby WiFi devices.
  - For passive GSM-based localization system, we investigate the impact of (i) detection rate of identified message between day and night and (ii) fluctuations in received signal strength versus mobility of people, opening and closing doors.
  - For network-based LTE localization system, we investigate the impact of (i) channel fading between subcarriers, (ii) modelling the received signal strength in indoor environments and (iii) studying the accuracy of timing advance measurements indoors.

- **A passive GSM wideband receiver:** Most GSM-based localization systems rely on data acquisition from the network operator or directly from end-users, e.g., by installing a certain application at their MDs. Given our research requirements for passive acquisition of GSM uplink signals, we contributed the following:

  - We developed a single-band passive receiver that was able to capture, decode, and parse uplink and downlink messages with high reliability. The receiver performance was tested against a downlink benchmark and showed near 95% success capturing rate and against an uplink self-developed Android application and showed near 70% success capturing rate.
  - We developed an advanced wideband receiver that can channelize in real-time up to 76 GSM channels using an optimized CPU implementation of a channelizer (channel splitter) and more than 125 GSM channels using a newly implemented GPU channelizer. The hosting machine may not have enough processing resources to analyze all channelized streams. Hence, we developed a distributed processing system over an IP network to a set of servers dedicated for processing channelized streams.

- **Reliable localization algorithms without the participation of users:** For localization applications, we contributed to four main parts to localize MDs without their collaboration:

- We developed two proximity-based localization algorithms, namely, Combined Differential RSSI (Combined Differential RSSI (CDRSS)) and Weighted Circumcenter (WCC), which can overcome the lack of MDs' transmitted power and unknown indoor environment to localize MDs. Both algorithms were verified using the proposed GSM passive system and achieve a localization error mean in the range of 3m.
- An original filtering algorithm based on robust statical estimators was developed, which detects and filters out contaminated radio signals (outliers). The proposed algorithm was also verified using the proposed GSM passive system and shows an improvement of localization error mean in the range of 75% at different locations compared to WCC algorithm without filtering.
- To enhance further the localization accuracy of indoor passive localization systems, we present a hybrid-network indoor localization method using multiple radio interfaces. The proposed solution detects changes in signal quality and employs different weighting schemes to favor the signal with higher quality. The proposed solution was verified using MDs with active WiFi and GSM interfaces and achieves a localization error mean down to 1.7m.
- Finally, we propose RTPoT - a light-weight positioning and tracking algorithm, which operates at the edge of LTE network closer to users and relies only on network measurements. RTPoT benefits from the high packet rate in LTE to filter out outliers and characterize indoor radio channels without any a prior knowledge of the deployment environment. The specific environment parameters, such as the path-loss exponent, are passed to a range-based localization algorithm to localize LTE devices. The proposed solution was verified using indoor LTE network and shows a localization error mean in the range of 2.7m with a deployment density of 0.89 base-station/100m^2.

Chapter 2

Background and Related Works

In areas where satellite technologies like GPS fail to provide a reasonable level of positioning accuracy, there is a definite interest among network operators, service providers, and regulatory bodies in serving these environments with positioning solutions that are accurate, scalable and also cost effective. To support indoor localization, various solutions leverage available indoor radio signals, such as WiFi, GSM, and LTE. However, service providers for such technologies operate on a large scale. To keep system resources optimized and minimize upgrade and maintenance costs, network operators have a clear trend to exploit these radio access technologies on the cloud. The RAN operation on the cloud requires two main components: a SDR hardware responsible for exchanging information with end-users and a SDR implementation applying the radio system in software. Our ultimate goal is indoor localization. Hence, we focus on radio parameters exported from indoor SDR systems for localization. These parameters will be fed later to a localization algorithm to estimate locations of target objects. However, due to the challenging characteristic of the indoor environment, we require first, the removal of contaminated signals from radio measurements before they can be fed to the localization algorithm and second, a robust algorithm to model the target environment instantaneously is required.

2.1 GPS Technology and Indoor Challenges

A satellite navigation system, such as GPS, consists of multiple satellites that send radio signals from space to capable devices (GPS receivers) on earth. First, GPS receivers perform time synchronization with satellites' clocks. Then, the distance between a satellite and a GPS device is calculated considering the time it takes the radio signal to travel from a satellite to the device $\triangle t$ multiplied by the speed of light c [120]. GPS devices have to lock onto at least four or more satellites to estimate their location (longitude, latitude, and altitude). GPS is an excellent example that offers high accuracy, scalability, and privacy (devices only know their own locations) for a broad range of outdoor applications. Higher accuracy signals of GPS satellites are restricted to military and governmental agencies. Other satellite-based solutions, such as Galileo, promise to offer higher accuracy than GPS publicly for every user [190]. However, such solutions are still in the development phase. Since the GPS signal is weak, about 1 x 10^{-16} watts measured at the surface of the earth, devices can only achieve lock when they have a clear line of sight with satellites. GPS signals pass through transparent objects, such as thin clouds, glass, and so on. Nevertheless, the GPS signal does not pass through solid objects, such as buildings. Hence, GPS cannot localize capable devices indoors, in dense urban areas, or similar environments. Moreover, GPS devices might require a long time to lock onto satellites, from 30 seconds to several minutes. Thus, GPS devices are relatively power-hungry, making their intensive use inefficient on battery-based devices, such as smartphones. In terms of security, GPS signal modulation and frequency spectrum is publically available. Thus, GPS is vulnerable to blocking, jamming or spoofing attacks [75].

Because conventional GPS has the difficulty of providing reliable performance when signal conditions are poor, several solutions have been proposed to improve the situation. Indoor GPS is a solution based on devices that act as a virtual satellite with signal correction capabilities. Assisted GPS (A-GPS) or Differential GPS (D-GPS) are two other solutions targeting areas uncovered by typical GPS signal. AGPS describes a system where outside sources, such as an assistance server and ground stations, help a GPS receiver to perform its task to make range measurements and position solutions. The ground stations or assistance servers communicate with the GPS receiver via a wireless link. The ground station may also transmit GPS-like signals, this approach is called Pseudolites navigation system. Hence, the Pseudolites approach provides an extra source to GPS receivers. If the ground stations are transmitting the difference between their position indicated by the GPS and their exact known fixed location, this technique is called Differential GPS (D-GPS). A-GPS and D-GPS improve the location accuracy from 15m down to 10cm. However, the deployment and maintenance costs, together with high battery consumption remains an obstacle from adopting these solutions to a broad range of applications.

Unlike GPS transmitter, BTSs of cellular networks, such as GSM and LTE, are located in relatively close distance to end users [50]. Each BTS covers a particular geographical area called a cell. Moreover, the relatively high transmitted power of cellular BTSs (compared to GPS) makes cellular radio signals available outdoor and in most indoor areas as shown in Figure 2.1. Hence, several solutions for indoor localization solutions take advantage of available cellular signals indoors to localize and track MDs [11, 169, 8]. These solutions are attractive due to the widespread adoption of users to cellular MDs [50]. However, these solutions can be extended to all available indoor radio [75]. A Large number of papers, e.g., [80] and [191], argue that UWB radio offers excellent means to determine a device's location with high precision. Unfortunately, at the time of this work, UWB-based solutions have a longer time for deployment and are expensive. Decawave published

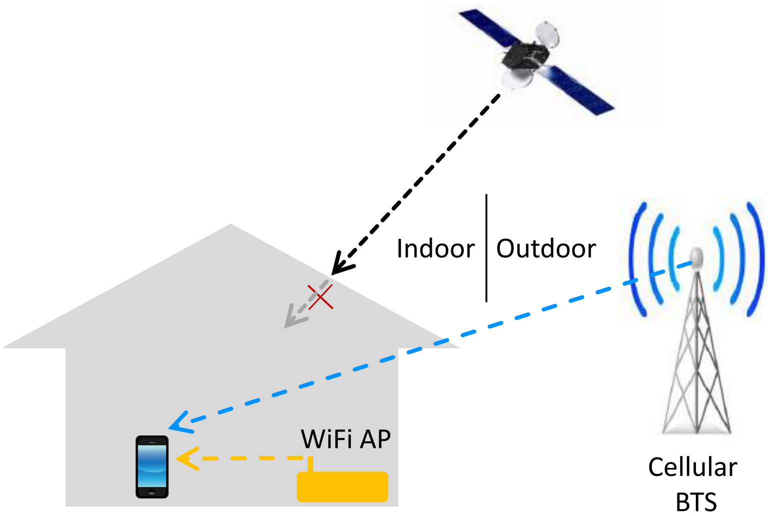


Figure 2.1: Availability of Cellular and WLAN signals indoors

in late 2014 the world's first single-chip UWB transceiver [55], enabling the development of cost-effective localization and tracking systems. Equally, many studies campaign for the use of WiFi [46, 86] or Bluetooth [91, 127]. Both Bluetooth and WiFi have an ubiquitous support from personal devices and are convenient for the quick development of practical solutions. Cellular-based localization systems have the advantage to localize MDs outdoor and indoor. However, the sparse deployment of cellular BTSs compared to Wireless Local Area Network (WLAN) Access Points (APs) makes cellular-based localization systems less accurate [189].

2.2 Bluetooth and WiFi Radio Technologies

Bluetooth is a widely used technology for short-range communication. An essential feature in Bluetooth is the low-power consumption compared to other technologies, such as WiFi. Bluetooth uses short-range 2.4 GHz ISM band spread spectrum (2400 – 2483.5 MHz). As illustrated in Figure 2.2-b, Bluetooth consists of 79 radio channels with a bandwidth of 1 MHz each. To mitigate signal interference, Bluetooth implements the frequency-hopping Code Division Multiple Access (CDMA) access scheme, where the frequency of the channel is changing very rapidly according to a sequence that constitutes the spreading code. The modulation onto the carrier frequency is done by using Gaussian Frequency-Shift Keying (GFSK). For an inquiry procedure (scanning procedure to discover available devices) to be successful, a Bluetooth device should be in discoverable mode.

The inquiry procedure contains two 16-channel subsets known as trains. Each train takes 10 ms to complete. By specification [97], each train must be repeated 256 times to allow sufficient time to collect all inquiry responses. The specification also dictates that at least three train switches must occur, meaning that there

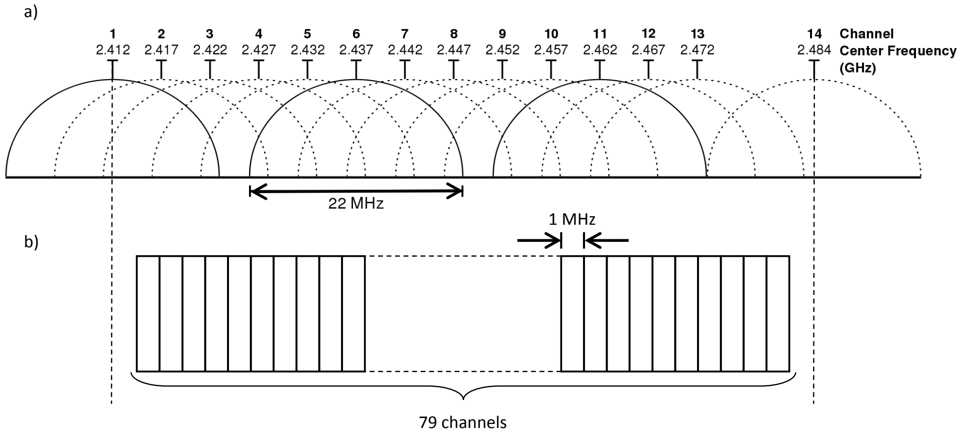


Figure 2.2: Graphical representation of Wi-Fi channels in the 2.4 GHz band.

must be two iterations of each train. Running both trains twice, at 256 times per iteration, allows the inquiry device to ensure that all listening devices in range will be on a standard frequency and be in the inquiry scan substate during at least one inquiry time slot. A complete inquiry procedure requires 10.24 seconds (2 trains x 2 iterations x 256 times x 0.01 seconds = 10.24 seconds).

WiFi is one of the most used indoor wireless technology. WiFi follows a series of standards in IEEE 802.11 WiFi systems exchange data in the unlicensed 2.4 GHz ISM band. Data is transferred on BPSK and QPSK constellations at 11 Mbps. WiFi uses both Frequency Hopping Spread Spectrum(FHSS) and Direct Sequence Spread Spectrum (DSSS) technologies defined in the IEEE standard. The bandwidth of a WiFi channel is 22 MHz, which uses a chipping rate of 11 MHz (for spread spectrum access technique). As illustrated in Figure 2.2-a, a WiFi system looks like a broadband Jammer to the Bluetooth node. Contrary to Bluetooth, there is no inquiry procedure defined in WiFi. A MD becomes visible only after it sends out a request to associate to an access point. In the associated state, there is a periodic exchange of control messages. In ad hoc wireless networks, WiFi sensors periodically send beacons. The beacon interval can be set in the configuration of the sensor. The default beacon interval is set to 100 ms, which provides excellent performance for many applications. In the beginning, one of the WiFi sensors assumes the responsibility for sending the beacon. After receiving a beacon frame, each sensor waits for the beacon interval and then sends a beacon if no other sensor does so after a random time delay. This ensures that at least one sensor will send a beacon, and the random delay rotates the responsibility for sending beacons.

When a Bluetooth transmission occurs at the same frequency occupied by a simultaneous WiFi transmission, some level of radio interference can occur. the amount of interference depends on the strength of each signal. When a Bluetooth device encounters interference on a channel, it deals with the problem by hopping to the next channel and retransmitting again. In such a way, Bluetooth attempts to avoid interference from a WiFi network. The result will be a degradation of data throughput (or detection rate).

Despite the large number of studies addressing radio-based indoor localization for WiFi and Bluetooth [82, 137, 173, 189], only a few investigate the various factors that impact the localization system [15, 96, 167]. To our knowledge, a detailed study, covering several factors and reflecting their impact on both Bluetooth and WiFi signals has not been conducted so far. We present in Chapter 3 our experimental setups and findings for such a study.

2.3 The Global System for Mobile Communications

GSM is perhaps the most successful technology of the last twenty years [64]. GSM divides the target geographical area into smaller radio areas called cells. Each cell has its own BTS to exchange information with MDs over the Air interface (*between MDs and BTSs*) [64]. As illustrated in Figure 2.3, a number of BTSs is controlled by one Base Station Controller (BSC). Most of user signaling is done inside the Mobile Switching Center (MSC). Users' information is stored in the Visitor-/Home Location Register (VLR/HLR) and gets authenticated by the Authentication Center (AuC). Over the Air interface, GSM is a FDD-based system that uses a combination of two techniques: Frequency-Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA). GSM uses two bands of frequencies, one for the downlink that carries information from the network to MDs, and another band for the uplink that carries information from MDs to the network. The lower frequency band is assigned to uplink communication while the higher frequency band is allocated to downlink communication. This design is based on the fact that transmitting radio signals on lower frequencies experiences less path loss than it does at higher ones (c.f. Section 2.10). Hence, this design supports battery saving on MDs. A set of frequencies, called channels, is allocated for each cell in each band as illustrated in Figure 2.4.

Each downlink or uplink band equals 25 MHz and each frequency channel is 200 KHz. Hence, the available transmission channels in each direction (downlink and uplink) N_{chan} = 25 MHz/200 KHz - 1 = 124 (one channel is used as a guard channel). It is important to mention that each channel on the downlink has its paired channel on the uplink. Downlink or uplink channels are defined by their Absolute Radio Frequency Channel Number (ARFCN). Since there is a 20 MHz as a guard band between the downlink and uplink bands, each channel pair is spaced by 45 MHz (20 MHz + 25 MHz). The frequency set of each cell is used by several geographically spaced cells according to a predefined reuse pattern (defined by the network).

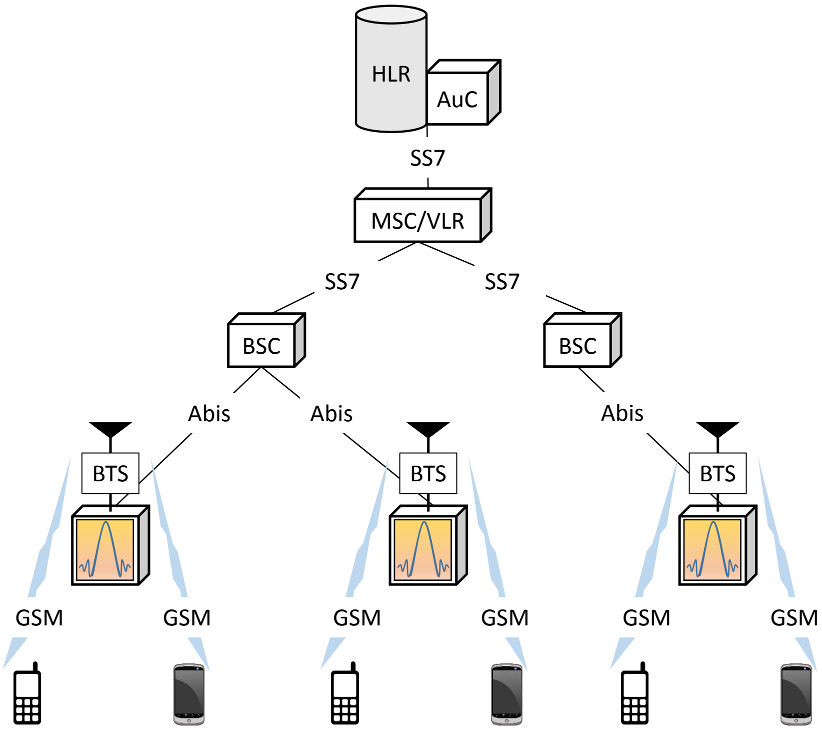


Figure 2.3: General architecture of GSM cellular network

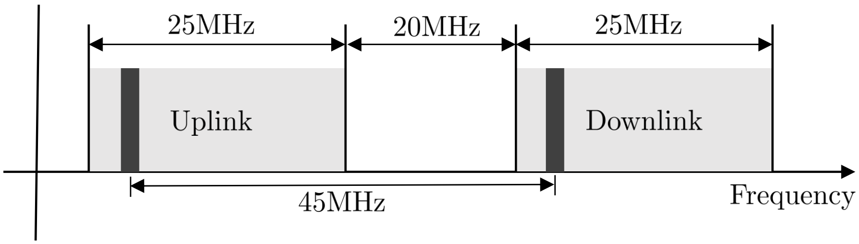


Figure 2.4: Downlink and uplink bands in GSM

2.3.1 GSM Logical Channels

GSM manages the transmission between different MDs using TDMA in a round robin fashion. This means that each ARFCN is shared among different MDs over time. GSM uses a variety of channels in which the data is carried. GSM channels are divided into physical and logical channels. The physical channels are distinguished by their resource allocation (frequency and timeslot), whereas the logical channels are distinguished by the type of information they carry (control or data). The logical channels can be divided into common and dedicated channels as illustrated in Figure 2.5 [64]. The Common Channels (CCH) contain multiple specific channels:

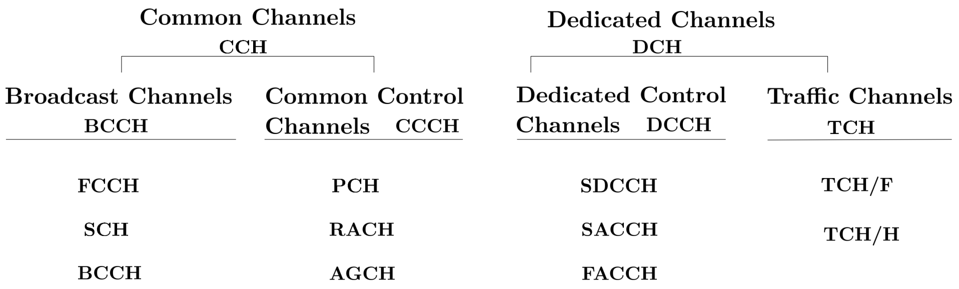


Figure 2.5: Logical channels classification in GSM

- Broadcasting Channels (BCH) (only downlink):
  - Broadcast Control Channel (BCCH) carries a repeating pattern of system information messages, which describe the identity and a special configuration of the broadcasting BTS.
  - Frequency Correction Channel (FCCH) carries a sequence of zeros, which produces a fixed tone in the GMSK (Gaussian Minimum Shift Keying) modulator output equal to 67.7 KHz. —item Synchronization Channel (SCH) allows MDs to identify nearby BTSs and synchronize to the serving BTS TDMA frame. The SCH data payload carries the TDMA Running Frame Number (RFN) and the Base Station Identity Code (BSIC).
- Common Control Channels (CCCH):
  - Paging Channel (PCH) (Downlink) is used by the BTS to find a target MD. Paging messages use one of the MD's identities as discussed later in Section 4.2.1.
  - Random Access Channel (RACH) (uplink) is used to synchronize uplink transmission of GSM MDs with their serving BTS.
  - Access Grant Channel (AGCH) (Downlink) carries BTS's responses to channel requests sent by MDs via the RACH.

The Dedicated Channels (DCH) in GSM can also be divided into two sub-categories:

- Dedicated Control Channels (DCCH):
  - Standalone Dedicated Control Channel (SDCCH) is used for some transactions, such as initial call setup, MD registration, and SMS transfer.
  - Fast Associated Control Channel (FACCH) is used to transfer signal quality information from MDs to assist the BTS with the handover process.

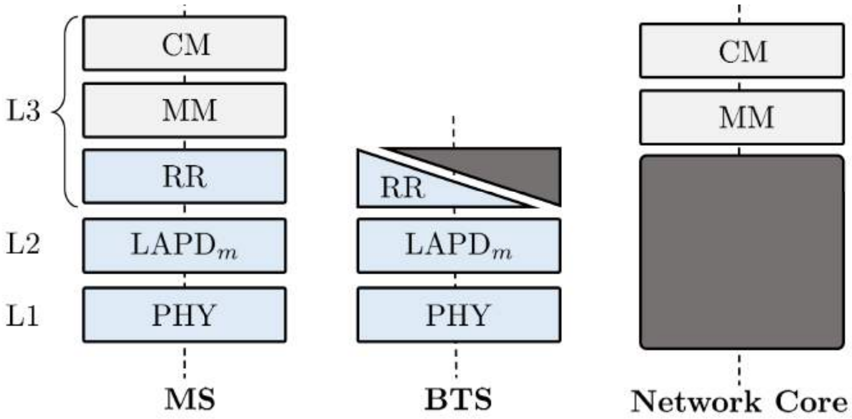


Figure 2.6: GSM OSI model

Data	Message Type 8 Bit	Type ID 8 Bit

Figure 2.7: GSM Layer 3 message structure

- Slow Associated Control Channel (SACCH) replaces speech data with signaling information for short periods.

- Traffic Channels (TCH):
  - Encoded Speech, such as voice audio that is converted into digital form and compressed.
  - User data, such as text messages or a picture message.

2.3.2 Signaling between BTS and MDs

The only GSM-specific signaling of OSI layers 1, 2, and 3 can be found on the Air interface, where a signaling protocol called LAPDm (Link Access Procedure, Channel Dm) is used. Understanding the structure of these layers is important for decoding and parsing GSM messages. An illustration of these layers and their interaction between MDs and the serving network is depicted in Figure 2.6 [64].

The Network Layer (Layer 3): Layer 3 handles network resources allocation, mobility management and call-related management between various network entities as shown in Figure 2.6 [69]. The message format of Layer 3 is depicted in Figure 2.7. The Type ID is used on the Air interface to classify all messages into groups and allows, within Layer 3, addressing various protocols as follows:
Radio Resource Management (RR) layer is used to manage logical and physical channels, such as channel allocation, handover, Timing Advance (TA) calculation, power control, and frequency hopping. Mobility Management (MM) layer uses allocated channels by the RR to exchange data between MDs and the serving

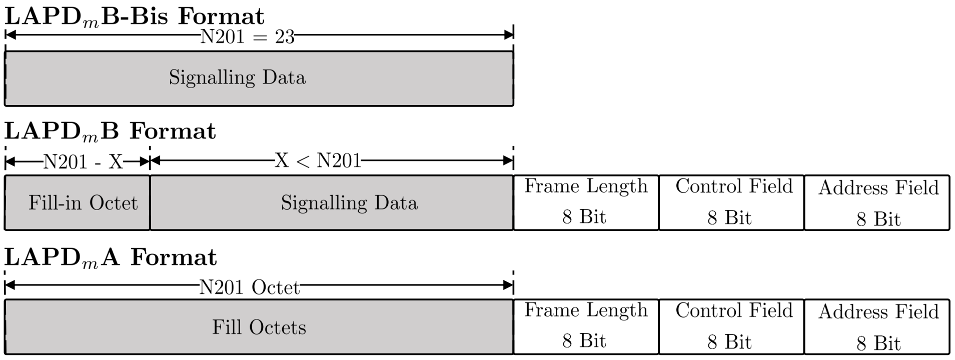


Figure 2.8: Burst structure in GSM

network. MM manages location and security information. Connection Management (CM) layer uses also allocated channels by the RR to support (i) Call Control (CC) sublayer responsible for managing call procedures, (ii) Supplementary Services (SS) sublayer responsible for managing supplementary services, such as call forwarding, waiting and hold, and (iii) Short Message Service (SMS) sublayer responsible for handling the routing and delivery of short messages between MDs.

LAPDm Frame Structure (Layer 2): Three different formats of LAPDm messages with length (23 bytes) are defined as shown in Figure 2.8 [64, 67]. The A-format is used when no payload is available. This is important to maintain an active connection in DCCH downlink or uplink channels. The B-format is used for transfering actual point-to-point signaling data at every DCCH and ACCH channels. If the data size is less than the frame size, this remaining space is filled with fill-in octets. The Bbis-format is used when the frame identification is not necessary, such as in point-to-multipoint channels (BCCH, PCH, and AGCH are CCCHs). Both the A-format and the B-format are used in both directions uplink and downlink. The Bbis format is required for the downlink only. The address field defines the communication direction, the service access point identifier, and a protocol discriminator. The control field specifies the data sent or received sequence number and 1-bit identifier. The frame length field is defined as follows: Bit 0 indicates if the current octet is the last one of the frame length indicator field. Bit 1 indicates if the entire message is longer than the data field of what the LAPDm frames allow. If yes, the information has to be partitioned and transmitted on multiple consecutive frames. Bits 2–7, indicate the actual length of the information field. N201 [=18 octets for SACCH; = 20 octets for FACCH/SDCCH].

The Physical Layer (PHY) (Layer 1): Each time-domain window over the Air interface is called a burst. Bursts are organized in cycles of 8 slots called TDMA frames. The burst duration is 156.25-bit periods that represent about 577 μs. Note that the 8.25 bits of each burst are considered as guard bits. This is just a representation for a period during which nothing is transmitted to ensure that one

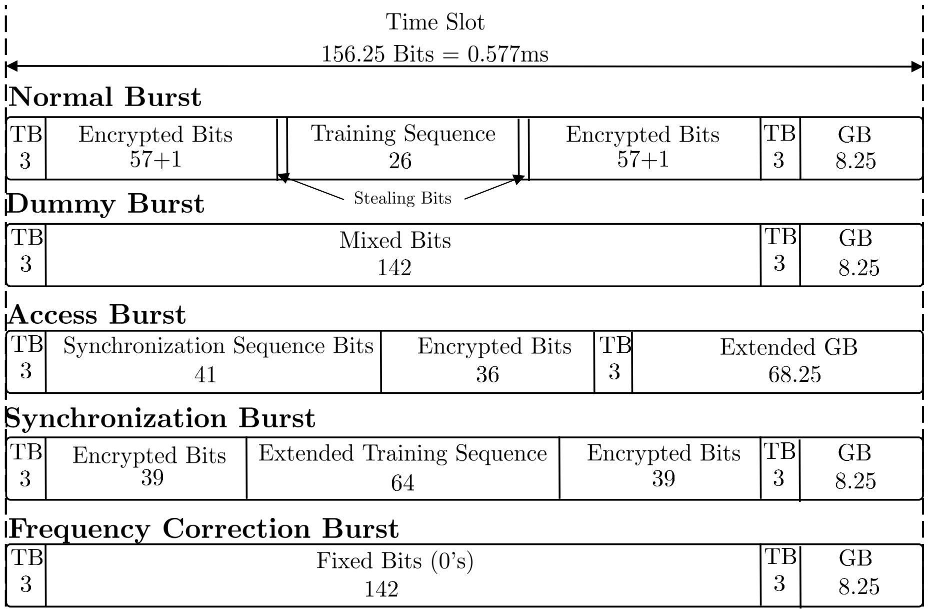


Figure 2.9: Burst structure in GSM

burst does not run into another. Figure 2.9 illustrates different types of bursts each of which has its particular functionalities [64].

The Normal Burst (NB) is used to carry data and most signaling messages. A NB has a total length of 156.25 bits. It is made up of two 57 bits information blocks, a 26 bit training sequence used for equalization, 1 stealing bit for each information block, which indicates whether the burst is being used for voice or data, 3 tail bits at each end, and an 8.25 bit guard sequence [66]. Training sequence bits are used for equalization, i.e., bits which get the BTS and MDs in ”tune” with each other. As shown in Table 2.1, there are 8 different Training Sequence Codes (TSCs) with corresponding training sequence bits. Note that the TSC is defined by the GSM network but fixed per BTS. Each burst leaves 3 bits on each side containing 0’s. This designed is made to compensate for the time required for the power to rise to its peak during a transmission. The Dummy burst is similar to the NB but carrying meaningless data. Access bursts are transmitted on uplink RACH channels by MDs The length of the guard band in the Access burst 68.25 bits defines the maximum allowed distance for accessing the network, which is equivalent to 37.5 km. Synchronization Burst (SB) is broadcasted on a downlink SCH channels. SB payload data contains information about the serving BTS as the TDMA RFN and the BSIC. The extended training sequence is used by MDs for precise time (symbol) synchronization with the serving BTS. Frequency correction Burst (FB) is broadcasted on FCH channels for frequency tuning (adjusting the sampling clock). The 142 fixed bits contain a standard modulated signal of only ’0’s.

In the uplink, MDs synchronize their transmission to the BTS using the time-base counter (initialized from downlink synchronization), and shall be 3-time slots behind the transmission received from the BTS. The round trip propagation delay BTS-MD-BTS may cause time overlap between different MDs transmission, this overlap occurs due to different transmission distances between MDs and the serving BTS, each 553m represent 1-bit delay == 3.69 μs. To compensate the propagation delay, the GSM system uses the synchronization sequence in the access burst to calculate how far it arrived after the expected arrival time, i.e., TA, and report the TA to the MD. To compensate the propagation delay, the GSM system uses the synchronization sequence in the access burst to calculate how far it arrived after the expected arrival time, i.e., TA and report the TA to the MD. For any other communication on the uplink, signaling or data transfer, MDs use NBs.

2.3.3 Mobile Device Power-up Scenario

To capture a GSM MD messages, it is important to understand how the MD behaves just after the power is turned ON [64].

- Immediately after turning on power, a MD starts searching for C0 channels (channels that contains FCCH, SCH, and BCCH frames). To do this, the MD shall examine all Radio Frequency (RF) channels in the system and tune to the channel with the highest signal power.

- Then, the MD shall attempt to synchronize its frequency and timing clocks to read BCCH information. C0 channels can be identified by, for example, searching for frequency correction bursts. The RF channel C0 can only be carried on specific multiframe structures as illustrated in Figure 2.10 (combinations 4 and 5). The Power Spectral Density (PSD) of FB is located 67.6 KHz from the BCCH central frequency.

Table 2.1: Different training sequences for a GSM NB.

Training Sequence Code (TSC)	Training sequence bits
0	0,0,1,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,1,0,0,1,0,1,1,1
1	0,0,1,0,1,1,0,1,1,1,0,1,1,1,1,0,0,0,1,0,1,1,0,1,1,1
2	0,1,0,0,0,0,1,1,1,0,1,1,1,0,1,0,0,1,0,0,0,0,1,1,1,0
3	0,1,0,0,0,1,1,1,1,0,1,1,0,1,0,0,0,1,0,0,0,1,1,1,1,0
4	0,0,0,1,1,0,1,0,1,1,1,0,0,1,0,0,0,0,0,1,1,0,1,0,1,1
5	0,1,0,0,1,1,1,0,1,0,1,1,0,0,0,0,0,1,0,0,1,1,1,0,1,0
6	1,0,1,0,0,1,1,1,1,1,0,1,1,0,0,0,1,0,1,0,0,1,1,1,1,1
7	1,1,1,0,1,1,1,1,0,0,0,1,0,0,1,0,1,1,1,0,1,1,1,1,0,0

- The SB of the SCH channel in the TDMA frame immediately follow the FB as illustrated in Figure 2.10. The SB has a long training sequence of 64 bits, which is used for time synchronization. In this way, the MD can read and decode synchronization data from the SB, the BSIC and the RFN.
- The exact channel configuration of the selected cell is obtained from the BCCH data as well as the frequencies of the neighboring cells. The MD can now monitor the PCH of the current cell and measure the signal levels of the six neighboring cells with the strongest received signal level.

> In the GSM terminology, time synchronization consists of two different tasks: (i) symbol synchronization where the symbol boundaries are defined and (ii) frame synchronization where the TDMA frame boundaries are defined. In the first task, the receiver will be able to identify the starting symbol of a GSM burst. In the second task, the receiver will be able to identify the location of that particular burst within the whole Multiframe structure defined inside the BCCH information. It is important to mention that both tasks do not require any change of the receiver sampling clock but the sampling counter.

For both downlink and uplink channels, information coming from the data link layer (184 bits) will be spread over four NBs. As illustrated in Figure 2.11, the 184 information bits will be encoded using a shortened binary cyclic code, bits (184-223) are the parity bits. The resulted block (224 bits) is completed with four tail bits equal to zero at the end of the block. A 1/2 rate convolutional encoder is applied to the resulted block. The result is a new block with 456 bits length. The coded bits are then reordered and interleaved according to a known algorithm, without adding any extra bit. The interleaved block is split into eight sub-blocks, with 57 bit length each. Each two sub-blocks are mapped to two BIs (Burst Indexes). Every two BIs will be mapped into one NB. Hence, four NBs are required to hold a complete message. From the RFN information inside the SB, the MD gains precise information about the current timer inside the 51-multiframe. Moreover, since the MD is time synchronized with the serving BTS multiframe timer, the MD can extract the four NBs and apply the reversed signal processing to extract bits of the data link layer.

2.4 The Long Term Evolution

LTE is the most recent mobile communication protocol for high-speed data and voice, which outperforms GSM. The high-level network architecture of LTE (c.f. Figure 2.12) is comprised of three main components: the user equipment, the radio access network, and the evolved packet core. The first LTE-capable devices were

Multiframe structure type 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Downlink	F	S	B				C				F	S	C				C				F	S	C				C				F	S	C				C				F	S	C				C				-

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Uplink	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Multiframe structure type 5

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Downlink	F	S	B				C				F	S	C				C				F	S	D 0				D 1				F	S	D 2				D 3				F	S	A 0				A 1				-
	F	S	B				C				F	S	C				C				F	S	D 0				D 1				F	S	D 2				D 3				F	S	A 2				A 3				-

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Uplink	D 3				R	R	A 2				A 3				R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	D 0				D 1				R	R	D 2			
	D 3				R	R	A 0				A 1				R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	D 0				D 1				R	R	D 2			

Multiframe structure type 6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Downlink	B				C				C				C				C				C				C				C				C				C				C				C				-	-	-

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Uplink	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Multiframe structure type 7

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Downlink	D0				D1				D2				D3				D4				D5				D6				D7				A0				A1				A2				A3				-	-	-
	D0				D1				D2				D3				D4				D5				D6				D7				A4				A5				A6				A7				-	-	-

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Uplink	A1				A2				A3				-	-	-	D0				D1				D2				D3				D4				D5				D6				D7				A0			
	A5				A6				A7				-	-	-	D0				D1				D2				D3				D4				D5				D6				D7				A4			

F: FCCH **B**: BCCH **D**: SDCCH **R**: RACH **S**: SCH **C**: CCCH **A**: SACCH

Figure 2.10: Different types of GSM TDMA downlink/uplink multiframes

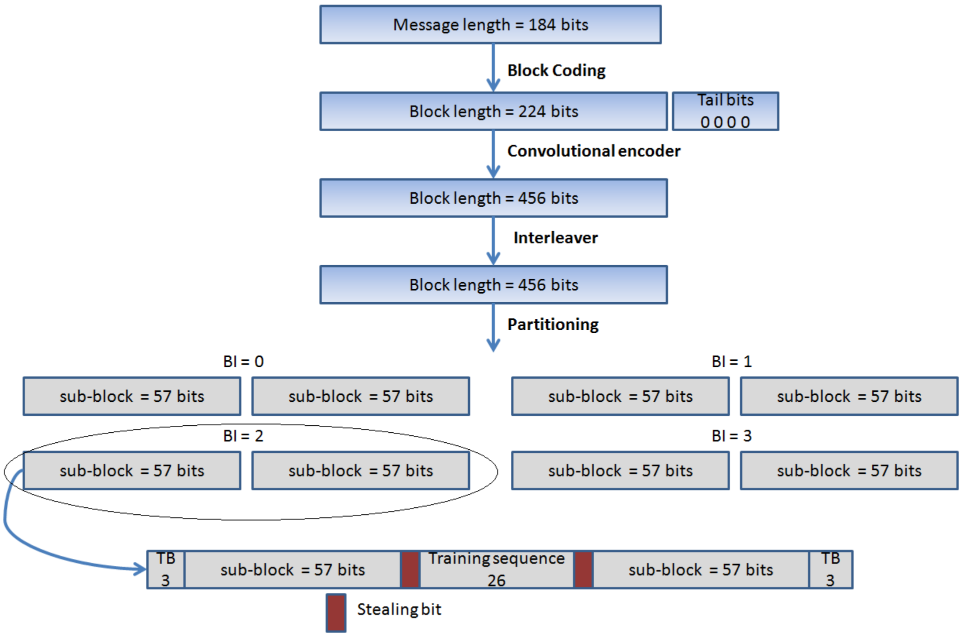


Figure 2.11: GSM message encoding

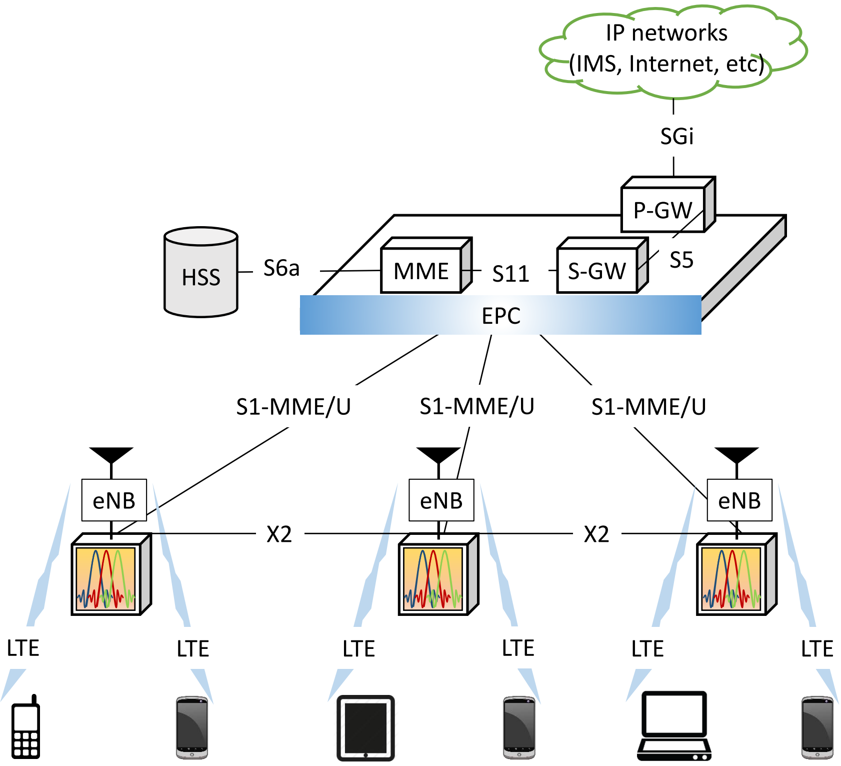


Figure 2.12: General architecture of LTE cellular network.

USB dongles. Other devices are available nowadays, such as smartphones, tablets and laptops with embedded LTE. The internal architecture of the User Equipment (UE) for LTE is identical to the one used by GSM. A LTE UE is a mobile equipment that runs the target standards and equipped with a Subscriber Identity Module (SIM) card used for identification and authentication purposes. The LTE RAN handles the radio communication between UEs and the core of the network [9]. The RAN base station, called evolved Node B (eNB), sends and receives radio transmissions to all served UEs over the Air interface. Each eNB connects with one Evolved Packet Core (EPC) using the S1 interface and it can also be connected to nearby eNBs using the X2 interface. Figure 2.12 shows some components that form the LTE EPC to make it simple [50]. A brief description of each component is as follows. The Home Subscriber Server (HSS) contains information about all the network operator's subscribers. The Mobility Management Entity (MME) controls the high-level operation of UEs by means of signalling messages and HSS. The Serving Gateway (SGW) acts as a router, which forwards data between eNBs and the Packet Data Network (PDN) Gateway (PGW). Finally, the PGW communicates with the outside world such as the internet, private corporate networks or the IP multimedia subsystem.

2.4.1 LTE PHY Layer

The LTE PHY is based on Orthogonal Frequency Division Multiplex (OFDM) [50]. In LTE downlink, Orthogonal Frequency-Division Multiple Access (OFDMA) is used, whereas in the uplink Single-Carrier Frequency-Division Multiple Access (SC-FDMA) is used. OFDMA is a multicarrier transmission technique, which divides the available spectrum into several narrowband subcarriers with few kilohertz subcarrier spacing while each one is being modulated with a low data rate stream [34]. OFDM is further using the Fourier principle to transform all these subcarriers to the time domain, where we deal afterward with what so called, OFDM symbols.

In LTE, subcarrier spacing is defined with 15 KHz, which relates to an OFDM symbol duration of 66.7 µs. A resource block (RB) consists of 12 subcarriers occupying a bandwidth of 180 KHz and 7 OFDM symbols in case of short cyclic prefix (6 OFDM symbols in case of extended cyclic prefix) spanning one slot of 0.5 ms as illustrated in Figure 2.13. One RB is the smallest unit assigned to an UE in the uplink for transmitting, or in the downlink for receiving information. 7 OFDM symbols are used (a guard time interval = 4.7 µs) and 6 OFDM symbols (= 16.67 µs). A resource element (RE) is the smallest unit defined the in LTE, which consists of one OFDM subcarrier during one OFDM symbol interval. Subcarriers of one RE can be modulated by QPSK, 16QAM, or 64QAM for downlink while 64QAM in uplink is not yet released (UE category 5 and higher).

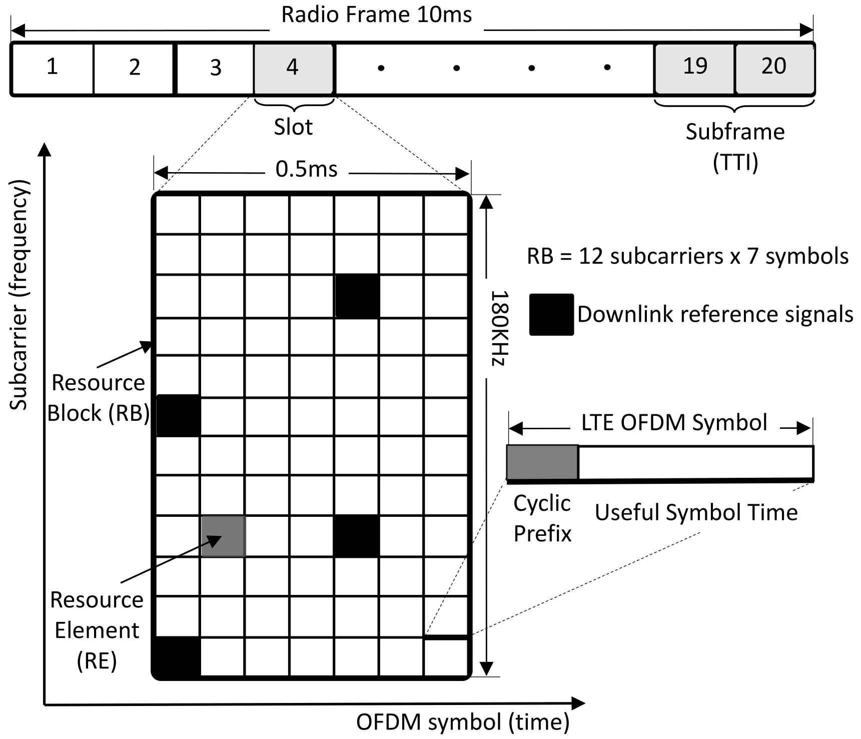


Figure 2.13: LTE Radio frame definition (short cyclic prefix).

LTE uses rate adaptation algorithm that adapts the Modulation and Coding Scheme (MCS) and different transport block size (TBS) [70]. This adaptation is running according to the quality of the radio channel measured by Channel Quality Indicator (CQI) and thus improves the bit rate and robustness of data transmission. In the downlink, where pure OFDMA is used, the phase and power of all subcarriers are adding up, which accumulate to a high peak-to-average-power-ratio (PAPR) [34]. PAPR requires linear power amplifier and RF power transistor capable of handling this power. These requirements are very challenging and hard to be met mainly in the uplink with a limited battery on the mobile device. Hence, SC-FDMA was introduced in the uplink, which is trying to combine the advantages of multicarrier schemes such as OFDMA with the single-carrier schemes such as FDMA [34]. The uplink PAPR depends on the modulation scheme to be used.

LTE Downlink

LTE physical layer uses in downlink two types of signals [9, 75]:

- Physical signals that are generated in the PHY layer and do not convey information to/from higher layers such as (i) primary and secondary synchronization signals (PSS and SSS), (ii) positioning reference signals (PRS), and (iii) cell-specific reference signals (CRS).

- Physical channels that retrieve information from higher layers, such as (i) physical broadcast channel (PBCH), (ii) Physical Downlink Control Channel (PDCCH), and (ii)Physical Downlink Shared Channel (PDSCH).

PSS and SSS are used by the UE to obtain cell identity and frame timing. CRS signals are transmitted to well-defined REs in every downlink subframe and span the entire cell bandwidth. In the frequency domain, every 6th subcarrier carries a reference symbol out of the generated reference signal pattern. In the time domain, every 4th OFDM symbol contains REs that carry a reference signal pattern. CRS signals are used for channel estimation and coherent demodulation at UEs. eNBs enable CRS patterns with a reuse factor of 3. However, since signal to noise and interference ratio for CRS of neighbouring cells needs to be at least -6 dB, the positioning reference signal was introduced to provide more power and less Inter-Cell-Interference (ICI) for positioning purposes with a reuse factor of 6. PRS is characterized mostly by the signal bandwidth, typically 1.08 MHz (6 Physical Resource Blocks (PRBs)). PRS configurations are submitted to each UE individually from the SMLC.

The PBCH is used to broadcast general information about the serving cell. PDCCH carries all information required by an UE to transmit and receive in the PDSCH. The mapping of physical channels to physical resources takes place in the Medium Access Control layer (MAC).

LTE Uplink

Similar to the downlink, LTE physical layer uses two types of signals on uplink as follows [9, 75]:

- Physical signals are generated in the PHY such as (i) demodulation reference signal (DMRS) and (ii) sounding reference signal (SRS).
- Physical channels retrieve information from higher layers such as (i) Physical Uplink Shared Channel (PUSCH), Physical Uplink Control Channels (PUCCHs), and physical random access channel (PRACH).

To help eNBs to estimate the channel quality at different frequencies and hence, find a suitable scheduling allocation, the SRS has been defined at the UE with a constant amplitude and over a well-defined bandwidth. PRACH is used by the UE to initiate uplink time synchronization with the eNB and ask for PUSCH allocation for data transmission. Using the PRACH message, the TA value is estimated at the eNB and reported back to the UE. One TA unit is equivalent to 16 T_S at 30.72 MHz sampling rate (T_S = 1/30720000 = 32.55 ns) and is translated to 78.12m. Resource element allocation of the PUSCH, which is used to carry UE uplink information, is decided by the serving eNB. Uplink resource allocation, i.e., scheduling, is calculated every transmission time interval (TTI) of 1 ms duration. PUCCHs are transmitted over dedicated resource element, which are located at the edge of the available bandwidth. The PUSCH DMRS is mapped to the 4$^{\text{th}}$ SC-FDMA symbol of the slot during normal cyclic prefix and to every 3$^{\text{rd}}$ SC-FDMA slot during extended cyclic prefix. Demodulation reference signals associated with the PUSCH are used by the eNB to perform channel estimation and allow for coherent demodulation of the SC-FDMA received signal.

2.4.2 Resource Allocation in LTE

In LTE eNB, a number of PHY and MAC layer parameters are jointly controlling the transmission's resource allocation for the eNB downlink and UE uplink through a scheduler [9]. There is no dedicated channel allocated between an UE and the serving eNB. Time and frequency resources are dynamically shared between UEs in downlink and uplink channels. Scheduling is the process through which the eNB allocates shared radio resources among UEs and enables the transmission and reception of data in uplink and downlink channels [50, 9]. The scheduler is an essential component of the eNB MAC layer used to optimize the cell or system capacity maximization subject to fairness and multiple quality-of-service (QOS) constraints. At each TTI, the scheduler signals 3 ms in advance the downlink and uplink resource allocation to UEs using Downlink Control Information (DCI) [24]. There are multiple DCI formats depending on the allocation type (downlink or uplink) and its particular usage. A DCI conveys the following information to UE required prior to send and receive data, namely (i) the physical resource allocation, (ii) MCS and its corresponding transport block size (TBS), (iii) TA alignment, (iv) the number of available hybrid automatic repeat request (HARQ) processes, and (v) transmit power control information. DCIs are transported through PDCCH identified by a radio network temporary ID (RNTI) issued by the network.

The scheduler also controls the MCS for optimized spectral efficiency. The primary DLSCH and ULSCH unit is one resource element given a particular MCS. However, the schedule needs measurement reports about the downlink and uplink channel conditions to make proper allocations. In downlink, the scheduler uses the following inputs to allocate resources for downlink transmissions: (i) The amount of data to be transmitted, (ii) type of data, (iii) available radio resources, (iv) downlink channel conditions measured through CQI, Moreover, (v) QOS constraints, such as user throughput, drop packet rate, and so on. CQI is collected at eNB from all UEs in a cell. The UE determines CQI to be reported based on measurements of the downlink reference signals. Typically, a Signal-to-Noise-and-Interference Ratio (SNIR) to CQI mapping algorithm is used to calculate CQI values. The reported CQI is a number between 1 (worst) and 15 (best) indicating the most efficient MCS that would give a block error rate (BLER) of 10% or less. UE CQI reports have two formats: (i) wideband CQI reports based on measurements across the entire downlink allocation, (ii) sub-band CQI reports based on measurements across preconfigured subsets of the downlink allocation.

In uplink scheduling, the eNB evaluates every TTI, which UE requires allocated uplink resources. The scheduler uses the following inputs to allocate resources for uplink transmission: (i) buffer status report (BSR) indicates the amount of data available for transmission in the uplink buffers of an UE, (ii) power headroom report (PHR) indicates the additional return power available at an UE, Moreover, (iii) CQI based on uplink physical reference signals: SRS based, periodically sent by UEs, and PUSCH based, calculated from actual transmitted data. With this information, eNB gets an idea about how much more bandwidth UE is capable of using in a subframe.

2.4.3 Radio Measurements in LTE

In LTE, both the UE and the eNB are required to perform all necessary measurements to characterize properly radio channels and ensure the transmission's QoS. Measurements are used for a variety of purposes including cell selection, scheduling, handover, power control, and positioning services [165]. The periodicity of these measurements is related to the UE connectivity state inside the network, namely:

- Active UEs are connected to one eNB and have one or more on-going traffic flows currently.
- Connected UEs are associated with one eNB but without any ongoing traffic sessions. However, UEs periodically transmit control messages based on the cell-specific configuration.
- Disconnected UEs are in the vicinity of the eNB, but currently not ON (or in airplane mode).

In our research, we consider active and connected UE states, which allow us to perform localization inside the network as described in Chapter 7.

UE Measurements

UE measurements, which are of our interest for positioning applications and can be supported by standard LTE UE manufacturer:

- RSSI is defined as the average total received the power of resource elements over the full bandwidth. RSSI measurements vary with LTE downlink bandwidth and the subcarrier activities (e.g., data transfer activity).
- Reference Signal Received Power (RSRP) is a power measurement defined by REs of CRS spread over the allocated bandwidth. The power per RE is determined from the energy of the useful part of the OFDM symbol, excluding the cyclic prefix.

- Reference Signal Received Quality (RSRQ) is mathematically defined as (N*RSRP)/RSSI, where N is the number of used RBs over which RSSI is measured. RSSI is the average total received the power of resource elements over the full bandwidth.

Typically, a UE reports RSRP and RSRQ back to the serving eNB (reported inside the Radio Resource Control layer (RRC)). Other measurements, such as reference signal timing difference (RSTD) or UE-Rx-Tx time difference, require special hardware and are not supported by all UEs.

eNB Measurements

We present only those eNB radio measurements of interest, which are supported by all the standard eNB implementations:

- Timing advance is used to compensate for the propagation delay as the signal travels between UE and eNB. When an UE transmits a PRACH preamble, out of 64 preambles, the eNB responds on the PDSCH channel with TA and other parameters. A preamble length in the frequency domain is equivalent to 1.08 MHz. Hence, the TA resolution is limited to 0.52 μs.

- RSSI measurements over the entire downlink allocation (called wideband RSSI) or over a preconfigured subset of the downlink allocation (called subband RSSI) are used by the eNB as an indicator for uplink power control algorithms. RSSI measurements are performed per TTI on PUSCH, PUCCH, and SRS allocations.

- SNIR is a measure of CQI in both subband and wideband cases. CQI measurements are valuable for link adaptation and scheduling.

- Uplink Time Difference of Arrival (UTDoA) was introduced in Release 11. The method uses the time difference measurements based on SRS at several eNBs (or location measurement units). Uplink positioning methods have no impact on the UE implementation.

2.4.4 Processing-Time Restrictions

This research considers LTE FDD, which requires signal processing with short delays at the subframe level in the PHY layer. The most critical processing deadline is imposed by the Hybrid Automatic Repeat Request (HARQ) protocol on the MAC layer. HARQ, which is a retransmission protocol between eNB and UE, states that the reception status of every received subframe has to be reported back to the transmitter. In LTE FDD-based networks, the HARQ Round Trip Time (RTT) equals 8 ms. The transmission time of a LTE time unit (subframe) T_{subframe} is equal to 1 ms. Each packet received at subframe k has to be acknowledged through an Acknowledgment (ACK) or Negative Acknowledgment (NACK) at subframe k+4, which

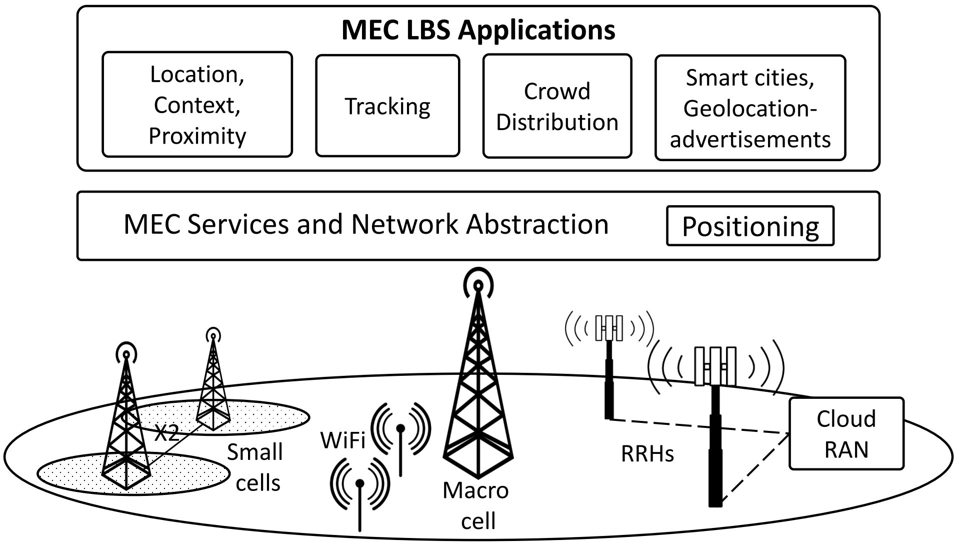

MEC LBS Applications
Location, Context, Proximity
Tracking
Crowd Distribution
Smart cities, Geolocation-advertisements
MEC Services and Network Abstraction
Positioning
X2
Small cells
WiFi
Macro cell
RRHs
Cloud RAN

Figure 2.14: Location based services in MEC.

in turn has to be decoded at the transmitter before assembling subframe k+8. This procedure is because acknowledgments control the retransmission mechanism (i.e., the transmitter has to decide on retransmitting the previously sent information or transmitting a new chunk of data). The total delay budget is therefore considered as 3 ms at the eNB.

2.4.5 Mobile-Edge Computing

A promising paradigm to meet the demand to move from a simple bit pipe to a smarter pipe is via the deployment of mobile edge computing (MEC) [145] or smarter edge nodes [89]. MEC is an emerging technology aiming to provide low-latency and high-bandwidth service environment *by deploying mobile applications at the edge of the network.* By moving application and content in close proximity to the UE, MEC offers a local cloud environment at the edge of the mobile network (hierarchically above eNBs) with direct access to real-time radio information (e.g. radio status, statistics) on an abstracted network topology. The network abstraction can be seen as a service layer (or middleware) that abstracts the low-level system functions and information through a set of APIs.

An example architecture of MEC is shown in Figure 2.14. The network abstraction is accessed by the positioning service to extract relevant radio signal information (e.g. received RSSI, estimated transmitted power P_{tx}, and timing advance from multiple communication points to determine the position of each UE. The area covered by the positioning service could be dynamically adjusted from a subset of the cells to all the cells per geographical region, radio access technology,

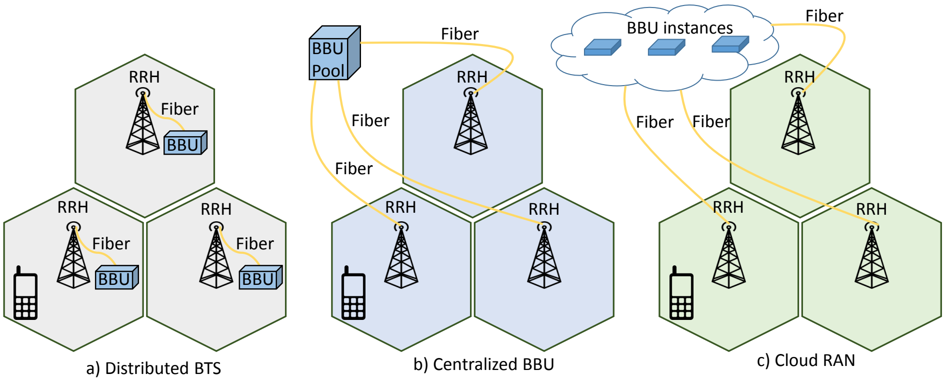


Figure 2.15: Cloud RAN evolution.

and/or provider basis. MEC is a complementary approach to the current and future cellular architectures [145, 89], such as heterogeneous and small cells networks (HetNet) and cloud radio access network (CRAN) with distributed Remote Radio Heads (RRHs).

An important MEC application to boost the performance of UE devices at the boundary between cells is Coordinated Multipoint (CoMP) in both downlink [135] and uplink [105] directions. The idea behind CoMP is that a UE receive almost equally power from neighboring eNBs at the edge of the cell between them [119]. Hence for downlink, the LTE network would coordinate downlink transmission from these eNBs to jointly transmit packets to a UE coherently or noncoherently. In a similar way, uplink CoMP is working, where a set of neighboring eNBs coordinate their reception to receive jointly transmitted packets from a UE.

2.5 Centralized-RAN (CRAN)

A significant problem emerges because typical service providers operate on a large scale. Current costs of building and operating a new infrastructure able to supply required data rates are superior to the revenue growth rate [50]. Therefore, service providers have to provide a high BTS density over large geographical areas by installing and keeping a large number of expensive integrated BTSs as shown in Figure 2.15-a. The problem will grow in the future with the trend of smaller and smaller cell sizes, e.g., picocells. Moreover, dedicated hardware BTSs have been designed to operate at the maximum load of a cell.

A new cost-effective RAN solution has to satisfy a multitude of requirements. First, it has to allow for fast upgrades and scaling satisfying the demand for quickly increasing and highly variable mobile traffic. Second, high capacity and network coverage have to be provided for reduced power consumption to provide a competitive mobile broadband service. Finally, mobile operators need to upgrade their network frequently and operate with multiple radio interfaces in a heterogeneous manner to meet ever-increasing amounts of mobile data traffic.

Centralized-RAN (CRAN) [138][85] as shown in Figure 2.15-b could be a solution to reduce costs and power consumption by sharing resources and exploiting load patterns of an individual geographical area at a given time (spatiotemporally load patterns). In networks with CRAN, Base-Band Units (BBUs) are no longer maintained alongside a BTS at a remote location. BBU and RRH are decoupled; the RRH remains at the previous location of the BTS , while the BBU migrates to a centralized processing pool. The CRAN solution allows the reaction to changes in user data traffic and mobility patterns. It can also allow for increasing spectral efficiency (data rates) by coordinated and joint signal processing. The current RAN architecture is not energy efficient because only 15–20% of all sites are loaded more than 50% of the total capacity [48]. One of the major benefits of a CRAN could be a perfect match between computational resources and hence power consumption with spatiotemporal traffic statistics over relatively large geographic areas. Moreover, since signal processing is centralized, it allows for more sophisticated joint spatiotemporal processing of radio signals, which could drastically increase spectral efficiency. This approach is considered by European projects such as Mobile Cloud Networking (MCN) [133], Sail [162], and iJoin [101].

2.6 Cloudification of CRAN (Cloud-RAN)

The next steps to lower the costs of running a CRAN rely on the successful adoption of virtualization and cloud computing technology allowing for a) migration from specific expensive hardware to general-purpose IT platforms, b) load balancing and c) rapid deployment and service provisioning. Virtualization and cloud-computing come at the cost of higher software complexity. The cloudification of the RAN is not new and the benefit of such an approach has demonstrated 71% of power savings in comparison to the existing system [102].

Cloudification of CRAN, to which we refer now as Cloud-RAN (c.f. Figure 2.15-c), is an interesting concept for both cloud providers and mobile telephony operators. There are many cloud computing paradigms such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) [142]. In the following, we concentrate on IaaS, which provides users with Virtual Machines (VMs) having processing capabilities, storage, network and other optional services to support various user applications. Running multiple VMs is accomplished through virtualization, which is a process that allows a single physical machine to act simultaneously as a few logical entities by using a software layer called “hypervisor”. There are multiple hypervisor solutions, such as XEN, KVM and LXC, each with its own advantages and disadvantages [183]:

- XEN is *Type I* hypervisor containing an abstraction layer between the physical hardware resources and VMs. XEN is loaded directly on the hardware and hence, it has close to native runtime processing performance. However, it requires updates in the OS kernel for compatibility.

- KVM, stands for Kernel-based Virtual Machine, is *Type II* hypervisor. KVM is loaded as an application in OS running on the hardware and hence, KVM contains two layers between the physical hardware and VMs: one with the underlying OS as another as the hypervisor. KVM is merged into the Linux kernel, but it has a slight performance degradation because of emulation.

- LXC is an application container that uses a new Linux feature called control groups (cgroups) to isolate groups of processes from each other while running on the same host. LXC offers less isolation than XEN and KVM but a lower overhead by using cgroups for direct access.

Hence, the difference between *Type I* and *Type II* hypervisors rest on the number of times that translation occurs between the VM and the guest operating system to the physical hardware resources. For heavy processes and critical real-time applications that run in a virtualized environment, such as cloud-RAN, it is required to have minimum virtualization processing overhead. Cloud-RAN might be executed on a public cloud platform, in which multiple VMs share computational and storage resources. In such environments (typically KVM-based), processing time deadlines cannot be guaranteed, while typical practices of cloud providers such as over-subscription of resources (e.g., processors) amplify this trend even further. In this research, we study in a brief cloud-RAN solutions due to its added value for deploying localization and tracking applications as discussed in more details in Chapter 6.

2.7 Software Defined Radio Systems

Cloud RAN is defined as a SDR system running in the cloud. We found some definitions that could describe a SDR system. The SDR Forum has defined a SDR system as a radio in which one or more of the physical layer functions are implemented in software. Primarily a SDR system consists of radio front-end implemented in hardware and signal processing modules implemented in software [129]. The radio modules to transmit and receive signals run on a dedicated hardware platform, such as Digital Signal Processor (DSP) processors or generic hardware platform such as General-Purpose Processors (GPP). An ideal approach would be to convert the analog signal to digits and perform all software processing under commodity hardware, like a PC machine. However, this approach is not possible or practical for wideband applications, which require high sampling rate and proportional processing power. To support a broad range of applications using the

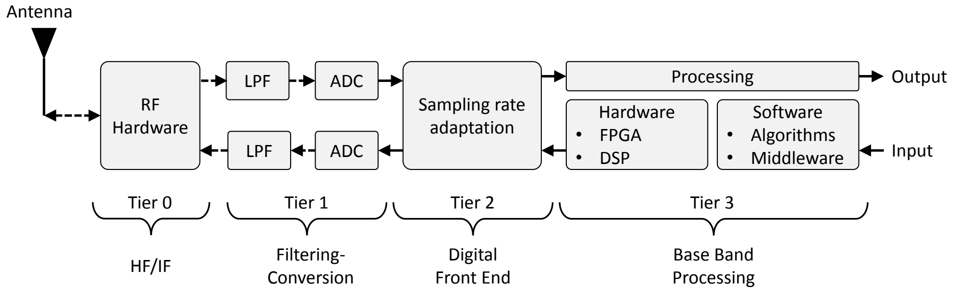


Figure 2.16: SDR generalized functional architecture

same SDR components, current SDR architecture is built on the multi-tier concept as illustrated in Figure 2.16 [71]. Tier 0 contains non-configurable radio components, which cannot be changed by software, such as antenna head. Tier 1 describes software controlled radio components with limited controllable functions, such as analog Low Pass Filter (LPF), Analog to Digital Converter (ADC), Digital to Analog Converter (DAC), Power Amplifier (PA), or Low Noise (LNA). A significant proportion of the radio is software configurable in Tier 2 components. These components are usually programmable, such as Digital Down Converter (DDC) inside an Field-Programmable Gate Array (FPGA). Tier 3 contains the target software modules with all required protocol stack layers. These modules can operate on dedicated hardware components, such as DSP and FPGA or on commodity machine with GPP units. Commodity hardware with GPP support has the advantage of fast prototyping compared to the dedicated FPGA approach. When the system bandwidth increases, such as in LTE up to 20 MHz, the challenge to perform all signal processing modules on a GPP-based machine (Tier 3) also increase.

Current SDR architecture brings a set of advantages to software developers, operators and end-users, such as (i) interoperability to support multiple standards through various configuration and multi-band radio capabilities, (ii) adaptability with faster updates towards new technology standards, (iii) flexibility to optimize between hardware, software, and user demands, and (iv) optimum utilization of hardware platforms and easier deployment.

2.7.1 Hardware

For a complete transmitter and receiver chain, we have to include hardware components with an analog radio and other relevant impairments that we would encounter in a real radio system. The list of SDR hardware vendors is growing quickly. The SDR hardware ranges from very expensive to very cheap transceivers. One might ask, which SDR hardware would be optimal to use? The answer depends on many factors amongst the application scenario of interest, target equipment, and total price.

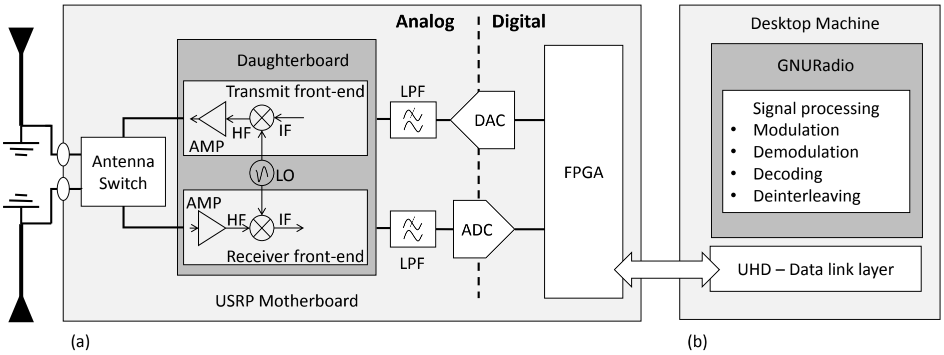


Figure 2.17: SDR block diagram based on USRP hardware

The most popular SDR platforms are the Universal Software Radio Peripheral (USRP) developed by Ettus Research [71] and the Wireless Open Access Research Platform (WARP) developed by Rice University [26]. Compared to the USRP, the WARP platform has more flexibility in terms of the number of daughterboards (up to 4 daughterboards with independent RF front-ends). The WARP platform contains different onboard implementations of the PHY and MAC layers, which gives it more processing capabilities than the USRP. However, the relatively high cost of the WARP platform compared to the USRP prevents us from adopting it in our research. The USRP [71] is one of the most popular, cost-effective and flexible platforms to host an SDR system. A block diagram of USRP hardware is shown in Figure 2.17-a. We are precisely using a set of USRP N210 devices equipped with one SBX daughterboard each (we call them N210 for simplicity) [71]. These devices operate over a very wide spectrum range (50 MHz - 6 GHz) and are, therefore, suitable for a broad range of wideband applications. The upper path in Figure 2.17-a marks the transmitting front-end (Tx), and the lower path marks the receiving front-end (Rx). Both paths can operate simultaneously inside the USRP daughterboard. For example, the received radio signal will be amplified before being converted from the High-Frequency (HF) to the Intermediate-Frequency (IF) band. Then it will be filtered using a LPF with a dynamic bandwidth up to 20 MHz. The filtered signal is then digitized inside the ADC, which has 14-bit precision for each In-phase and Quadratic (I/Q) samples. The ADC digitizes the signal with a *fixed sampling rate* equal to 100 Msps (Mega sample per second), but only the FPGA can process samples at this speed. The N210 FPGA changes the exported data to 16-bit samples when configured in 8-bit mode (8 bit I and 8 bit Q). It can otherwise use 32 bits per sample (16 bit I and 16 bit Q). Then, the FPGA export the resulted samples to the hosting machine over Gigabit Ethernet (GbE).

The USRP Hardware Driver (UHD) controls the USRP hardware. The UHD driver is an open-source software component providing access to the USRP from the host platform, as illustrated in Figure 2.17-b. Applications can use the USRP through the UHD API or, more likely, through the connectors provided for various

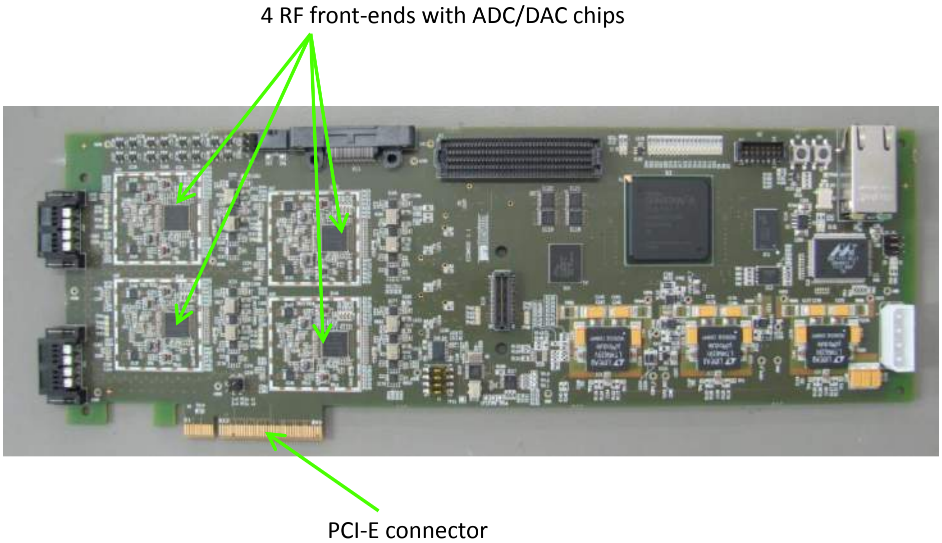


Figure 2.18: ExpressMIMO2 architecture [72]

platforms. The UHD provides access to multiple USRP devices in a homogeneous setup. It means that applications using UHD can benefit from various USRP capturing data at different locations. Up to our knowledge, there is no built-in mechanism to use a single USRP from several hosts simultaneously using the UHD driver.

Other SDR boards, such as the ExpressMIMO2 board developed by Eurecom [72], have been designed targeting unique radio technologies and applications. Some modules of the EXPRESSMIMO card are illustrated in Figure 2.18. The advantages of the ExpressMIMO2 board over the USRP N210 are (i) the support of four independent RF front-ends (equivalent to four daughterboards), (ii) MIMO support, (iii) and direct connection to PCI-E, which means more capability to transfer high data rates. Moreover, the sampling frequency of the ExpressMIMO2 board is fixed to 30.72 MHz, which is the same for LTE. Hence, the ExpressMIMO2 board is mostly dedicated to LTE-based SDR solutions. Exchanged samples between the hosting machine and the ExpressMIMO2 board is done through a particular Linux driver, called the **openair_rf.ko**, developed by Eurecom for the board.

In our research, we used the USRP N210 device as a RRH for GSM-based localization (c.f. Chapter 4) and the ExpressMIMO2 board as a RRH for LTE-based localization (c.f. Chapter 7).

2.7.2 Software

With the ever growing popularity of wireless communication technologies, the demand for highly innovative, flexible radio systems with a shorter time to market is increasing. The open source GNURadio (GR) framework offers most SDR general functionality for signal acquisition as shown in Figure 2.17-b [81]. GR is a *CPU-based implementation* mainly built using C++. GR comes with an extensive set of existing signal processing blocks, such as modulators, Fourier analysis, channel models. GR Python applications are capable of using C++ modules using the SWIG interface. SWIG is typically used to generate the 'glue code' to call C/C++ interfaces. Using this technique, Python can be used for high-level programming without any performance penalty. For this reason, Python applications are primarily used for flowgraph management or fast prototyping of blocks. GR does not provide out-of-the-box applications for specific radio communications standards (e.g., 802.11, ZigBee, and GSM), but it can be used to develop implementations of any band-limited communication standard.

There are different types of GR processing blocks, but they all share a set of characteristics [81]; they consume and/or produce data and process it continuously inside a *work* function. A GR processing block is a set of independent software tools and libraries. Connected blocks in GR exchange consumed/produced samples inside buffers. Samples are typed *gr_complex* in GR, but are in fact only a redefinition of the C/C++ `std::complex` data type. Complex numbers are represented as two consecutive single-precision Floating-Point Units (FPUs), thus taking 64 bits. GR processing blocks are connected through their input and output ports, and a block may have several input and output ports.

Distributing the workload of a GR process is equivalent to running blocks in parallel. The distribution process is done automatically when running a GR application: by default, each block is executed on its thread. A GR application thus takes advantage of multicore CPUs without requiring extra work from the programmer. While thread priority of each block can be pre-configured for performance tuning, the thread scheduling is solely managed by the operating system. The GR scheduler handles calling the *work()* function with optimal expected produced items (noutput_items), so there is no starving block in the flowgraph. The GR scheduler also handles buffer management so that there is no need for explicit synchronization inside processing blocks.

GR takes advantage of architecture-specific operations through the Vector-Optimized Library of Kernels (VOLK) machine [178]. The VOLK machine is a collection of GR libraries for arithmetic-intensive computations. Most of the provided VOLK implementations are using x86 Single Instruction Multiple Data (SIMD) instructions, which enable the same instruction to be performed on multiple input data and produce several results (vector output) in one step. However, the available processing resources might reach an end with the increasing width of the system spectrum. *A channelizer is a GR processing block used to split a wide-band spectrum into a set of equally-spaced channels.* In SDR systems as shown in Figure 2.20, two main components can be identified:

- A channelizer that receives a stream of digitized samples of a wideband analog signal (with real and imaginary components) and splits it into M equally-spaced (in the frequency domain) streams.
- A set of processing instances, e.g., GSM receivers, which can process the channelized streams in real-time.

If a machine is able to channelize a certain number of channels, it *does not* mean it has a remaining processing capacity to analyze them.

Architecture of the Polyphase FilterBank Channelizer

A Polyphase Filterbank (PFB) channelizer is an efficient technique for splitting a signal into equally spaced channels [87]. A PFB is mainly composed of two main modules: M Finite Impulse Response (FIR) filters (the filterbank) and an M-point Fast Fourier Transform (FFT). A basic implementation of the PFB is illustrated in Figure 2.19, which represents the PFB channelizer input with M FDD channels that exist in a single data stream and the resulted M Time Division Multiplexed (TDM) output channels. The fundamental theory of polyphase filters is the polyphase decomposition [126]. We discuss these two principal components of the channelization process.

FIR filters: are digital filters with a finite impulse response. The transfer function of FIR filter samples is expressed in Equation 2.1.

$$\mathrm{f}[n] = \sum_{k=0}^{N-1} h[k]x[n-k] \tag{2.1}$$

N is the number of FIR filter coefficients (known as filter taps) , $x[k]$ are FIR filter input samples, $h[k]$ are the FIR filter coefficients and $\mathrm{f}[n]$ are FIR filter output samples.

A **FFT** is a fast way to calculate the Discrete Fourier Transform (DFT) of a vector x. The FFT operation transforms the input signal from the time domain to the frequency domain. The M-point FFT transfer function is expressed in Equation 2.2.

$$y[k] = \sum_{m=0}^{M-1} \mathrm{f}[m]e^{-2\pi jmk/M} \tag{2.2}$$

where $\mathrm{f}[n]$ is the input and $y[k]$ is the output. The time it takes to evaluate an FFT on a computer depends mainly on the number of multiplications involved. FFT only needs $Mlog_2(M)$ multiplications.

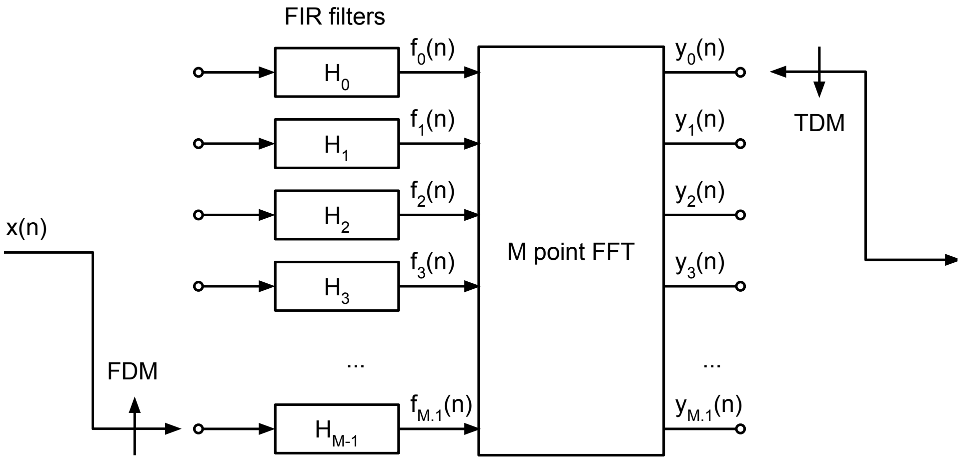


Figure 2.19: PFB channelizer.

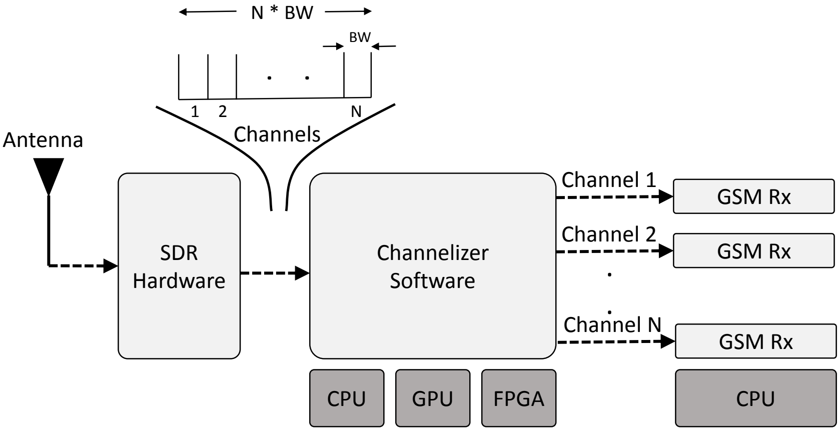


Figure 2.20: The concept of wideband channelizer.

SDR Implementations of the PFB Channelizer

PFB channelizer implementations are widely available for various platforms [159, 177, 6]. The channelizer performance varies based on the underlying processing hardware architectural [6, 29]. Each processing hardware has its advantages and challenges. While FPGA-based approaches can provide the required performance for a broad range of wideband applications, they lack the run-time flexibility of GPP-based approaches [29].

CPU-based implementations are widely used due to their natural availability of commodity computers and easy memory management. However, the CPU approach requires specific architecture optimization and does not reach Graphic Processing Unit (GPU) or FPGA processing performance [155]. Hence, we consider in Section 4.3.1 optimizing an existing GR CPU-based channelizer using advanced CPU instructions [63].

Recent studies have shown the processing advantages of using a GPU-based channelizer. While discrete GPUs are becoming cheaper, they set several challenging to battery-based hardware due to their high power consumption [6]. GPUs are relatively easy to take advantage of due to the various efforts of several GPUs vendors on general-purpose computing on graphics processing units (GPGPUs). GPUs are not only able to perform a fixed set of graphics-related operations, but also suited for general-purpose processing as we would expect from general-purpose processors. There are several technologies available to do GPGPU, such as CUDA programming model created by NVIDIA or OpenCL framework [185]. At the time of our research, there was no open-source tool that implement a GPU-based PFB channelizer as a GR module. There was a project to add GPU support to GR [83], however, this project is deprecated. While the design of a PFB channelizer is known (c.f. Figure 2.20), the challenging task is to implement it efficiently on a GPU and integrate it with a complete system. The authors in [6, 7, 57, 109] show the possibility to perform a PFB channelizer with a GPU underlying processing unit, however:

- None of these solutions are provided as an open-source, whereas our proposed solution presented in Section 4.3.2 is published as an open-source [38].

- The performance of these solutions includes only the GPU processing time without addressing the added latency for data transfer between system memory (RAM) and GPU memory. Our proposed solution considers the data transfer latency in the performance measurement and proposed solutions to reduce it.

- Our proposed solution is implemented as a GR module for ease of integration with complete projects, such as our proposed GSM receiver.

SDR Applications

An enormous array of real-world radio applications are using GR, such as tracking satellites [174], radar systems [157], GSM networks [141], all in computer software. In GR, a radio system is represented by a graph similar to circuit connections in the hardware domain. To our knowledge, until now, there is no passive tool that can offer comprehensive capturing and interpretation of GSM uplink signals without collaboration with end-users or network operators. We present in Section 4.2 our GSM receiver that overcomes the challenges of (i) synchronization with the end users in time and frequency, (ii) signal power recovery and (iii) message parsing. The two projects, which are most relevant to us, are Airprobe [13] and OpenBTS [141].

Airprobe. Airprobe is an open-source project aiming to intercept and parse signals in the GSM downlink using an USRP device [13]. A schematic diagram of

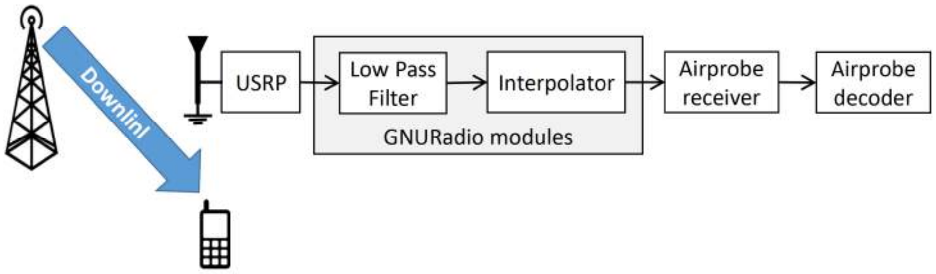


Figure 2.21: Airprobe system architecture [13]

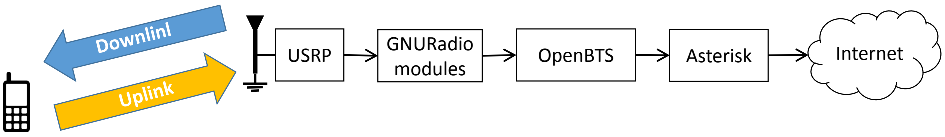


Figure 2.22: OpenBTS system architecture [141]

Airprobe architecture is illustrated in Figure 2.21. Airprobe contains (1) USRP source, LPF and Interpolator modules (taken over from GR) and (2) airprobe-receiver and airprobe-decoder. The *airprobe-receiver* module contains the signal processing chain, which demodulates, deinterleaves and decodes the captured GSM signal. The resulting bits are parsed to meaningful parameters inside the *airprobe-decoder* module. The *airprobe-receiver* follows the GSM MD power-up scenario (discussed in Section 2.3.3) to apply frequency and time synchronization and reconstruct messages on C0 channels, such as BCCH information. It is important to mention that *Airprobe supports single channel interception*. In late 2014, the Airprobe project was fully integrated with GR community under the name of *gr-gsm*.

OpenBTS. OpenBTS is another open source project that emulates GSM functionality [141]. It implements the GSM Air interface of a BTS up to layer 3. Since an USRP attached to OpenBTS acts as a real BTS, MDs can connect to it following the same standard procedure as in a typical GSM network [64]. OpenBTS uses Asterisk software [28] to connect calls to users. A schematic diagram of OpenBTS architecture is illustrated in Figure 2.22. There are many security risks associated with the OpenBTS operation and the encryption methods used that allow breaking into the network. The Man-in-the-Middle (MiTM) attack is proposed by different researchers to get a full insight into the identity of target users (required for localization) and their radio settings for localization [111]. In the MiTM attack, an attacker secretly relays and possibly alters the communication between two parties without either party knowing that the channel between them has been compromised. However, this approach breaks the law of the network operator and privacy rules of users and hence, will not be used in our work.

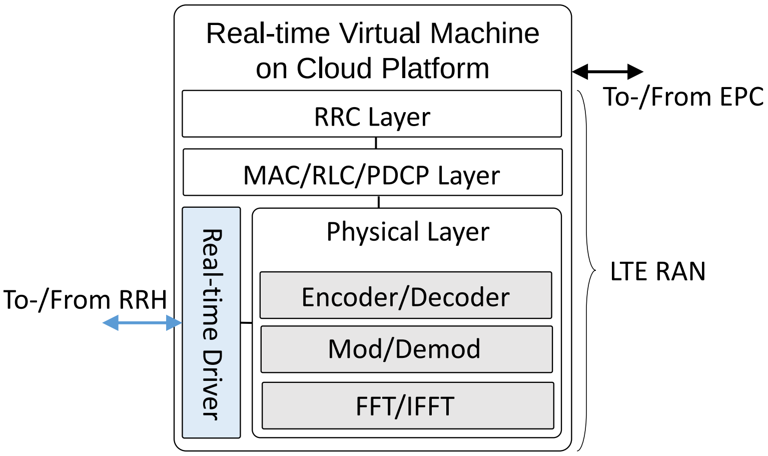


Figure 2.23: OAI eNB architecture.

Open Air Interface. Several software implementations of the LTE eNB already exist, e.g., a) Intel solutions based on a hybrid GPP-accelerator architecture aiming at a balance among a flexible IT platform, high computing performance and good energy efficiency [164], b) Amarisoft LTE solution featuring a fully-functional pure-software LTE eNB [1] and c) OpenAirInterface (OAI), developed by EURECOM, which is an open-source SDR implementation of LTE including both the RAN and the EPC [73]. In this research, we consider only the OAI implementation. The OAI emulation platform is an open-source software, which implements (up to know a large set of) the LTE 3GPP Release-8 standards [72]. OAI provides an eNB wireless protocol stack containing the PHY, MAC, Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP) and RRCs as well as Non-Access-Stratum (NAS) drivers for IPv4/IPv6 interconnection with other network services [140]. Regarding the PHY layer, OAI implements the signal processing chain for OFDMA (Downlink) and SC-FDMA (Uplink) as described in Sec. 6.2. OAI uses optimized C code for Intel architectures (using MMX/SSE3/SSE4 instruction sets for signal processing) for efficient numerical operations. Figure 2.23 illustrates a schematic diagram of OAI eNB implementation with a particular focus on LTE PHY layer modules, such as FFT/IFFT, modulation/demodulation, and encoding/decoding [140].

Note that the OAI eNB code contains calibration parameters (obtained through manual balancing of the ExpressMIMO2 board), such as total amount of power gain for RSSI measurements. However, OAI eNB code does not yet contain such calibration for time measurements, such as the timing advance.

Distributed Processing

For wideband applications, such as a wideband GSM receiver, if a machine is able to challenge a certain number of channels, it might not have remaining processing

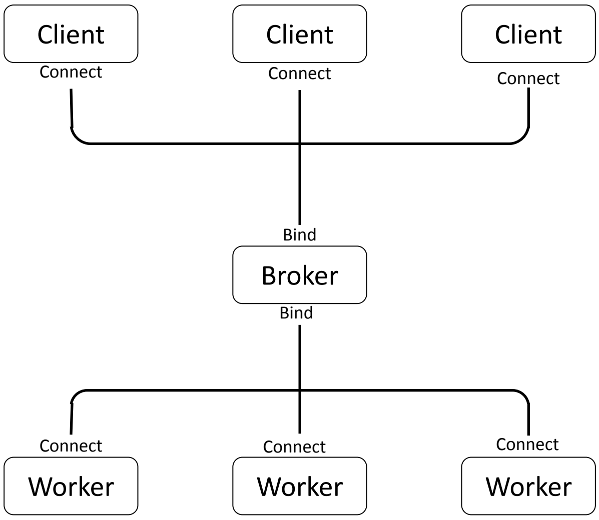


Figure 2.24: Architectural overview of ZMQ messaging library.

powers to analyze the channelized streams. Thus, channelized streams have to be distributed to other processing machines, mostly over an IP network. For such a scenario, we require to have a new layer that distributes the load of the system to support an arbitrary number of channels. For this task, we will use the ZeroMQ (ZMQ) messaging library [93]. Compared to competitors, such as ActiveMQ [5] and RabbitMQ [156], ZMQ is faster, sometimes significantly so. It is, therefore, more suited for highly intensive messaging or synchronization systems. ZMQ is not point-to-point protocol, it defines an overall topology of a distributed system as shown in Figure 2.24. In this figure, clients publish their needs for processing resources to the broker and the workers publish their abilities for processing to the broker also. The broker is the stable point in the network that map requests with available resources. Mapping to our wideband GSM receiver, clients are the channelized streams from the PFB channelizer and workers are distributed machines over an IP network with the GSM-receiver tool.

ZMQ relies on asynchronous communication. In the asynchronous model, the ZMQ can abstract all connection setups, reconnect, and reduce latencies (asynchronous adds non-blocking mode to queued messages).

2.8 Active and Passive Localization

In previous subsections, we presented alternatives to GPS that can be used by radio-based indoor localization systems. Previous survey papers, such as [123, 49], classified indoor localization systems based on the used radio technology and localization algorithms. However, another classification can be done based on the device involvement in the localization process. Depending on the participation of the tracked MD (indoor or outdoor), two types of localization systems can be distinguished [12]. In active systems, the MD communicates with several ANs and the localization process is a collaborative effort. Examples of such systems are the use of WiFi APs or GSM BTSs as ANs. On the contrary, in a passive system the ANs are hidden to the target MD and only overhear the target's radio transmissions. In this thesis, we make use of both passive localization systems, as described in Part I, and active localization systems, as described in Part II.

2.8.1 Active Localization

A localization system is described as active if the localization process is performed by the tracked device [122, 113] or the network side [160, 194]:

- In **network-based** localization, several fixed measurement units inside the network are responsible for collecting users' radio measurements [79, 134]. Then, an individual server inside the network performs the localization process.

- In **user-based** localization, the MD collects radio measurements from several transmitters of fixed known locations. The localization process can be done inside the MD directly. If the device does not have the capability to compute the location or for battery life consideration or does not have the knowledge of exact transmitters' locations, it is possible to send the radio measurement back to the network to estimate its location, which is called **user-assisted** localization.

In both cases, cooperation is required either from end users or the network operator. Active localization systems have the advantage of accessing various types of measurements and information required for localization, such as radio resource allocation and P_{tx}. The implementation of multi-user localization is relatively straightforward in active systems, which have the unique identity of each participating MD in the localization process.

2.8.2 Passive Localization

The passive localization term in most published literature refers to Device-Free Localization (DFL) [188, 154, 147] . DFL makes use of changes in RSSI patterns (transmitted by a set of signal transmitters) to detect the presence of *moving* objects. For example in intrusion detection scenarios, DFL uses only WiFi AN with periodic beaconing without any MD participating in the localization process. While DFL does not require any MD carried by the tracked person, it set several challenges towards localizing static people, identifying a localized person, localizing multiple individuals [188]. Moreover, DFL requires a high deployed density of transmitter and receiver nodes. Moreover, DFL beaconing processing make the system visible.

However, our passive localization approach makes use of an active MD carried by the tracked person. On the contrary to active systems, this approach does not require the participation of the communicating parties but relies on overhearing radio signals and their subsequent processing. While passive systems offer invisibility to the network operator and localized people, they set several challenges to signal capturing, localization algorithms, and security [19, 18].

2.9 Localization Systems

2.9.1 Localization in Cellular Networks

Cellular positioning is of tremendous interest to network operators. The GSM networks apply cell-ID approach with MD's TA for positioning correction, i.e, enhanced cell-ID (ECID). However, the large cell radius ($\sim$35 Km) and low spectrum bandwidth (BW = 200 KHz for downlink and uplink channels) offered limited positioning accuracy in the range of 500m without using any additional radio metric (or localization algorithm) [134]. Recall that TA resolution is *inversely* proportional to the system bandwidth. The ECID (Cell-ID + TA) idea was picked up again by the WCDMA networks. The WCDMA standard uses a bandwidth of 5 MHz for each downlink and uplink channel, which results in more accurate TA correction [165]. With the increasing number of LTE-capable MDs, the direction becomes clear to develop location-based services based on LTE [134, 18, 79]. 3GPP LTE specifies a wide range of positioning methods with specific functional and performance requirements [75, 165]. Also, LTE assigns uniquely developed Positioning Reference Signals (PRS) and dedicated Serving Mobile Location Centers (SMLC). Advanced methods like the Adaptive Enhanced Cell-ID (AECID) fuse geographical cell descriptions with fine-grained power or time radio measurements [165, 134] Radio measurements are collected at the user equipment or eNB terminals for UE-assisted (within 3GPP Release 9) and network-based (within 3GPP Release 11) methods. LTE network-based positioning has several advantages, such as (i) it does not require any additional signaling interface, (ii) it has minimum footprint impact on network capacity, and (iii) minimum requirements for target devices.

However, collected information from cellular or WLAN networks has to be processed using some localization algorithms to estimate locations of MDs. RSSI-based localization algorithms can be classified into two categories. The first category explores the relation between signal strength and distance using explicit proximity assumptions or propagation models. The second category uses fingerprinting where a signal strength database is filled with measurement records during an offline phase, and the location is estimated by fitting the online measurements of this database [75]. Our need for an accurate real-time system without expensive offline phase and less calibration costs make the first approach, such as range-based algorithm, more favorable [170].

2.9.2 Input Parameters to Localization Algorithms

After describing an overview of radio technologies and the potential SDR implementations, we focus on radio measurements providing input to localization algorithms. Ideally, localization algorithms should be only aware of radio measurements, a device identity and not of any related personal data. On the one side, the device identity is a technology dependent parameter. For example, in Radio Frequency Identification) (RFID), the tag identity is the frequency [37]. In Bluetooth and WiFi technologies, the MAC address is used to identify devices [59]. In cellular technologies, multiple identities are tagged with a MD simultaneously for different communication flows, such as Temporary Mobile Subscriber Identity (TMSI) and International Mobile Subscriber Identity (IMSI). Indeed, a localization system is required to decode the radio message till it reaches a valid identity to support certain services. On the other side, radio metrics is hardware dependent and can be almost provided for any communication technology. Different hardware design, e.g., antennas, are built with different capabilities to serve a particular type of application scenario.

A radio-frequency technology can provide feedback on multiple parameters related to signal reception, which can be used for localization. Here we present few radio metrics that are commonly used by SDR localization systems:
Received Signal Strength Indicator (RSSI): RSSI is a measurement of signal power at the receiver. As a signal leaves the MD, it attenuates, meaning that the signal power drops. The signal attenuation differs between different environments with respect to distance. For example, in free-space with LOS communication links, the signal attenuation is less with respect to the distance than indoor. If the transmitted power P_{tx} is known, RSSI can be used to estimate the distance the signal has traveled using a particular propagation model [84, 112]. However, RSSI poses an exciting challenge indoors because it is affected by various parameters such as NLOS reception, multipath propagation, opening and closing of doors, mobile objects, temperature and humidity variations [30, 77]. Hence, RSSI measurements vary at the receiver and may deviate from the theoretically anticipated values.

Time (Difference) of Arrival (TOA/TDOA): Time of Arrival (TOA) measures the signal propagation delay from the transmitter to the receiver. The separating distance between the transmitter and the receiver represents the propagation delay multiplying by the signal propagation speed ($\simeq$ speed of light) [126]. However, this approach requires precise knowledge of the signal transmitting timestamp, which requires high-resolution time synchronization between the transmitter and the receiver [182]. To overcome this issue, Time Difference of Arrival (TDOA) was proposed to measure the time difference of a received signal at two time-synchronized receivers or vice versa [191]. The timestamp resolution is in direct relation to the signal bandwidth [46, 130]. Moreover, multipath propagation produces multiple copies of the signal at the receiver, which poses a challenge to detect the shortest propagation path in NLOS and multipath environment.

Angle of Arrival (AOA): Angle of Arrival (AOA) is a measurement of the radio propagation direction at the receiver with a reference orientation. One standard approach to measure AOA is using an antenna array [186]. The measurement accuracy is proportional to the size of the antenna array, which also increases the cost. However, AOA can also be measured using multiple fixed directional antennas or one directional antenna rotating with a constant angular speed. However, the resulted accuracy does not justify the high deployment cost and complexity of this approach, in particular for a medium to large deployment [125]. Furthermore, such approach might require device compatibility in heterogeneous setups. Moreover, AOA inaccuracy increases indoors with multipath propagation and NLOS reception.

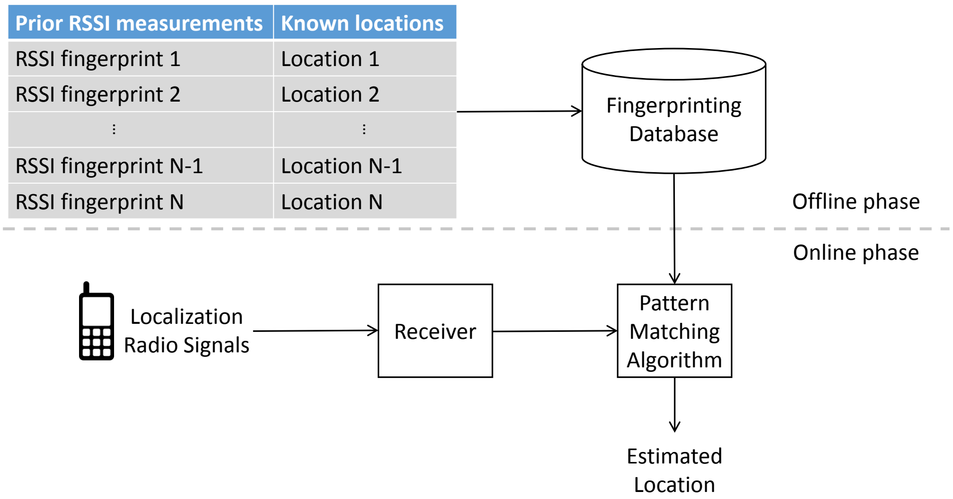
Because existing radio metrics might not achieve comparable performance in all environments, a hybrid approach that combines different communication parameters is considered to meet a broad range of accuracy and performance requirements [172]. Nevertheless, this approach requires a complex processing, and its deployment overhead and cost remain high.

RSSI is widely and natively used in wireless networks for link quality estimation and power control. It can be obtained from most off-the-shelf wireless network equipment. The natural RSSI acquisition and relatively low deployment costs of the system compared to other radio metrics, such as TDOA or AOA, makes RSSI-based localization a preferable solution [12]. Hence, our research orientation is based on RSSI analysis, modeling, and algorithm development.

2.9.3 Fingerprinting-based localization

Three main approaches are commonly used for indoor localization using RSSI measurements: (i) fingerprinting, where an offline recorded radio map of the target environment is leveraged to infer locations using a matching algorithm with online measurements, (ii) propagation based algorithms, which calculate distances to the target device using a path loss model, and (iii) proximity algorithms, which relies on relations among RSSI measurements at different ANs to estimate locations.

Fingerprinting algorithms are also called radio-map-based approaches. The idea behind this approach is to characterize the area of interest through an offline training phase [171, 110] as shown in Figure 2.25. Fingerprinting-based localization is composed of two main phases. In an offline phase, multiple ANs record the



Prior RSSI measurements	Known locations
RSSI fingerprint 1	Location 1
RSSI fingerprint 2	Location 2
⋮	⋮
RSSI fingerprint N-1	Location N-1
RSSI fingerprint N	Location N

Figure 2.25: Fingerprinting localization algorithm.

RSSI values of a test device for a large set of sample points in an area to creating a database or a radio map. The vector of RSSI values for a single location is called the fingerprint of that point. In an online phase, the RSSI measurements from an unknown MD are compared to the recorded database to find the location with closest match [31].

The challenging part of fingerprinting approaches is the matching algorithm because it determines the accuracy and response time of the system [30, 149]. There are two main matching techniques, deterministic and probabilistic [123, 110]. In the deterministic procedure, such as the nearest neighbor algorithm, the distance in the signal space is calculated from the online measurements and each location point vector in the database. In the probabilistic technique, additional probabilistic information, e.g., based on movement history or floor plan, is cooperated in the process.

Fingerprinting approaches reduce the deployment cost by leveraging the existing infrastructures. Though fingerprinting solutions tend to perform well indoors, fingerprinting presents two significant *drawbacks* that limit its applicability for certain location-based applications: flexibility and fast deployments.

- Fingerprinting requires extensive manual calibration efforts and time to build the offline database.
- Environmental changes (outdated database) have an adverse impact on the localization accuracy. The assumption that the indoor space remains consistent from the offline phase to the online localization phase does not hold real-world environments. Thus, there is a need to update the database periodically. Such updates require re-calibration, which is time-consuming and may be challenged by limited physical access to the area of interest.

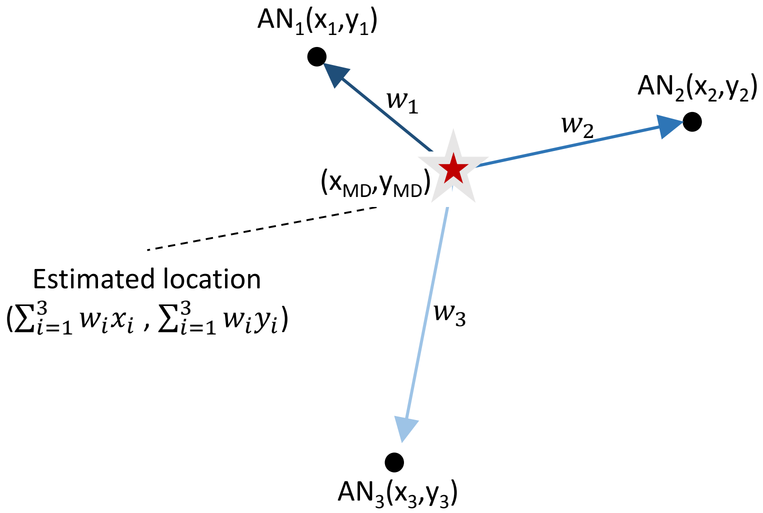


Figure 2.26: Proximity-based localization: ANs pull the estimated location propotional to RSSI value.

- The accuracy of the estimated position depends strongly on the size of the database, i.e., the number of fingerprints, and the number of RSSI measurements per fingerprint [143].
- RSSI fingerprints suffer from many indeterministic parameters, such as the orientation of MD, body shadowing, and multipath propagation.

2.9.4 Proximity-based Localization

Proximity is the simplest method for localization. It might also be the only available method without knowing the exact MD's radio settings and the environment layout. In proximity approaches, the position of the MD is not calculated from explicit distances to the ANs but relative to AN positions. In this process, both connectivity and RSSI information can be used [61]. For connection based localization, the finite communication range of the ANs can be used to derive a coarse position of the target user. The decrease in signal strength with distance can be further exploited to obtain a finer position as illustrated in Figure 2.26. The accuracy of proximity solutions is proportionally linked to the density of ANs over the sensed area and underperforms in sparse deployments. This type of localization, however, requires less calibration effort than range-based or fingerprinting-based solutions.

Since any set of ANs that do not lie on a single line can be seen as forming a polygon, geometric calculations can be applied to determine a point that can represent the target's location. The most common approach used in proximity methods is the centroid, i.e., the center of mass of a polygon [92]. The set of ANs that form the polygon can be denoted by C_{AN} and the criteria to include an AN differ, e.g., all nodes are in hearing range to the target device, the k nodes with the highest RSSI, and so on. The main drawbacks of the basic centroid approach are:

- ANs should be uniformly spread over the area of interest.
- The system accuracy is proportionally related to AN deployment density, potentially leading to high costs.
- Given the same set C_{AN}, the estimated coordinates $(x_{\text{est}}, y_{\text{est}})$ are the same for all real locations $(x_{\text{r}}, y_{\text{r}})$. The reason is that no RSSI values are considered in the centroid calculation.

Hence, the Weighted Centroid (WC) method is proposed in multiple studies [36].In the WC approach, each AN contributes with its coordinates $(x_i, y_i)$to the location estimate calculation proportionally to its RSSI level received from the target MD according to Equation 2.3.

$$(x_{\text{est}}, y_{\text{est}}) = \left(\frac{\sum_{i=1}^{N_C} w_i * (x_i, y_i)}{\sum_{i=1}^{N_C} w_i} \right) \forall \ \text{AN}_i \in C_{\text{AN}} \tag{2.3}$$

w_i is the weight associated with the i^{th} AN and N_C is the size of the set C_{AN}. In the Linear Weighted Centroid (LWC) approach, the weight of AN_i is proportional to its RSSI level, i.e., $w_i = \text{RSSI}_i$ [114]. Hence, the AN with the highest RSSI will pull the centroid strongest towards itself. The Adaptive Weighted Centroid (AWC) has been proposed [32] to achieve a higher accuracy by using adaptive weights. Weights are calculated as a function of d_i, the initial estimated distance between the target device and the ith AN, and a constant g, usually depending on the propagation model, i.e., $w_i = 1/(d_i)^g$. In real environments, the constant g is determined during periodic transmissions between ANs. Typical proximity-based solutions, such as [114, 179, 193], are faced with several challenges:

- Beacon transmissions in the GSM spectrum are not permitted due to licensees restrictions. Moreover, standard RAN implementation in cellular networks does not allow beaconing between BTSs. This approach might only be possible using the unlicensed band, such as for WiFi-based solutions.
- Most of proposed solutions rely on absolute RSSI for weight calculations. However, RSSI is a function of transmitted power. Instantaneous knowledge of this parameter is hard to obtain unless it is set to a known constant, which is not the case for WiFi and cellular networks because of power control algorithms.

We present in Section 4.5 an innovative approach that overcome these challenges and localize MDs without any offline calibration processes or interaction with the end-users or network operators. The proposed solutions leverage differential RSSI information. A set of real-indoor experiments was conducted to validate the proposed solutions.

2.9.5 Range-based localization

Range-based algorithms use ranging metrics, such as RSSI, to estimate the distance d between a transmitter (T_x) and a receiver (R_x) based on a channel model. The idea behind this technique is to use radio metrics from a particular MD as dependent location parameters. The lognormal shadowing model [158, 15, 121], expressed in Equation 2.4, is an example of a radio model that translates the received power P_{rx} to distance d.

$$P_{rx}(d) = A - \underbrace{10\,\alpha \log\left(\frac{d}{d_0}\right)}_{\text{path loss}} - \psi, \tag{2.4}$$

α is the path loss exponent and $\psi \sim N(0, \sigma^2)$ reflects the log-normal shadowing effect with zero mean and σ standard deviation. Coefficient A is related to T_x and R_x antenna gains, transmission power P_{tx} and power loss at a reference distance d_0, which has to be determined experimentally. The use of channel models for localization faces several challenges. First, since the wireless transmission medium varies over time, the α and ψ parameters can only be obtained statistically over an extended period [146]. Second, P_{tx} and the T_x antenna gain for a specific MD are difficult to know, unless provided by the MD. Third, P_{rx} depends on the MD's hardware specifications and antenna orientation [60].

In power-based localization systems, the trilateration technique is a standard approach used to estimate the location of an object in 2D/3D [132, 39]. Trilateration algorithms use distance measurements d_i at three/four different reference points with known location, AN = $[x_i, y_i]$ in our context. If the number of AN is more than three/four, the method is called multilateration [184, 180]. The distance d_i estimated by RSSI is presented as a circle with a radius around the AN$_i$ in 2D setups as illustrated in Figure 2.27. The intersection of three AN radius provides a point or an area representing the possible location of the target MD. Note that RSSI acquisition and distance calculation can be measured at the network or the MD side. Here we discuss the linear least square method used to minimize the localization error. Ideally, we want the absolute distance error ϵ between the transmitting Sensor Node (SN) (with an unknown position) and the receiving SN (with

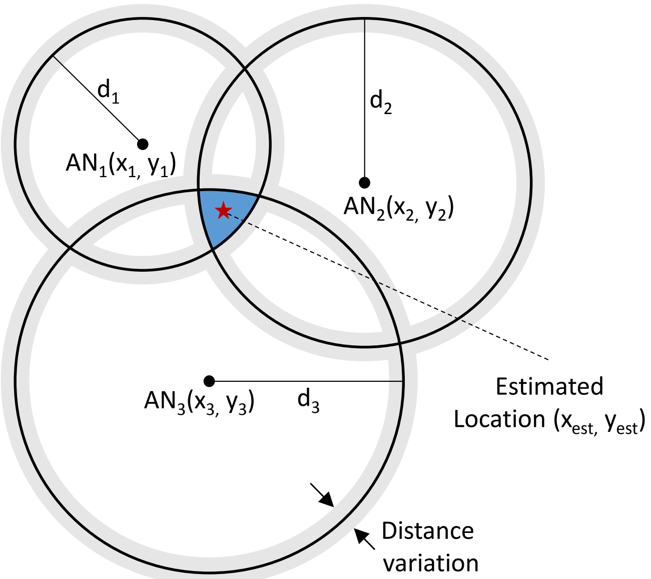


Figure 2.27: Localization area provided by trilateration approach

known position) to be zero.

$$\epsilon =| \sqrt{(x_i - x_{\text{est}})^2 + (y_i - y_{\text{est}})^2} - \widehat{d}_i |= 0 \tag{2.5a}$$

$$(x_{\text{est}}^2 + y_{\text{est}}^2) + x_{\text{est}}(-2x_i) + y_{\text{est}}(-2y_i) - \widehat{d}_i^2 = -x_i^2 - y_i^2 \tag{2.5b}$$

$\widehat{d}_i$ is the distance estimation to SN$_i$ with (x_i, y_i) coordinates using Equation 2.4 and $(x_{\text{est}}, y_{\text{est}})$ are the unknown coordinates of the target SN. Subtract the Equation 2.5b from the previous ones with AN$_k$ = $(x_k$,$y_k)$ to get rid of quadratic terms $(x_{\text{est}}^2$+$y_{\text{est}}^2)$.

$$2x_{\text{est}}(x_k - x_i) + 2y_{\text{est}}(y_k - y_i) = \widehat{d}_i^2 - \widehat{d}_k^2 - x_i^2 - y_i^2 + x_k^2 + y_k^2 \tag{2.6}$$

Combining Equation 2.6 from different SNs and re-arrange, this would result in a linear system expressed in Equation 2.7.

$$\boldsymbol{H}\widehat{\boldsymbol{x}} = \boldsymbol{b} \tag{2.7}$$

$$\widehat{\boldsymbol{x}} = \begin{bmatrix} x_{\text{est}} \\ y_{\text{est}} \end{bmatrix}, \boldsymbol{H} = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) \\ 2(x_k - x_2) & 2(y_k - y_2) \\ \vdots & \vdots \\ 2(x_k - x_{k-1} & 2(y_k - y_{k-1}) \end{bmatrix}, \text{ and}$$

$$\boldsymbol{b} = \begin{bmatrix} \widehat{d}_1^2 - \widehat{d}_k^2 - x_1^2 - y_1^2 + x_k^2 + y_k^2 \\ \widehat{d}_2^2 - \widehat{d}_k^2 - x_2^2 - y_2^2 + x_k^2 + y_k^2 \\ \vdots \\ \widehat{d}_{k-1}^2 - \widehat{d}_k^2 - x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 \end{bmatrix}.$$

In the case of RSSI-based localization, the estimated distance $\widehat{d}$ is disturbed by channel noise, obstacles, and other shadowing effects. Hence, Equation 2.7 might not have a unique solution. Therefore, the position of the unknown node can be calculated by minimizing $\| \boldsymbol{H}\widehat{\boldsymbol{x}} - \boldsymbol{b} \|^2$ Here we can use the least squares method to get a solution of Equation 2.7 as expressed in Equation 2.8.

$$\widehat{\boldsymbol{x}} = \left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{b} \tag{2.8}$$

The main challenge facing trilateration approach is to have an accurate propagation model to estimate the channel losses accurately with respect to distance [51, 90, 121]. Similar to the proximity-based approach, many of these solutions rely on exact knowledge of MDs' radio settings. However, in a realistic scenario, this information is not available neither at the network side nor fully at the user side. This hard task, as described in Section 2.10, is affected by random occurrences that make the distance estimation vary in an unpredictable ways. Hence, circles might not intersect, and no singular solution exists.

2.9.6 Hybrid Localization Techniques

To deliver more accurate indoor localization system, various literature studies propose advanced hybrid solutions, which combine multiple localization techniques (hybrid-technique) or radio metrics (hybrid-metric). One choice of such a hybrid-technique solution is the combination of fingerprinting localization and TOA method using UWB signals [107]. This approach aims to reduce the calculation complexity of fingerprinting localization. Fingerprinting approaches require offline stage to create the database and TOA approaches require precise transmission timestamps of device packets. Hence, both fingerprinting and TOA localization do not fit the passive requirement of our system. Moreover, the authors in [107] consider the use of UWB signals in NLOS conditions and an Additive White Gaussian Noise (AWGN) channel. These considerations are considered as an oversimplification of reality. A more advanced hybrid-technique approach that combines fingerprinting with range-based localization is proposed in [115]. This solution relies on a sparsely populated fingerprinting database and benefits from radio propagation models to improve localization. However, this solution is still obliged to a time-consuming calibration process of fingerprints.

Hybrid-metric techniques that do not depend on fingerprinting are also increasing in popularity. Those solutions obtain multiple radio metrics, such as RSSI, TOA, TDOA or AOA from a single signal source and apply ranging or triangulation techniques to derive the distance or angle between transmitter and receiver. The different outputs are then combined to improve localization estimates [4, 192, 117]. Such hybrid approaches tend to perform well in comparison with conventional indoor localization systems. However, combining parameters from the same radio signal gives rise to a particular problem. Radio metrics from a single Radio Interface (RI) are highly correlated. As such, if metrics of a radio signal are impaired, all metrics' quality gathered from this signal will be affected by multipath propagation and NLOS reception. Therefore, we expect these approaches to have a lower performance or even diverge in real indoor environments.

Our proposed solution presented in Chapter 5 is a hybrid-network localization approach. More specifically, the proposed solution relies on RSSI measurements from multiple RIs active at the same time, such as GSM and WiFi, or LTE and WiFi. The proposed solutions are based on (i) probabilistic approach that combines RSSI measurements from different RI before being fed to the localization algorithm and (ii) another probabilistic approach that combines estimated locations from different RIs after the localization algorithm.

2.9.7 Tracking using Kalman Filters

Moving UEs poses a challenge to the proposed localization algorithm. Various tracking algorithms have been proposed in the literature to cope with RSSI changes, such as Kalman filters [181, 75]. Kalman Filter (KF) was proposed by Rudolf Kalman in 1960 to removes noise and other inaccuracies from a set of measurements. KF is a well-known solution for prediction for RSSI-based localization [148, 166, 104, 168, 181]. It is known that KF provides the minimum mean square error estimate of state variables of a linear system where the state is assumed to Gaussian distribution. One of the main advantages of KFs is the computational efficiency in the implementation using only matrix and vector operations on the mean and covariances of Gaussian input. In our research, we use KF to filter out RSSI or location measurements from noise and other inaccuracies. The input to the KF is real measurements $\mathbf{z}_k$ and the output is KF estimations $\mathbf{x}_k$, k is the state. We present in this section a brief highlight on the Kalman filtering approach, detailed derivation of the Kalman filter can be found in [150]. The two main equations of the KF are presented as below:

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \tag{2.9a}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{2.9b}$$

Equation 2.9a shows that any estimated measurements $\mathbf{x}_k$ can be evaluated using its previous value $\mathbf{x}_{k-1}$, a control signal $\mathbf{u}_k$ and processing noise $\mathbf{w}_k$. Note that the control signal is optional and might not exist in a model. Equation 2.9b tells that any input measurement $\mathbf{z}_k$ is a linear combination from its estimated measurement $\mathbf{x}_k$ with measurement noise $\mathbf{v}_k$. $\mathbf{A}_k$ is the state transition matrix, which relates the previous state at $k-1$ to the current state at k. $\mathbf{B}_k$ is the control matrix, which relates the input control signal $\mathbf{u}_k$ to the estimated measurement $\mathbf{x}_k$. $\mathbf{H}_k$ is the observation matrix, which relates estimated measurement $\mathbf{x}_k$ to the real measurement $\mathbf{z}_k$. $\mathbf{w}_k$ and $\mathbf{w}_k$ are considered independent Gaussian distributions with $\mathbf{Q}_k$ and $\mathbf{R}_k$ covariance, respectively.

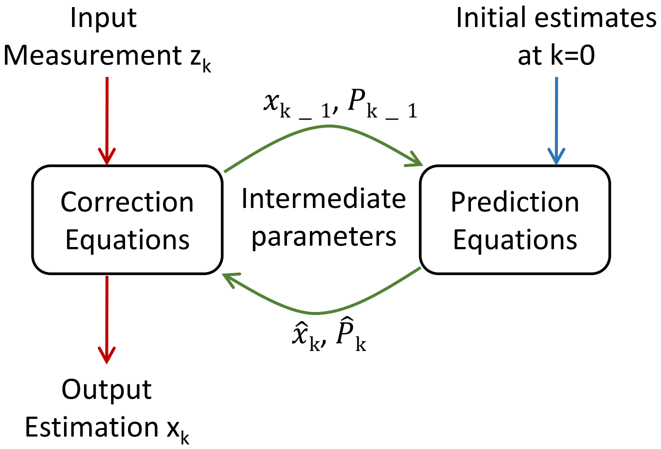


Figure 2.28: Iterations in Kalman filter

The filtering algorithm consists of two iterative phases as shown in Figure 7.14: a prediction phase and a correction phase. Equations in each phase are applied to each k^{th} measurements. The prediction phase contains equations that use the current estimation to predict a priori estimation for the next time step. The correction phase contains equations that use the real input measurements to correct the prior estimation of the prediction phase. For the first iteration $k = 1$, the prediction phase has to be initialized with input parameters representing the state at $k = 0$.

The specific equations for the prediction phase are presented in Equation 2.10.

$$\hat{\mathbf{x}}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k \tag{2.10a}$$

$$\hat{\mathbf{P}}_k = \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}_k \tag{2.10b}$$

$\hat{\mathbf{x}}_k$ is a priori state estimate at step k and $\hat{\mathbf{P}}_k$ is a priori estimation error covariance at step k. Equation 2.10a shows the prediction step for an estimation measurement and Equation 2.10b shows the prediction step for the error covariance. In the first iteration $k = 1$, $\mathbf{x}_{k-1}$ and $\mathbf{P}_{k-1}$ are provided as an input. Both prior information in Equation 2.10 are used in the correction phase, which contains the following equations:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{R}_k^{-1} \tag{2.11a}$$

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k) \tag{2.11b}$$

$$\mathbf{P}_k = (I - \mathbf{K}_k\mathbf{H}_k)\hat{\mathbf{P}}_k \tag{2.11c}$$

I is the identity matrix. Equation 2.11a is used to calculate the Kalman gain $\mathbf{K}_k$, which is considered as weights for the measurement residual $(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k)$ as shown in Equation 2.11b, which represent the k^{th} KF estimation $\mathbf{x}_k$. Finally, Equation 2.11c shows the calculation of the error covariance $\mathbf{P}_k$.

2.9.8 Accuracy Metrics

To evaluate the performance of the localization and tracking algorithms described in previous sections, a set of metrics are available, such as error mean μ, standard deviation σ and root mean square (RMS). These metrics show how well the ground truth and estimated position are matched (i.e., the distance d_n between estimated and ground truth positions). μ quantifies the amount the bias of localization errors as illustrated in Equation 2.12.

$$\mu = \frac{1}{N}\sum_{n=1}^{N} \underbrace{\sqrt{(x-\widehat{x_n})^2 + (y-\widehat{y_n})^2}}_{\text{distance } d_n} \tag{2.12}$$

N is the number of estimated locations over a period T and $(\widehat{x_n}, \widehat{y_n}) \;\; \forall \;\; n \in 1:N$ are x,y-coordinates of the estimated positions. σ quantifies the amount of scatter or dispersion of estimated locations as illustrated in Equation 2.13.

$$\sigma = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(d_n - \mu)^2} \tag{2.13}$$

However, RMS includes both bias μ and scatter σ metrics together as illustrated in Equation 2.14 [35]. It is clear that RMS is greater or equal to the average localization error μ.

$$\text{RMS} = \sqrt{\mu^2 + \sigma^2} \tag{2.14}$$

2.10 Channel Modelling

Accurate indoor localization algorithms, such as range-based algorithms, require accurate modeling of the target environment. Radio signals are harder to reconstruct indoors than outdoors. Indoor environments differ significantly from outdoor setting due to the smaller dimensions and the many more obstacles in the signal path. This results in different propagation scenarios, including reflected, diffracted, LOS or NLOS as illustrated in Figure 2.29. These obstacles can be part of the indoor construction, e.g., walls and doors, as well as individual objects such as furniture. Also, the indoor environment can change radically with the simple movement of people, opening and closing of doors, and so on. For these reasons, deterministic models with constant parameters are not often used. In the commonly used log-normal shadowing model, the Path Loss Exponent (PLE) is a critical parameter. It measures the rate at which the RSSI decreases with distance in a particular propagation environment [128]. An accurate knowledge of the PLE is required to obtain an accurate estimation of the separation distance with its corresponding RSSI measurement. In this section, the PLE is considered as an *unknown constant* in the deployment area. However, in indoor and dense urban environments, the relation between received signal strength and distance is extremely

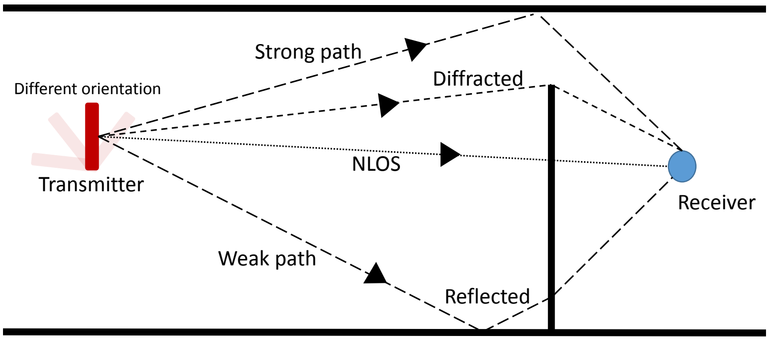


Figure 2.29: Different types of radio signals in typical indoor environment.

hard to model due to severe multipath fading and NLOS conditions. Furthermore, offline calibration is not applicable to the most of real-time scenarios (e.g. geo-advertisements, personalized pricing). A realistic scenario is an online estimation of the PLE from a limited and usually a small set of RSSI measurements, without any prior knowledge of actual value.

Characterizing the indoor radio channel has been an active research area dating back to the early '90s, e.g., [88]. Meanwhile, standardization organizations and research institutions have defined several indoor propagation models for different applications [10, 14, 43, 153]. Most of these models originate from the Free Space (FS) model with some slight modification according to the target scenario. The free space Path Loss (PL) model, or called the Single-Slope (SS) model for indoor environments, can be formulated in the logarithmic scale as in Equation 2.15.

$$PL_{FS} = PL_0 + 20log_{10}(d) + 20log_{10}(f_c) \tag{2.15}$$

PL_{FS} expresses the PL value in dB unit, PL_0 is the PL at a reference distance, f_c and d are in GHz and meters unites, respectively. Equation 2.15 shows that higher frequency signals, such as WiFi at 2.4 GHz, tend to suffer from more loss introduced by distance and obstacles for its poor abilities of diffraction and reflection compared to lower frequency signals, such as GSM at 900 MHz. This fact is important for hybrid-network localization algorithms, which combine RSSI measurements from different RIs operating at different frequencies; due to the different behavior of radio signals with distance. However, for *ease of description*, most propagation models, inherited from Equation 2.15, *omit* the frequency part [44]. The SS model expresses that the PL increases with the distance at a fixed rate *in all directions*. This assumption is a simplification of reality where the PLE is path dependent. Given indoor propagation paths are not homogeneous, the Multi-Slope (MS) model considers the PLE as a location dependent variable, as expressed in Equation 2.16 [53].

$$PL_{MS} = PL_s + 10\alpha_s 10g_{10}(d), \quad d_{s-1} \leq d \leq d_s, s \in 1, 2, ... \tag{2.16}$$

where PL_s and α_s are the reference PL and PLE corresponding to the sth distance range from d_{s-1} to d_s. However, the MS-model requires additional calibration to estimate α_s and PL_s compared with the SS-model. Other forms consider the indoor layout as the only factor in their PL-model without the distance component, such as the Attenuation Factor (AF)-model expressed in Equation 2.17 [53].

$$PL_{AF} = PL_{base} + A_F \tag{2.17}$$

PL_{base} can be PL_{SS} or PL_{MS}, A_F is the attenuation factor caused by obstacles like walls and floors as expressed in details in the Multi-Wall-and-Floor (MWF)-model expressed in Equation 2.18 [53].

$$A_F = \sum_{p=1}^{P} \sum_{k=1}^{K_p^w} A_{pk}^w + \sum_{q=1}^{Q} \sum_{k=1}^{K_q^f} A_{qk}^f \tag{2.18}$$

A_{pk}^w is the attenuation caused by the kth wall of type p, K_p^w is the number of walls of type p, and P is the number of wall types. We apply the same terminology to the floor summations containing A_{qk}^f, K_q^f, and Q, respectively. MWF-model parameters require extensive field measurements, which involves intensive costs of both time and manpower. However, it is hard to achieve all these parameters whose accuracy have a substantial impact on the MWF-model. What matters more is that the penetration loss also changes with frequency. The signal with higher frequency tends to be more attenuated. However, the MWF model can also be added to the SS-model to include the distance and frequency factors.

These channel models and channel parameters mentioned above are widely used in the literature. However, they are just representative of a general view of some communication system and are not related to any particular scenario (radio technology and indoor layout). There are many other factors, which influence the PL, such as antenna heights, antenna orientation, the mobility of MDs, and so on. However, for the indoor scenario (more especially our research context of indoor offices), some factors can be negligible, such as antenna heights and MDs mobility. The antenna height and the speed of the MD have a limited impact on indoor deployment. To obtain realistic PL-model of a particular scenario (indoor layout and radio technology), a calibration process must be performed based on empirical measurements of the target situation. For a realistic performance of an indoor localization system, the calibration process must be done online and in real-time or quasi-real-time to overcome the dynamic characteristic of the indoor environment.

We propose in Chapter 7 an iterative approach that estimate the PLE parameter blindly and use real online measurements from target UEs, i.e., we do not exploit the PLE parameter in an offline stage. The proposed solution does contain a beaconing step and does not have any prior knowledge of the target environment and subsequently any calibration cost. The proposed solution relies on a robust statistical algorithm to eliminate contaminated RSSI measurements due to the environment or contaminated distance measurements by the iterative approach.

2.11 Outlier Detection and Mitigation

The modeling accuracy of the environments and consequently the localization accuracy depend highly on the robustness of the receiver against contaminated radio signals; we call them outliers. Classical estimators can withstand a moderate number of outliers [94]. However, in dense indoor environments when the number of outliers increases, e.g., due to shadowing and multipath propagation, robust estimators must be used to improve localization accuracy [42].

2.11.1 Indoor Narrowband Radio Propagation

In this section, we briefly discuss indoor propagation characteristics of narrowband radio signals, such as GSM. A scenario of indoor radio propagation is shown in Figure 2.29. The channel impulse response in the time domain, as illustrated in equation 2.19, is affected by the environment and the location of objects around the transmitter and the receiver [158].

$$h(t) = \sum_{k=0}^{L_p-1} \alpha_k e^{-j\theta_k} \delta(t - \tau_k) \tag{2.19}$$

L_p is the number of multipath components. α_k, θ_k, and τ_k are the amplitude, phase, and delay of the k^{th} path, and $\delta(t)$ is the Dirac delta function [2]. A radio system with bandwidth B treats multipath signals that arrive within a period $\tau < 1/B$ as a single component at the receiver antenna. Hence, signals in narrowband systems, such as in GSM, are subject to multipath effects more often than signals in wideband systems, e.g., ultra-wideband technology. In narrowband systems, $h(t)$ can be divided into two main components:

- Short-term variation $h_{\text{st}}(t)$ (fast fading) is caused by fast changes, constructively or destructively, in the received signal power. These variations are due to changes in the phase of the received signals from different paths.
- Long-term variation $h_{\text{lt}}(t)$ (slow fading) is caused by obstructed objects that attenuate the signals passing through them and by the signal path loss due to the distance between transmitter and receiver.

The received signal $x(t)$ can be completely described as the convolution of transmitted signal $s(t)$ and $h(t)$ as illustrated in equation 2.20:

$$x(t) = \underbrace{h_{\text{lt}}(t) * s(t)}_{\text{considered-normal signal}} + \underbrace{h_{\text{st}}(t) * s(t) + n(t)}_{\text{nondeterministic variation}} \tag{2.20}$$

$n(t)$ denotes additive white Gaussian noise. The total received power in narrowband channel measurements can be characterized by the probability density function. It includes (i) the received power of the considered-normal signal including propagation influences and (ii) nondeterministic fast power variation with additional channel noise. The latter component can be as strong as the former component. To increase the accuracy of a localization system, it is required to estimate only the considered-normal signals without considering the nondeterministic variations. This estimation will then be fed into the localization algorithm.

2.11.2 Estimators of Radio Metrics

Wireless systems, which adopt signal strength or timing metrics for localization, consider high metrics variation as an indication of outliers [47]. NLOS reception, multipath propagation, and abnormalities caused by incorrect hardware calibration are some of the most important sources of outliers in indoor environments. Considering RSSI-based localization systems, two important parameters are widely used to quantify the RSSI variation: (i) μ as a measure of signal amplitude and (ii) σ as a measure of signal dispersion for a given set of RSSI measurements. There are many estimators for the μ and σ parameters [98]. Let E_n be an estimator for the μ parameter, and D_n be an estimator for the σ parameter, where n is the number of independent RSSI measurements, denoted as $X_n = (\text{RSSI}_1, ..., \text{RSSI}_n)$. The key issue for accurate localization techniques is to minimize the influence of outliers on the estimator output.

Classical Estimators

Classical estimators are considered sensitive towards outliers because they treat all the samples equally. The numeric average, i.e., the sample mean $\widehat{\mu}$, and the sample standard deviation $\widehat{\sigma}$, are examples of classical amplitude and dispersion estimators. Their mathematical representations are given by the following equations:

$$\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \text{RSSI}_i, \tag{2.21}$$

$$\widehat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\text{RSSI}_i - \widehat{\mu})^2} \tag{2.22}$$

Another estimator is the Maximum-Likelihood Estimator (MLE). It maximizes the likelihood (or minimizes the negative log-likelihood) of signals with a given distribution function $f(x)$ [108]. MLE estimates the signal parameters ($\widehat{\mu}_{\text{MLE}}$ and $\widehat{\sigma}_{\text{MLE}}$) proportional to the probability of a signal measurement inside the measurement set as illustrated in Equations 2.23 and 2.24.

$$\widehat{\mu}_{\text{MLE}} = \arg\min_{\mu} \sum_{i=1}^{n} -\log\left(f(\text{RSSI}_i - \mu)\right) \tag{2.23}$$

$$\widehat{\sigma}_{\text{MLE}} = \arg\min_{\sigma} \sum_{i=1}^{n} \log(\sigma) - \log\left(f\left(\frac{\text{RSSI}_i - \widehat{\mu}_{\text{MLE}}}{\sigma}\right)\right) \tag{2.24}$$

MLE is considered as the most traditional method for deriving statistical estimators. However, MLE estimators require accurate distribution functions $f(x)$ that model the behavior of RSSI measurements X_n without outliers, which is not feasible in indoor environments as described in Section 2.11.1. If the chosen distribution is misspecified, MLE becomes sensitive to outliers [56]. With distribution misspecification, we mean the situation in which we do not have a distribution function that fits to the probability distribution of RSSI measurements. We chose the Gaussian distribution with the MLE estimator as a distribution function $f(x)$ due to its common usage in other studies.

Robust Estimators

The robustness of estimators determines the efficiency of the system in detecting outliers. To overcome MLE problems, M-estimators [98], the name is derived from maximum, suggest to generalize the role function $-log(f(x))$ in the MLE estimator by another function called the loss function $\rho(x)$. The main advantage of M-estimators is that no statistical models or prior information on the RSSI distribution in LOS/ NLOS are required. Hence, M-estimators overcome problems arising from the dependence on the RSSI distribution. The resulting MLE expressions for μ and σ parameters become:

$$\widehat{\mu}_{\text{M}} = \arg\min_{\mu} \sum_{i=1}^{n} \rho\left(\text{RSSI}_i - \mu\right) \tag{2.25}$$

$$\widehat{\sigma}_{\text{M}} = \arg\min_{\sigma} \left(\frac{1}{n} \sum_{i=1}^{n} \rho\left(\frac{\text{RSSI}_i - \widehat{\mu}_{\text{M}}}{\sigma}\right)\right) \tag{2.26}$$

Many choices of $\rho(x)$ function have been proposed. Nevertheless, no estimator can exclude outliers always correctly. However, when the number of outliers inside a measurement set increases, the Huber function ρ_H expressed in equation 2.27 [98], obtains the highest robustness against outliers than other choices [27].

$$\rho_H(x) = \begin{cases} x^2/2 & , |x| \leqslant b \\ b|x| - b^2/2 & , |x| > b \end{cases} \tag{2.27}$$

The choice of the tuning constant b plays a significant role in the performance of the M-estimators to outliers and the performance of false detection. The bigger the tuning constant is, the lower is the robustness of the estimator, and the better is the performance of false detection.

The Breakdown Value of An Estimator

Most open space localization techniques assume that the majority of measurements come from LOS communication [47], i.e., uncontaminated signals. Because NLOS signals have larger variation than LOS signals, NLOS signals can be treated as outliers. However, in indoor environments where multipath propagation and NLOS reception are dominant, the number of outliers become higher. However, how many outliers can an estimator detect? The breakdown value ε_n^* is a measure of the estimator robustness against outliers [99]. The maximum breakdown value for any estimator dictates the percentage of outlying measurements that causes the estimator to break down. A theoretical expression for the breakdown threshold for any amplitude $\varepsilon_n^*(E_n, X_n)$ and dispersion $\varepsilon_n^*(D_n, X_n)$ estimators are derived [99] and expressed in equations 2.28 and 2.29 respectively.

$$\varepsilon_n^*(E_n, X_n) \leq \frac{1}{n}\left\lfloor\frac{n+1}{2}\right\rfloor \tag{2.28}$$

$$\varepsilon_n^*(D_n, X_n) \leq \frac{1}{n}\left\lfloor\frac{n}{2}\right\rfloor \tag{2.29}$$

For a large number of signals, the maximum breakdown value given by equations 2.28 and 2.29 equals 0.5. If outliers represent more than ε_n^* of the total measurements, estimators cannot distinguish between considered-normal signals and outliers. However, the presence of a small portion of outliers, i.e. less than ε_n^*, can still have a distortion influence on the estimators' output.

2.11.3 Outlier Detection

Several proposals have been developed to filter out outliers before being fed into the localization algorithms [94]. There are three fundamental solutions for the problem of detecting outliers: (i) detection without prior knowledge of the observed signals, (ii) modeling both normal and outliers, and (iii) modeling only normal signals. Normal signals are the radio signals transmitted from the target MD. It might be possible to model regular and/or outliers of an active localization system in a *controlled environment*. However, in realistic indoor scenarios, factors that generate outliers are not deterministic. Moreover, calibration tasks for the localization process should be minimized. Hence, our focus is not to model outliers but rather to detect them in the observed signals. Depending on the number of observed variables, outlying detection techniques can be classified into univariate and multivariate methods.

Univariate Algorithms

The Z-value test was proposed [98] to measure the amount of dispersion by which the error of a signal ϵ_i varies. The signal error is defined as the difference between the instantaneous RSSI value and μ as expressed in equation 2.30.

$$Z_i = \frac{|\epsilon_i|}{\sigma} = \frac{|\mathrm{RSSI}_i - \mu|}{\sigma} \tag{2.30}$$

$|.|$ indicates the absolute value function. Signals with Z-value greater than a predefined threshold ϖ are considered as outliers. In many literature studies, ϖ is a constant equal to 3. It is hard to get the real signal's μ and σ values at a particular location due to specific site parameters such as floor layout and moving objects. Hence, a residual $\widehat{\epsilon}$ of an observed signal can be defined as the difference between the RSSI of that signal and the estimated amplitude of the total RSSI measurements [118], i.e., $\widehat{\epsilon_i} = \text{RSSI}_i - E_n(X_n)$. However, according to [118], residuals unlike errors do not all have the same σ value but depend on the estimator implementation. To overcome variations in the residual's dispersion, the studentized residual test S-test was introduced [190] with better theoretical properties towards outliers detection as follows:

$$S_i = \frac{|\widehat{\epsilon_i}|}{D_n^i(\widehat{\epsilon})\sqrt{1-h_{ii}}}, \quad 0 \leq h_{ii} \leq 1 \tag{2.31}$$

$D_n^i(\widehat{\epsilon})$ is the dispersion estimation of the residuals without the i^{th} measurement. h_{ii} is the diagonal entry of the hat matrix H, where H is defined as the orthogonal projection matrix onto the measurement space, i.e., $H = X_n(X_n^T X_n)^{-1} X_n^T$. h_{ii} is called the leverage of the i^{th} measurement, i.e., leverage of RSSI_i. Leverage is a measure of how far a signal deviates from the total set of measurements. Signals with leverage values closer to one are considered as potential outliers. The presence of $(1 - h_{ii})$ in the S-test denominator helps to magnify the S-value for potential outliers. However, the efficiency in detecting outliers using the S-test is determined mainly by the robustness of the estimators.

Multivariate algorithms

Multivariate outlier detection algorithms are harder than univariate ones. The simplest multivariate is the bivariate problems, where the input variable has only two components, such as RSSI and the corresponding distance. Bivariate problems have the advantage to be examined visually. The minimum covariance determinant (MCD) is a highly robust the estimator of multivariate location and scatter parameters. MCD substantially outperforms in terms of statistical efficiency other multivariant estimators such as the minimum volume ellipsoid (MVE) method [161]. The Fast-MCD method overcomes the high computation requirements and makes MCD available as a routine tool for analyzing multivariate data [100]. The MCD estimator consists of two steps. The first step is to search for a subset C of size h (with n/2 < h < n) whose covariance matrix has the smallest determinant. The second step is to detect outliers using the Mahalanobis distance (MaD) $d(i)$, as expressed in Equation 2.32 [94, 33].

$$d(i) = \sqrt{(\boldsymbol{z}_i - \boldsymbol{M})^T \boldsymbol{C}^{-1} (\boldsymbol{z}_i - \boldsymbol{M})} \quad \text{for} \quad i = 1, .., n \tag{2.32}$$

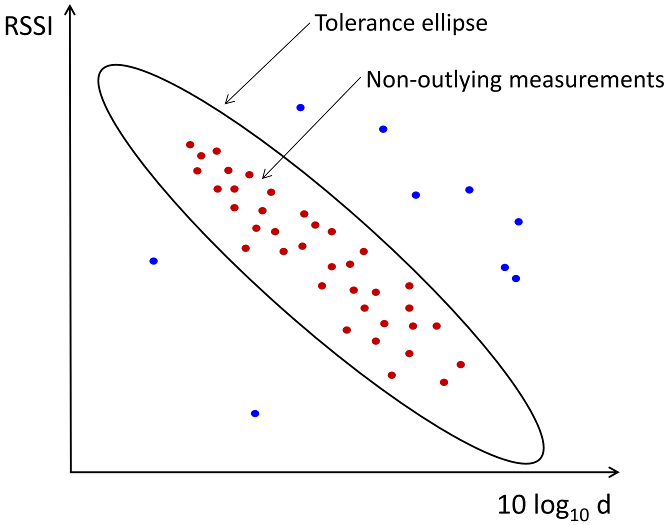


Figure 2.30: An example of bivariate MCD estimator

MaD is a measure of distance that accounts for the correlation between variables. $\boldsymbol{z_i} \in Z_n = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_n\}$ is a data set of p-variate measurements (2 in our case), $\boldsymbol{M}$ is the arithmetical mean and $\boldsymbol{C}$ is the covariance matrix. $\boldsymbol{z}_i$ is considered as an outlier if its MaD is greater than a certain threshold defined by the tolerance ellipse. The tolerance ellipse is defined as the set of points whose MaD = $\sqrt{\chi^2_{2,0.975}}$, the square root of the 0.975 quantile of the chi-square distribution with 2 degrees of freedom as defined [78] (c.f. Figure . The non-outlying measurements are $\boldsymbol{z_i} \mid d(i) \leqslant \sqrt{\chi^2_{2,0.975}}$. The MCD has its highest possible breakdown value of (n - h + 1)/n when h = [(n + p + 1)]/2, which yields to $\approx 50\%$. A good compromise between breakdown value (25%) and statistical efficiency (49%) is obtained when h = 0.75n (used in our experimental setup). The statistical efficiency of an estimator is a decision-theoretic measure of accuracy [52].

Part I

Passive Localization

Part I focuses on passive localization systems. Our target research aims to provide third-party solutions that are able to detect the presence and localize users indoors using active MDs. These systems do not have any relation to end users or network operators. Hence, each third party system has to have its own stand-alone components, such as sensor nodes, operating systems, localization algorithms, and databases. Currently, available mobile devices often support more than one radio technology, such as WiFi and the Global System for Mobile Communications (GSM). Moreover, the broad availability of both GSM and WiFi networks encourages research on using them to provide third party localization services. In Chapter 3, we only focus on WiFi, in which a set of SNs overhear transmitted messages from MDs. Later on, the idea was refined in Chapter 4, in which we use the GSM radio interface. It allowed us to identify pros and cons of the GSM passive system when compared to WiFi. Both WiFi and GSM implement their own protocol stack, which on smartphones comes with independent RF chips. Hence, simultaneous use of both interfaces is possible. For example, one can be simultaneously engaged in a call (GSM-based) and an online application use, such as Facebook, which requires connection to WiFi. Therefore, in Chapter 5, we develop a hybrid passive system that leverages information from both interfaces to improve the localization performance.

Chapter 3

WiFi-Based Localization Systems

3.1 Introduction

The vast deployment of indoor WiFi APs and the high adoption of WiFi capable devices among users, such as smartphones, make WiFi easily adopted by service providers for indoor localization. For example, in shopping malls or hospitals, WiFi APs are usually available and users typically are connected to them. In such places, our target is to provide a tool that *detects and localizes* users without interrupting their service or requiring them to run any additional software. Such a tool would help to detect frequent visitors and offer them special guidance or shopping offers. It would also help building administrators to focus their resources in places where people are mostly located. More specifically, our target is to install a third-party system (neither the network APs nor users' MDs) responsible for data acquisition and further processing.

Before moving to the localization step, it is important to study the parameters that affect our localization solutions. Our goal is to investigate which factors can be disregarded and which should be considered in the development of a localization algorithm. This chapter investigates (i) the impact of device's technical characteristics on radio measurements, (ii) manufacturing discrepancies among SNs, and (iii) the impact of device orientation. [1]. Without being exhaustive, we try to gain insights into the complex effects of each factor and the implications for indoor localization. Our purpose is to identify which factors we should consider and which we can disregard in the design of a localization system.

Our results show that technical factors, such as device characteristics have a smaller impact on the signal than multipath propagation. Moreover, we show that propagation conditions differ in each direction even in LOS conditions. We also noticed that WiFi and Bluetooth, despite operating in the same radio band, do not at all times exhibit the same behavior.

[1] It is ideal to study these factors in an anechoic chamber, which is able to remove multipath propagation and interference from any outside signal source. However, such a room is not available at our institute,

3.2 Sensor Setup

Technology: In order to observe the impact of various factors on the received signal, we deployed SNs, which can scan for transmissions on two interfaces - one for Bluetooth and one for WiFi (IEEE 802.11b/g). In the context of Bluetooth, we rely on the inquiry procedure described in Section 2.2 [97]. We prefer to work with the inquiry procedure due to several advantages:

- The RSSI reported by an inquiry procedure is not affected by power control and hence can be directly related to distance.
- An inquiry procedure can monitor a large number of target devices.
- We can gather measurements without requesting any privacy-sensitive information from the MDs.

In the context of WiFi, SNs overhear WiFi signals from target devices or beaconing of each other. A scanning SN can derive information on RSSI levels once it overhear these messages or any potential data messages.

Test-bed: All experiments were set up in an indoor office with dimensions 6.90x5.50x2.60m. A schematic is shown in Figure 3.2. The room contains desks, chairs and desktop machines. The SNs and MDs hang at 0.50m below the ceiling and 1.50m above the tables. Such a testing environment allows us to judge the relevance of the tested factors for a localization system under realistic propagation conditions. We use the Gumstix Board as a SN. The operating system of each SN is implemented on OMAP3530-based Gumstix Overo computer-on-module (c.f. Figure 3.1-a). As illustrated in Figure 3.1-b, Gumstix Boards are equipped with a WiFi interface (a TP-link adapter) compatible with the latest WLAN standards and allows packet monitoring. Data is stored on a micro-SD card, and an expansion board provides USB and Ethernet connectors for the development phase and power supply. The operating system is ADAM, a Linux-based distribution that contains a Bluetooth scanner module (scan_bt) and a WiFi scanner module (scan_wifi) to support both Bluetooth and 802.11b/g wireless communications [40]. Each SN consists of a shell script that can start, stop and restart both the WiFi- and the Bluetooth scanner. Both scanners collect information about the devices available on WiFi and Bluetooth, their MAC-addresses, and RSSI values. The collected information from SNs is sent to a central SN, which periodically sends the collected measurements to an outside database. Sensors' setup, software installation, and deployment were *done in a prior work to this research* [59].

Metrics: Our first challenge was to select the appropriate metric to compare performance. We consider four groups of parameters to characterize the RSSI, namely, instantaneous values, probability density function, mean and standard deviation, median and percentiles; as well as the response rate of a scan.

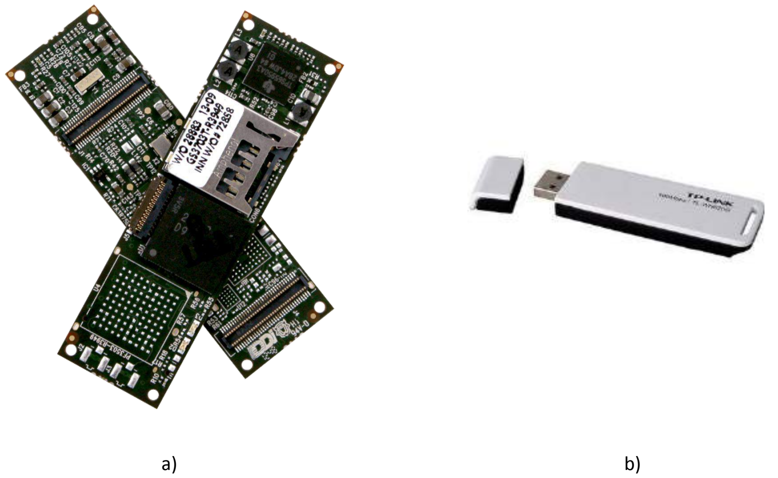


Figure 3.1: WiFi and Bluetooth SN module: a) Gumstix board, b) TP link adapter

3.3 Evaluation Parameters

Below we evaluate the impact of each of the four factors: technical characteristics, manufacturing discrepancies, direction-specific multipath, and channel reciprocity.

3.3.1 Evaluation A: Technical Characteristics

Evaluation Setup

RSSI, could additionally (on top of multipath effects) aggravate the problem of localization. To investigate how such differences affect the RSSI, we performed Experiment A. The Evaluation setup is shown in Figure 3.2. Three mobile phones by different manufacturers were placed (next to each other) at one and three meters away from the same SN. At each distance, measurements are gathered for 30 minutes, which corresponds to about 200 samples for WiFi and 400 for Bluetooth.

Instantaneous RSSI

Figure 3.3 shows the changes in time of the instantaneous RSSI of a Bluetooth signal from a distance of one meter. With instantaneous RSSI, we refer to a single momentary RSSI value. The sharp variations in the RSSI show how much this metric is affected by multipath. Therefore, relying on instantaneous RSSIs for localization can be misleading.

A better analysis would be based on metrics that can (partially) eliminate the impact of multipath. Multipath causes temporal, unpredictable RSSI variations. Evaluating a set of samples rather than a single value can isolate temporal changes and provide a more distinct main trend. We discuss the appropriate metrics in the coming three sections.

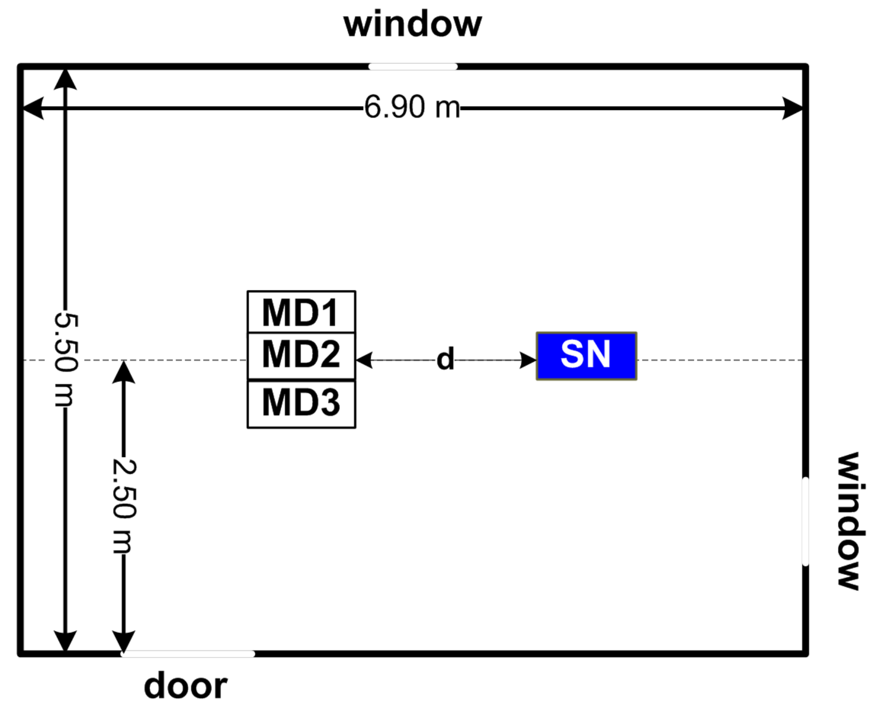


Figure 3.2: Experiment A: Setup.

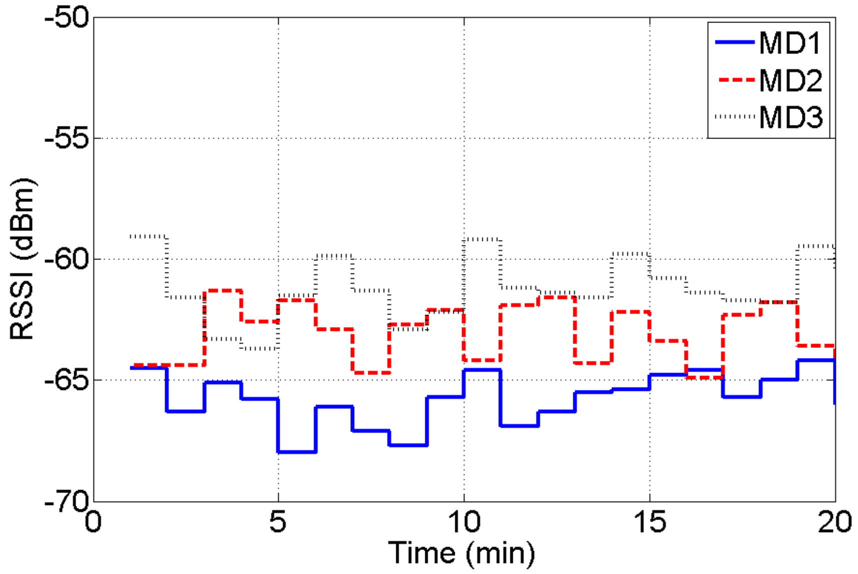


Figure 3.3: RSSI time variation of three MDs.

Empirical Histogram

The empirical histogram of the RSSI, constructed for each combination of MD and distance, is shown in Figure 3.4 for Bluetooth and in Figure 3.5 for WiFi. On the x-axis of a graph, we plot the RSSI values whereas the y-axis plots the number of RSSI measurements per unit interval (0.2 dBm) then normalized by the total number of RSSI measurements.

Although the histogram shapes are similar for the three MDs, the maximum RSSI value is not the same, suggesting that the impact of the technical characteristics of the device should not be underestimated. Further, as it can be expected, RSSI values are lower at three meters due to the larger path loss. Also, we notice that at distance one meter (upper row), the graphs are more compact. Whereas, at three meters (lower row), the histograms are wider, i.e., the set of observed RSSI values is larger. This observation can be explained by the stronger effect of multipath as distance increases. Another consequence of multipath is the slight asymmetry of the histogram with the longer tail towards lower RSSI values.

Differences between Bluetooth and WiFi are minor: WiFi signals have higher transmit power and subsequently stronger multipath components, which causes larger deviation of the RSSI signals, i.e., broader histogram shape. For MD2, we could not identify the reasons for the little effect of distance on its WiFi signal.

Mean and Standard Deviation

Although histograms and boxplots are very descriptive, they require the collection of many samples. Their use in a real-time localization system, where samples are evaluated every few seconds, is challenging. An easier to derive a set of metrics is the mean and standard deviation. The corresponding metrics for each histogram distribution in Figures 3.4 and 3.5 are shown in the upper left corner.

We note that the mean is often at 1-2 dBm offset from the median, see Figures 3.6 and 3.7. These differences are caused by the asymmetry in the histogram distribution - the mean and standard deviation take into account all samples, including outliers while the median excludes them. All other observations are consistent with previously made ones.

Median and Percentiles

An alternative to a histogram representation is a boxplot, which depicts a population's median, lower and upper quantiles, minimum and maximum, and outlier samples. Using Boxplots makes it easier to identify the main concentration of the RSSI values and how much the RSSI deviates. Another advantage of a boxplot is that outliers are visible; they are difficult to spot in a histogram due to their low probability.

The boxplots corresponding to the histogram distribution for both Bluetooth and WiFi are shown in Figures 3.6 and 3.7. Along with differences in the behavior

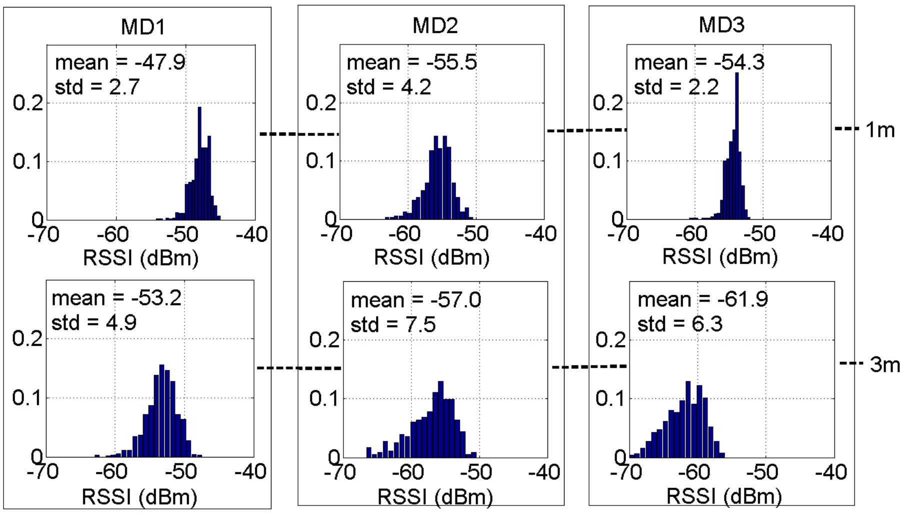


Figure 3.4: Histogram of the Bluetooth RSSI levels for three MDs measured at the same SN; distances one and three meters.

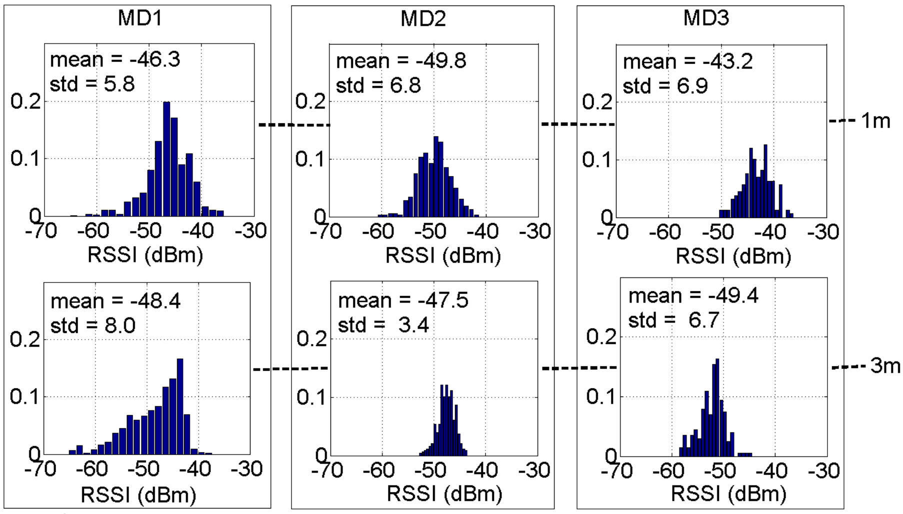


Figure 3.5: Histogram of the WiFi RSSI levels for three MDs measured at the same SN; distances one and three meters.

of mobile phones, we can directly observe a much larger deviation of RSSI values at three meters than at one meter. We also observe that WiFi signals are less robust to deviation than Bluetooth signals.

Response Rate

While RSSI-related metrics are vulnerable to multipath propagation, the Response Rate (RR) of a device is not and has the potential for localization. The response rate is defined as the average number of captured packets per minute that a device encounters for WiFi and Bluetooth technologies.

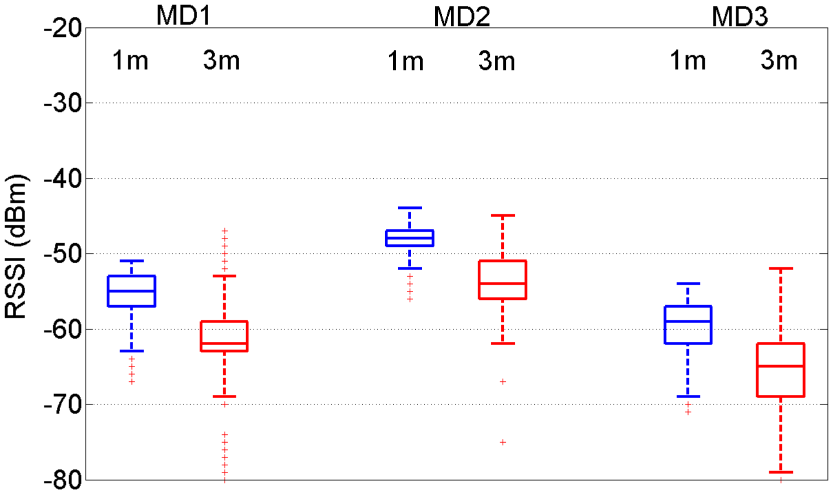


Figure 3.6: Bluetooth boxplots of three MD, RSSI measured by the same SN; distances one and three meters.

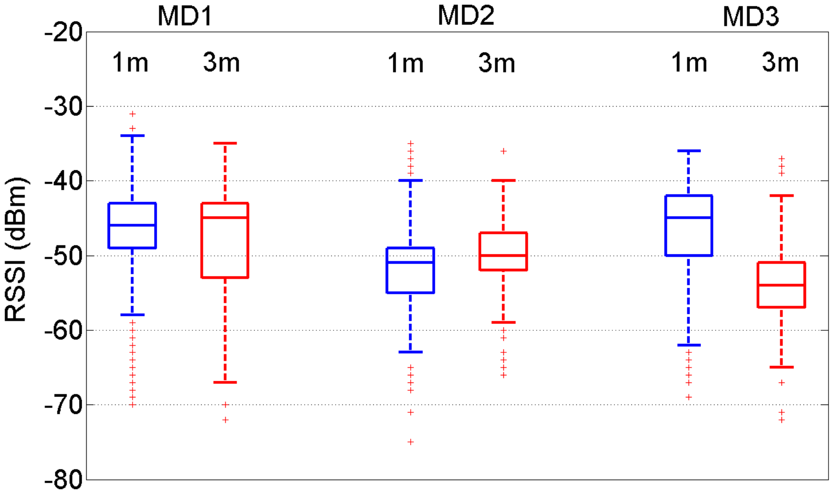


Figure 3.7: WiFi boxplots of three MD, RSSI measured by the same SN; distances one and three meters.

Results for the RR of both Bluetooth and WiFi for all studied scenarios are shown in Table 3.1. We see that the RR of Bluetooth signals varies in an incoherent way making it difficult to relate it to distance. Frequency hopping in Bluetooth causes the RR to depend on channel synchronization and obstructs its use for localization. No such discrepancies are observed in the case of WiFi, where the RR is a function of the distance. Although values for devices differ, the changes in RR in the range are consistent.

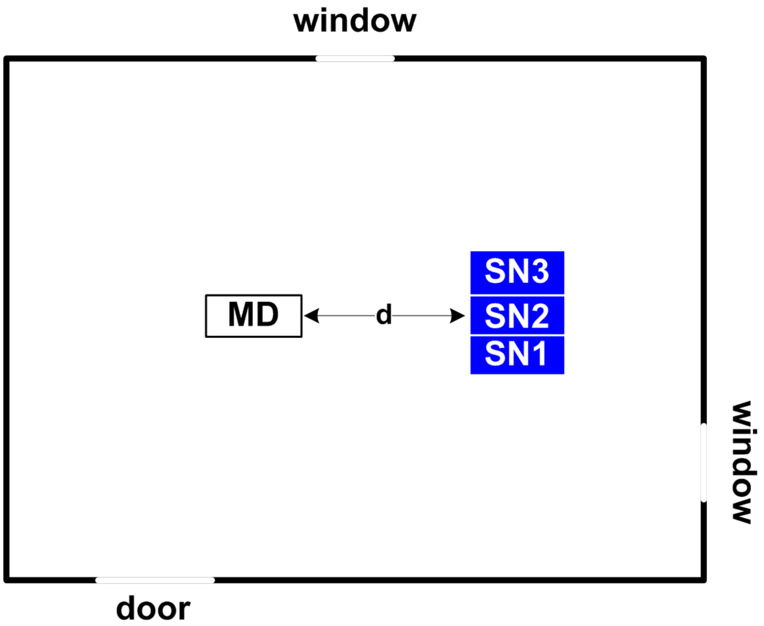


Figure 3.8: Scenario B: setup

3.3.2 Evaluation B: Manufacturing Discrepancies

Evaluation Setup

At the receiver side, technical factors that may affect the RSSI are manufacturing discrepancies between SNs. We expect smaller effects, compared to the impact of device specifics, since all sensors should comply with the same specifications. Experiment B was designed to examine the behavior of three SNs of the same manufacturer and model (Gumstix Overo Fire). The experiment setup is shown in Figure 3.8. We place SNs virtually at the same spot (sensor's dimensions cause some displacement) at distance one and three meters from an MD. At each distance, measurements were collected for 30 minutes. Based on the conclusions of Section 3.3.1 we selected as evaluation metrics the median and percentiles (depicted by a boxplot diagram) and the response rate.

Table 3.1: Experiment 1: Response Rates

	Bluetooth			WiFi		
	MD1	MD2	MD3	MD1	MD2	MD3
1m	12.9	6.8	11.3	23.3	12.9	14.2
3m	13.6	6.0	10.8	18.8	5.2	5.1

Median and Percentiles

The boxplots in the case of Bluetooth signals are shown in Figure 3.9. The median of different SNs varies in the order of 2-3 dBm. This variation is much less than the 10-15 dBm registered by different MDs in Figure 3.6; the RSSI deviation for SNs is also lower. In the case of WiFi, see Figure 3.10, the differences in the median between sensors increases to 5-6 dBm coming close to the results for device specifics.

It is hard to say with certainty that the differences exist because of manufacturing tolerances and not of multipath. We can, however, conclude that manufacturing tolerances seem to have smaller relative impact on the RSSI than device characteristics have and that a real system can abstract from their effects.

Other observations, already discussed in Section 3.3.1, continue to hold: (1) longer distances lead to lower and more spread RSSI measurements; (2) multipath propagation exhibits stronger for WiFi than for Bluetooth signals; and (3) RSSI of Bluetooth signals is lower compared to a WiFi signal.

Finally, it is easily seen that the height of the boxplot between 25 and 75 percentile are almost equal to all three distances. Hence, variance is consistent with different distances in one LOS direction.

Response Rate

The response rate RR of both Bluetooth and WiFi signals calculated at each SN is shown in Table 3.2. Two observations are worth noting. First, the RR of different sensors is similar, given the same technology and distance. This observation leads us to believe that manufacturing tolerances of SNs have less impact on the detection rate than MDs (c.f. Table 3.1). Second, the RR is difficult to relate to distance for Bluetooth signals but can be helpful in WiFi.

3.3.3 Evaluation C: Direction-Specific Multipath

Evaluation Setup

Depending on the direction in which a signal propagates it will face different obstacles in its path. Given the high density of obstacles indoors, we expect that directionality is important. As a proof of concept, we performed Experiment C,

Table 3.2: Experiment 2: Response Rates

	Bluetooth			WiFi		
	SN1	SN2	SN3	SN1	SN2	SN3
1m	20.6	16.2	19.7	43.4	30.1	41.0
3m	15.4	16.8	16.3	37.8	20.3	55.5

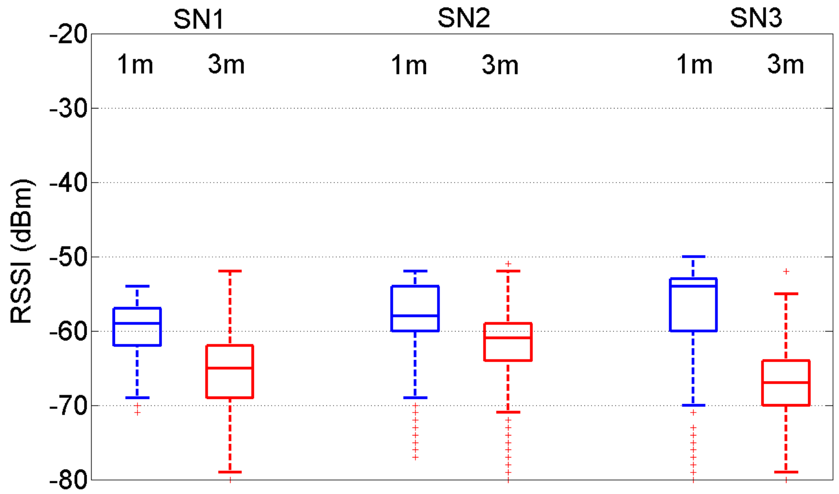


Figure 3.9: Bluetooth boxplots of three SNs measuring the RSSI values of the same MD; distances one and three meters.

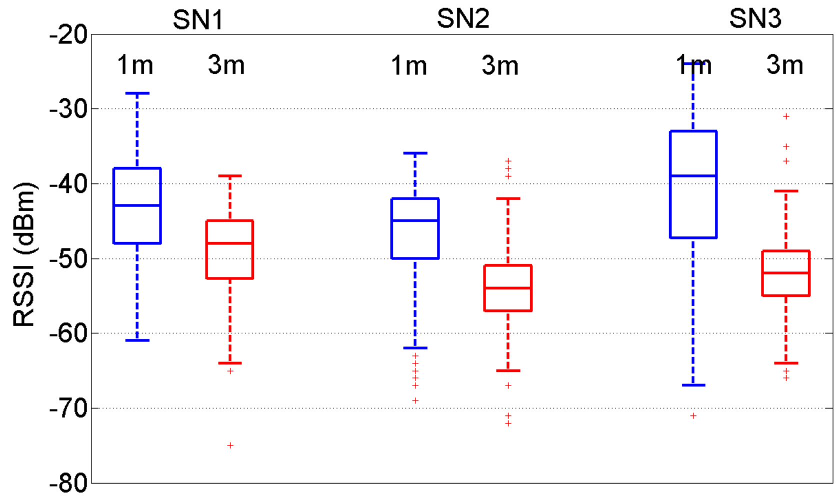


Figure 3.10: WiFi boxplots of three SNs measuring the RSSI values of the same MD; distances one and three meters.

based on the setup of Figure 3.11. Eight sensors in scanning modes (SN1-8) are organized in a grid around a central sensor (SN0) that periodically sends out WiFi beacons. The scanning nodes SN1 to SN8 collect RSSI measurements of SN0's beacons. The experiment was run for 24 hours to collect a reliable number of samples per SN (ten thousand), which allows us to construct a stochastic profile of the radio channels in each direction. The step size of the grid is one meter. SNs' antennas are omnidirectional.

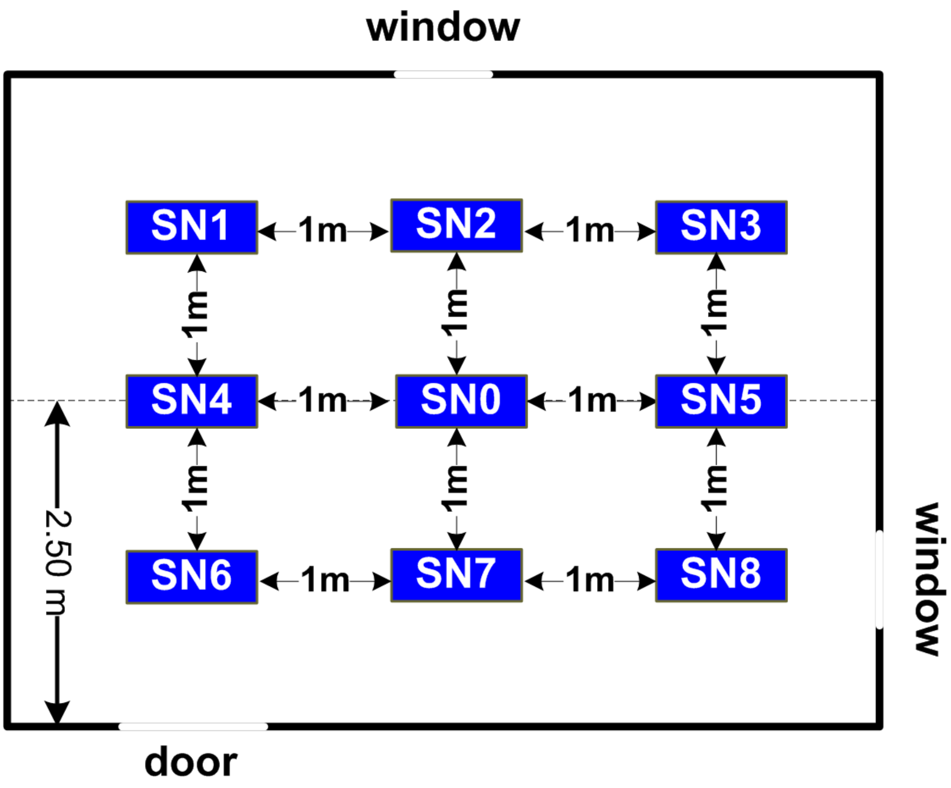


Figure 3.11: Scenario C: setup

Cumulative Distribution Function

The Cumulative Distribution Function (CDF), constructed by the scanning sensors, is shown in Figure 3.12. The position of the CDF graph in the figure corresponds to the position of the scanning sensor, e.g., the CDF graph at position left-middle corresponds to SN5.

Our main conclusion is that no two SNs have the same distribution of the RSSI values, which can be explained by the distinct propagation conditions that characterize each path. Despite the differences, there are certain similarities. CDF curves of SNs on the diagonals to SN0 (SNs 1, 3, 6 and 8) have a five dBm lower mean and a larger variance than SNs 2, 4, 5 and 7. Differences in the distance cause these results. i.e., the first group is at 1.4m from the center while the second is at 1m. SN6 is an exception with higher RSSIs, which we attribute to the node's location. A SN near a corner receives stronger reflected signals from the near walls than a SN in the center of the room.

Moreover, SNs from opposite directions show similar behavior. For example, SN4 and SN5 have RSSI values mainly spread between -40 and -30 dBm, while the CDFs of SN2 and SN7 are in the range of -45 to -33 dBm. Although the specific

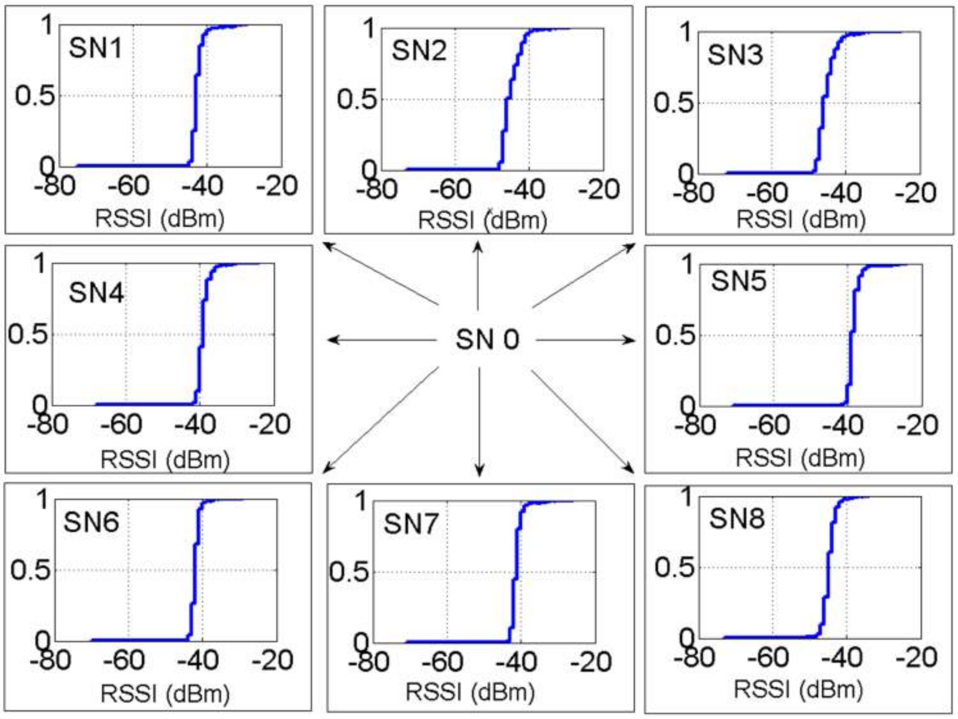


Figure 3.12: CDFs in eight communication directions.

causes of such behavior are hard to determine, we explain it with the asymmetric shape, i.e., rectangular, of the room and the consequences of that on signal propagation.

For practical deployment, our results imply that a localization algorithm should be able to differentiate between the direction from which a radio signal is received so it can compensate appropriately for specific multipath conditions.

3.3.4 Evaluation D: Channel Reciprocity

Evaluation Setup

Similar to the setup in Section 3.3.3, we deployed 16 SNs on the ceiling with 1m distance between each SN in both x and y directions (c.f. Figure 3.13). This setup has a very high deployment density. Consider all estimated locations will be approximated by a point on a grid, the minimum accepted localization error should less than 0.5m. This scenario emulates a shopping mall where SNs can be deployed on the ceiling and maintain LOS communication links with target devices. In this experiment, we focus on WiFi technology. Since both WiFi and Bluetooth technologies are operating the same frequency (2.4 GHz), we believe that Bluetooth experiments would yield to the same conclusions as WiFi ones. In our setup, all SNs send and receive beacons.

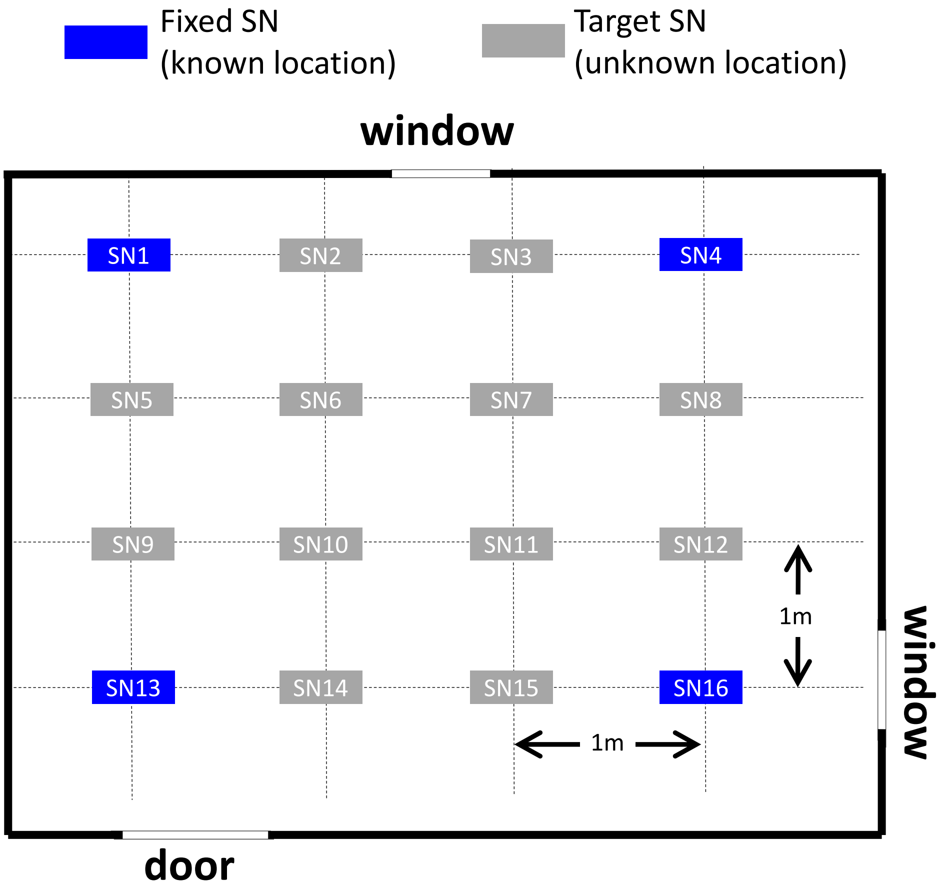


Figure 3.13: Evaluation setup for LOS localization using WiFi SNs.

Median and Percentiles

In the first scenario, we only consider SNs 1, 4, 13, and 16 as receivers and the rest as transmitters. In the second scenario, we only consider SNs 1, 4, 13, and 16 as transmitters and the rest as receivers. From this experiment, we want to examine channel reciprocity: if the transmission from x→y will result in the same RSSI measurements as the transmission from y→x. More specifically for localization, we examine the question of whether measurements at the user side show the same performance at the network side, even in LOS conditions. We considered SN6 as an example. Results illustrated in Figure 3.14 show the RSSI boxplot of communication between SN6 and SNs 1, 4, 13, and 16 in both direction. In the first set of experiments (1→6, 4→6, 13→6, and 16→6), SNs 1, 4, 13, and 16 were transmitting beacons and SN 6 overheard their transmission. In the second set of experiments (6→1, 6→4, 6→13, and 6→16), SN1 was transmitting beacons and SNs 1, 4, 13, and 16 were overhearing its transmission. Considering any pair of channels (e.g., 6→1 and 1→6), we see a big variation of RSSI measurements. It is easily seen in Figure 3.14 that the height of the boxplot between 25 and 75

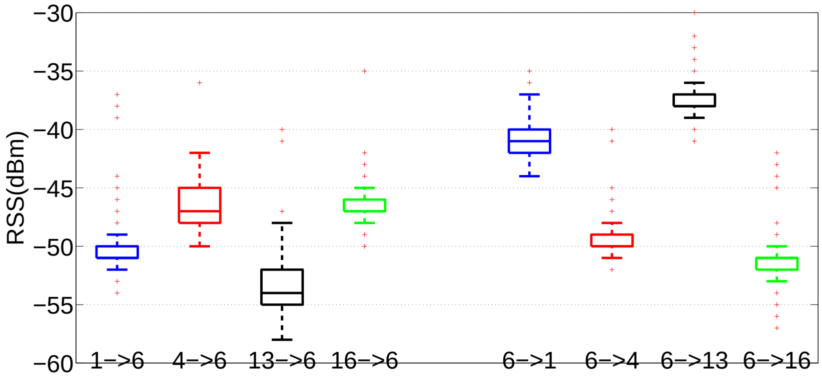


Figure 3.14: Bi-directional radio measurements for indoor LOS communication

percentile is not the same in different directions. Hence, variance is inconsistent over in different LOS directions. Using these results presented in Figure 3.14, we conclude that indoor channels are not reciprocal even in LOS conditions. These significant differences in RSSI measurements accumulate different factors, such as device manufacturers.

3.4 Experimental Setup for LOS Localization

We consider two scenarios for localization: (i) localizing target SNs (with unknown location) using measurements collected at the fixed SNs (with known location) and (ii) localizing the target SNs using radio measurements collected at their sides (the fixed SNs act as transmitters). Fixed SNs behave as a network and hence, the first scenario (i) represents a network-based localization and the second scenario (ii) would represent user-assisted localization. We expect user-assisted localization to show different performance than network-based positioning. To validate our assumption related to channels reciprocity, it is necessary to perform a complete localization algorithm.

3.5 Experimental Results

Results illustrated in Figure 3.15 express network-based localization (i.e. edge SNs 1, 4, 13, and 16 overhear transmitted beacons from the target SN). In our experiments, edge SNs overhear beacon transmission from SNs 2, 3, 5 - 12, 14 and 15. We collected measurement over 1 hour and aggregate it into periods of 5 min. We consider Equation 2.4 for signal propagation in free space. We consider the PLE indoor LOS equals 2 (similar to outdoor free-space). Given our previous knowledge of the transmitted power of WiFi SNs $P_{tx} = 20$ dBm, we perform localization based on multilateration techniques described in Section 2.9.5. For network-based

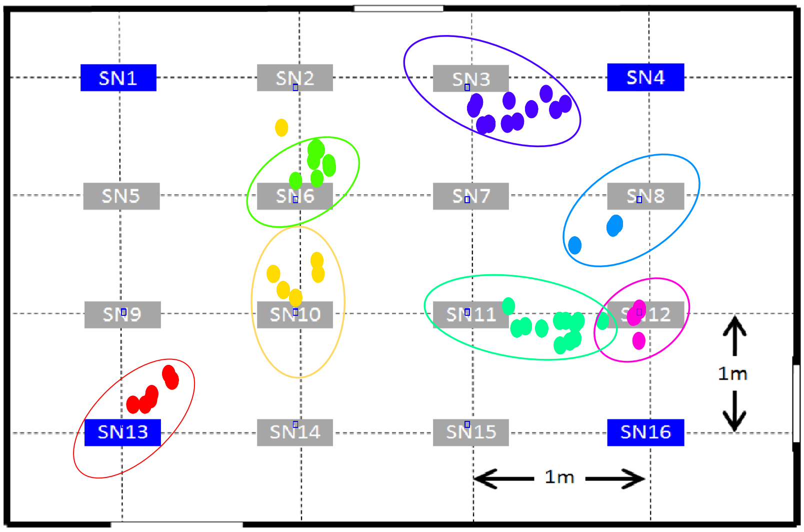


Figure 3.15: Network-based localization using 5 min aggregation.

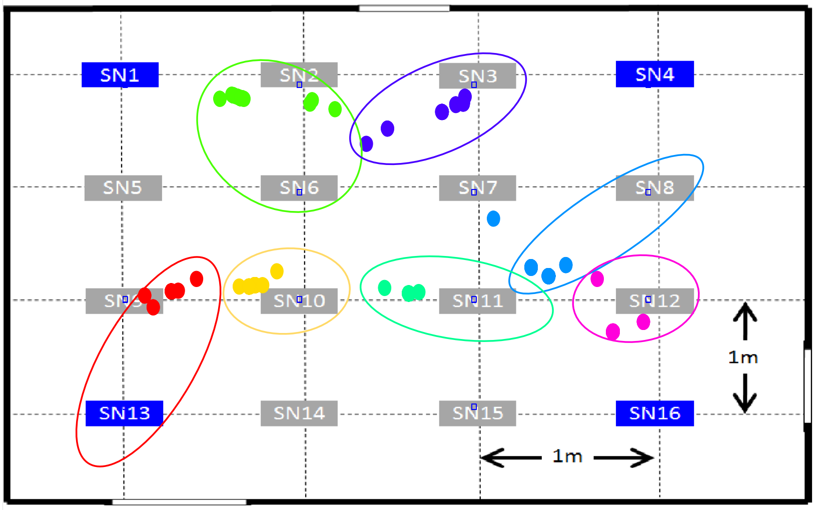


Figure 3.16: User-assisted localization using 5 min aggregation.

localization, average localization error μ is 0.439m, error deviation σ is 0.2m and and RMS is 0.484m. These results with the current deployment setup mean that a constant PLE is indoor LOS environment is not entirely accurate. Instead, a more smart solution that adapt the PLE with time is required. Moreover, the constant PLE assumption becomes more challenging in the case of NLOS and mobility scenarios. However, when applying the user-assisted localization (the target device overhear beacon transmission from edge SNs), we observed a difference in the localization error mean of 0.68m and RMS of 0.721m without much of a change in the localization error standard deviation. Results in Figure 3.16 show some example of user-assisted localization.

3.6 Conclusions

This chapter presented an investigative study of factors that affect radio signal strength and the implications for WiFi and Bluetooth localization systems. We designed real-world experiments to investigate the impact of technical characteristics of MDs (targets for localization), manufacturing differences of SNs (used for localization) and direction-specific multipath propagation. In parallel, we analyzed the usability of four signal metrics, namely, instantaneous values, probability distribution, median and percentiles, mean and standard deviation, as well as the signal's detection rate.

Our main conclusions are: (i) signal strength varies less between sensors of the same type than between MDs from different manufacturers; (ii) multipath seems to have dominant effect on signal strength; (iii) radio signals experience distinct propagation conditions in different directions; and (iv) the choice of evaluation metric depends on the time granularity of localization, i.e., mean values are convenient for real-time localization. Other factors, such as obstacles along the propagation path, can also affect radio signals, a topic of our future investigations. Moreover, since we are interested in a combined Bluetooth and WiFi localization algorithm, we intend to analyze any interference issues between the two technologies.

In terms of evaluation metrics, we conclude that the choice of metric depends on the time granularity needed by the localization algorithm. Probability density functions and boxplots are more representative than single RSSI value. However,

Table 3.3: Error comparison of localization error mean, standard deviation (meter) and RMS.

AN Location	Network-based			User-assisted		
	μ	σ	RMS	μ	σ	RMS
AN2	0.423	0.124	0.440	0.741	0.257	0.784
AN3	0.574	0.298	0.646	0.429	0.158	0.457
AN5	0.263	0.208	0.335	0.754	0.221	0.785
AN6	0.278	0.241	0.367	0.849	0.293	0.898
AN7	0.295	0.300	0.420	0.683	0.186	0.707
AN8	0.675	0.158	0.693	0.840	0.239	0.873
AN9	0.547	0.183	0.576	0.335	0.252	0.419
AN10	0.333	0.193	0.384	0.835	0.187	0.855
AN11	0.547	0.253	0.602	0.107	0.231	0.254
AN12	0.303	0.264	0.401	0.967	0.122	0.974
AN14	0.238	0.120	0.266	0.186	0.287	0.342
AN15	0.795	0.136	0.806	0.466	0.137	0.485
Average	0.439	0.206	0.484	0.680	0.214	0.721

they also require the collection of many RSSI samples. They are better used in localization applications whose principal purpose is the collection of long-term statistics. When a quick evaluation is desired, e.g., as in real-time systems, the mean of a group of samples is more convenient to handle. In all cases using a single instantaneous RSSI value is not recommended.

In terms of performance, we conclude that mobile phones show a significant difference, i.e., RSSI levels. This fact should be considered in the development of a localization algorithm. One possible approach to compensate for these differences is to not process absolute RSSI measurements of a device but to relate its measurements from several scanning SNs.

Data acquisition at the network side will result in a different localization error than at the user side. This is because radio channels are not reciprocal even in LOS conditions. The constant path loss exponent for indoor LOS scenarios is just an approximation that might degrade the overall localization performance, especially in NLOS conditions.

Chapter 4

GSM-Based Localization Systems

The wide availability of GSM networks encourages research on the use of GSM as a standard radio technology for service development. Also, GSM signals appear more stable over time in comparison to WiFi or Bluetooth signals [143], which is a crucial factor in the quality of the localization service. Detailed analysis of the comparison between WiFi and GSM signals are provided in Chapter 5. This signal stability is due to two main reasons: The first reason is related to the operating frequency range as discussed in Section 4.2.1. Recall that GSM operates at a lower frequency range than WiFi or Bluetooth. Hence, GSM signals show, ideally, more resilience to NLOS reception than WiFi signals. The second reason is related to signal interference. The GSM access scheme operates on a controlled and managed the licensed band, where spectrum resources are well optimized to avoid as much signal interference as possible. However, as discussed in Section 2.2, WiFi operates in the unlicensed band, where signals are vulnerable to radio interference depending on the number of *active* surrounding devices. In this chapter, we investigate the use of GSM signal for passive indoor localization (without collaboration with end-users or the network operator).

4.1 Introduction

Without possible collaboration with the network operator or end users, the design of a GSM-based passive localization system has several challenges related to the nature of the wireless medium and the GSM standards:

- Given the spectrum resource management by the network operator, how can we capture GSM radio signals, convert them to messages and parse the message content?

- Given the privacy restrictions to subscribers, can we identify the signal's source to provide accurate information for service providers? (e.g.,the number of GSM devices within a certain area).

- Given the collaboration constraints in a passive system, how can users be localized without any knowledge of their devices' radio settings?

This chapter discusses in detail these challenges and offers solutions to each one of them. Our contributions to this chapter are as follows:

- We present a complete setup for a single-/wideband passive GSM receiver. These receivers provide our proposed passive system with required information, such as RSSI and MD identity, without collaboration with end users or network operators. These receivers show a reliable performance of signal recovering with a success rate up to 99% of downlink messages and around 70% of uplink messages. Moreover, we show a detailed insight into scenarios that support passive wideband capturing of the GSM radio.

- In terms of processing capabilities for the wideband receiver, our proposed solutions can split up to 76 GSM channels using an optimized CPU-based implementation and more than 125 GSM channels using a GPU-based implementation.To process this large number of channels, we present a distributed approach of our GSM receiver over an IP network to servers for signal processing (Layers 1, 2, and 3).

- We developed two main localization algorithms that can overcome the challenges brought by the passive environment. We validate our proposed solutions using a set of real indoor experiments to localize multiple static users simultaneously and in real-time. Localization errors are in the range of 3m.

- Finally, to improve the localization performance, we proposed a filtering algorithm to obtain high-quality and stable radio metrics before being used in the localization process. The proposed filtering solution shows additional improvement up to 75% for locations with high signal variations.

4.2 GSM Single Frequency-Channel Receiver

As discussed in Section 2.3.2, all broadcast, control, and dedicated messages are combined from four NBs. Hence, our uplink received is expected to be agnostic to the message content. Given the GSM multiframe example in Figure 2.10, our target is to have one receiver able to capture all downlink and uplink messages (except FB, SB, and RACH). Recall that *a GSM MD does not send any synchronization messages in the uplink*. The only input parameter to our system should be the carrier frequency, which can be done in an offline process as will be discussed later in Section 4.2.2. However, the design of our receiver should take into consideration the specification of layers 1, 2 and 3 and perform the ”reversed” operation for every captured message.

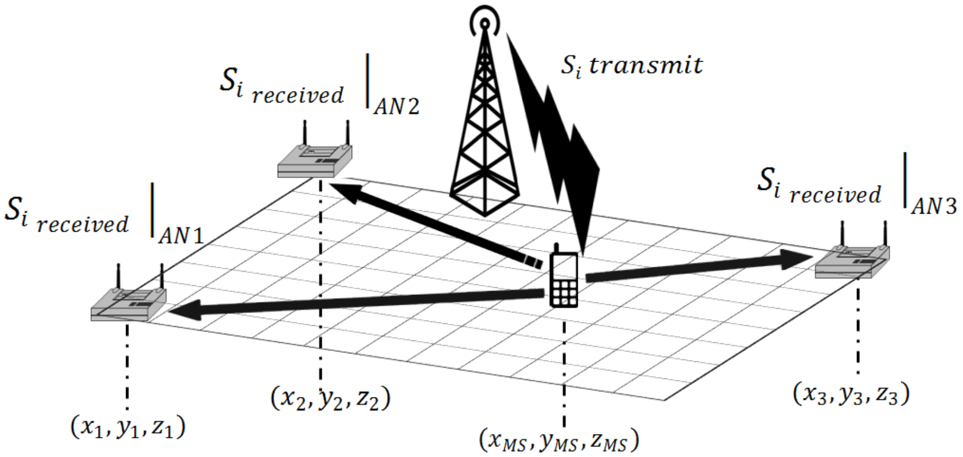


Figure 4.1: General design of a passive receiver system

4.2.1 Passive Receiver Requirements

A passive receiver relies on the concept of signal overhearing, in which the communication between two radio devices is overheard by a third device for the purpose of a particular application. The system operation is illustrated in Figure 4.1. A MD communicates with a BTS and the traffic is overheard by a number of passive SNs, termed ANs. ANs do not have any information about the MD location or transmission time and frequency. Ideally, BTSs cover geographical areas with hexagonal shapes. MDs located within a BTS's coverage area communicates only with that BTS. A passive receiver system has to deal with several challenges before becoming a commonly adopted solution. Radio signal acquisition is among the initial requirements and is the focus of this section.

GSM Signal Capturing

One of the challenges of a GSM-based passive receiver is to capture the radio signal of a particular user. In GSM, information is divided into blocks referred to as bursts, which fit into a single Time Slot (TS) of the GSM time frame [106]. Therefore, signal capturing requires frequency and time synchronization between transmitter and receiver. In order to support the MD in this task, the BTS periodically transmits FB and SB bursts.

An active system, running on either the mobile station or the GSM network, can use the availability of FB and SB to reconstruct the carried messages. This is how Airprobe works [13]. In a passive system, however, the MD does not cooperate or offer any information on frequency and time synchronization to recover uplink messages. MDs at different distances from the serving BTS will adjust their uplink transmission with different corresponding TA values. Different TA values are an additional difficulty for synchronization inside our proposed solution. To tackle the challenge of synchronization to the mobile user, we propose in Section 4.2.2 an approach relying on processing of NB to tune to individual MDs.

The first stage in message reconstruction is converting an analog signal to a digital bit sequence, carried out by the physical layer as illustrated. Next, since GSM uses time multiplexing with an elaborate multiframe structure; we need to recombine bursts into messages. A single message on the data link layer (184 bits) maps to 456 bits on the physical layer, which fit into four NBs located in four consecutive TDMA frames [106]. Failing to obtain the correct order of NBs or missing one of them would prevent the message reconstruction. Section 4.2.2 describes our solution to overcome this problem. Finally, the combined data should be interpreted according to the GSM standard. For this purpose, a message parser should be implemented. While modules for signal processing of the GSM physical layer are available, modules for message reconstruction and parsing in the uplink are more difficult to find. In fact, at the time of our research no such tools were available.

In particular, we found the following challenges for the message reconstruction process:

- Low SNIR - the total amount of received signal power at the anchor terminal P_{rx} is influenced by power control in GSM uplink as well as transmission path loss, channel noise N, and interference I from i_n co-channel cells. Successful message decoding requires an SNIR above a certain threshold, formally expressed by:

$$\frac{P_{rx}}{N + \sum_{i=1}^{i_n} I} \geq \text{SNIR}\,|_{\text{decodable message}} \tag{4.1}$$

  However, in uncontrolled environments like the proposed GSM passive receiver, the interference is not controlled at the USRP antenna. The SNIR issue is further investigated in Section 4.2.3.

- Time and frequency misalignment - imprecise synchronization of the anchor node and MDs leads to the erroneous reconstruction of messages, rendering them either unreadable or falsely parsed. The issues are discussed in detail in Section 4.2.2.

- Message encryption - encryption hides user data that is potentially useful for identification. The current work is limited to unencrypted messages.

GSM User Identification

In GSM, a MD can be identified by its IMSI or TMSI. Therefore, any passive receiver system, after capturing GSM radio signals, should reconstruct the carried

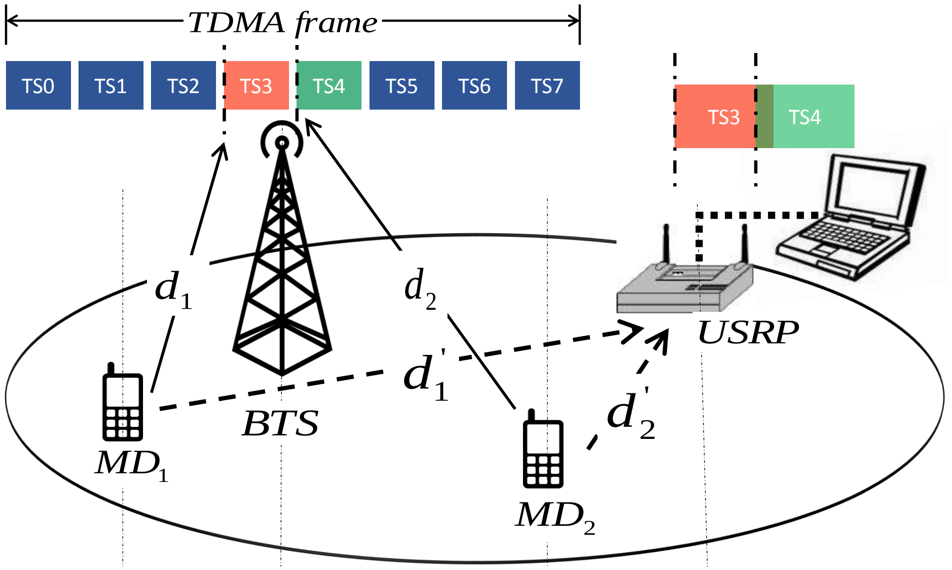


Figure 4.2: Time synchronization challenge in GSM uplink passive receiver

message to detect the presence of one of the two identifiers. In a preliminary investigation, we identified few procedures, which contain useful messages including IMSI or TMSI without encryption. These scenarios are (i) Mobile Originating Call (MOC), (ii) Mobile Terminated Call (MTC), and (iii) Location Update Request (LAU). Experimental analysis of these messages is presented in Section 4.2.3.

4.2.2 Passive GSM Receiver

The first step towards GSM signal reconstruction is system synchronization. Frequency synchronization is easily done, however, time synchronization is challenging. This approach is best illustrated by an example. As mentioned in Section 4.2.1, each MD has an individual TA value to compensate for its distinct delay in reaching the base station. Let there be two devices MD1 and MD2 at distances $d1$ and $d2$ from the base station. There will be two corresponding TA1 and TA2 values as illustrated in Figure 4.2. Unless co-located with the GSM BTS, a passive receiver will be located at the different $d1'$ and $d2'$ distances from MD1 and MD2, respectively. As a result, transmissions from the two devices are unsynchronized to the passive uplink receiver. Hence, messages from two MDs may overlap at the receiver terminal.

Symbol and Frame Synchronization Challenges

An ideal solution for the time synchronization challenge outlined in Sections 2.3.2 and 4.2.1 is to intercept downlink and uplink signals simultaneously as illustrated in Figure 4.3. In the first step of Figure 4.3-b, the receiver is tuned to the C0 down-

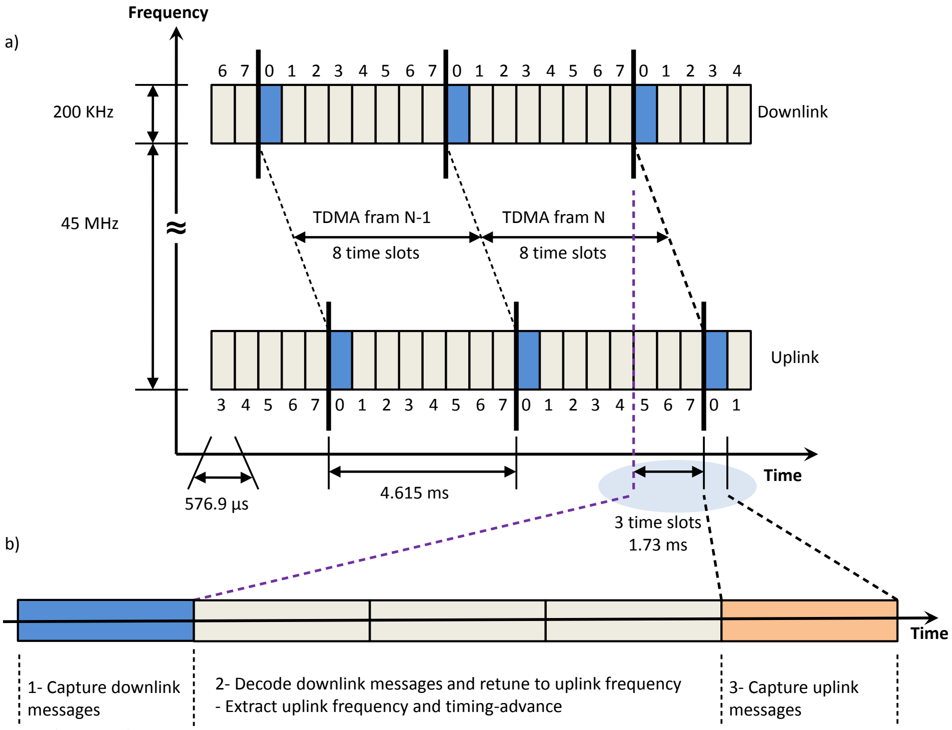


Figure 4.3: Ideal solution for fast retuning between downlink and uplink channel.

link channel (described in Section 2.3.3) and captures downlink messages (made out of 4 NBs), such as *Immediate Assignment* message transmitted on RR logical channel [64]. The *Immediate Assignment* message informs the target MD with its allocated uplink frequency and its own timing advance value. Given the fact that uplink transmissions are shifted 3 time-slots from downlink transmissions as shown in Figure 4.3-a [64], the receiver has 3 time-slots (=1.73 ms) to perform the following steps: (i) decode the downlink message and extract uplink frequency and timing advance value of a particular user, (ii) retune the USRP internal clock (frequency synchronization) to the uplink frequency, and (iii) tune the uplink receiver internal counter (time-synchronization) with the timing advance value. However, the implementation of such a system using commonly SDR hardware, e.g., USRP, faces several hardware limitations in our experience.

- *Radio Front End:* The currently available USRP hardware does not allow two front end receivers to be tuned individually for capturing GSM uplink and downlink channels in parallel. In February 2013, Ettus released the new Quad Receiver QR210, which integrates four individual front ends but at the high cost of several tens of thousands of dollars.

- *Bandwidth:* The frequency gap between the GSM uplink and downlink (in FDD) is higher than what the current USRP daughterboard supports, preventing capturing of the complete band by a single front-end receiver.

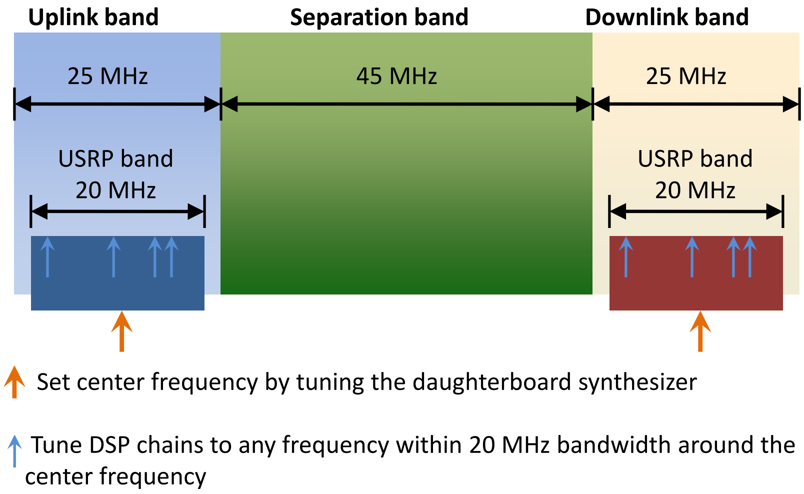

Uplink band
Separation band
Downlink band
25 MHz
45 MHz
25 MHz
USRP band
20 MHz
USRP band
20 MHz
Set center frequency by tuning the daughterboard synthesizer
Tune DSP chains to any frequency within 20 MHz bandwidth around the center frequency

Figure 4.4: Retunning the frequency of an USRP N210 device.

- *Processing Delay:* According to our measurements, the current latency of re-tuning in both hardware and software is around 5 ms, which is too high to allow quick switching between uplink and downlink. The retuning process in an USRP N210 device is made in two steps as illustrated in Figure 4.4.

It is, however, possible to connect two USRP devices in parallel to scan the uplink and downlink channels but at double system cost. Given the above restrictions, we propose to conduct time synchronization with the MD directly on the uplink, relying on the structure of the NB. Similar to downlink synchronization, where the long training sequence of the synchronization burst is used, we are recovering NBs by correlating the captured signals with their short training sequence TSC(NB).

Architecture of the Proposed Passive Receiver

A preliminary step to message recovery is frequency synchronization. This step is done by scanning the downlink to create a mapping between the serving cells and the uplink allocated frequencies in that cell. After tuning to an uplink frequency, as illustrated in Figure 4.5, the captured samples pass from the UHD (step a) to GR LPF and Interpolator (steps b and c). For time synchronization, the passive receiver applies the following phases on the received samples from the Interpolator: (i) discovery of the used TSC(NB); (ii) NB detection; (iii) message reconstruction and (iv) message parsing. These phases from step d in Figure 4.5 and are implemented in the 'gsm-receiver' block.

Determining TSC(NB) In Section 2.3.2, we introduced the TSC(NB) as one out of eight training sequences, identified by a sequence number. It's number indicates the TSC(NB) used by a cell in the cell BSIC. Hence, by listening to the

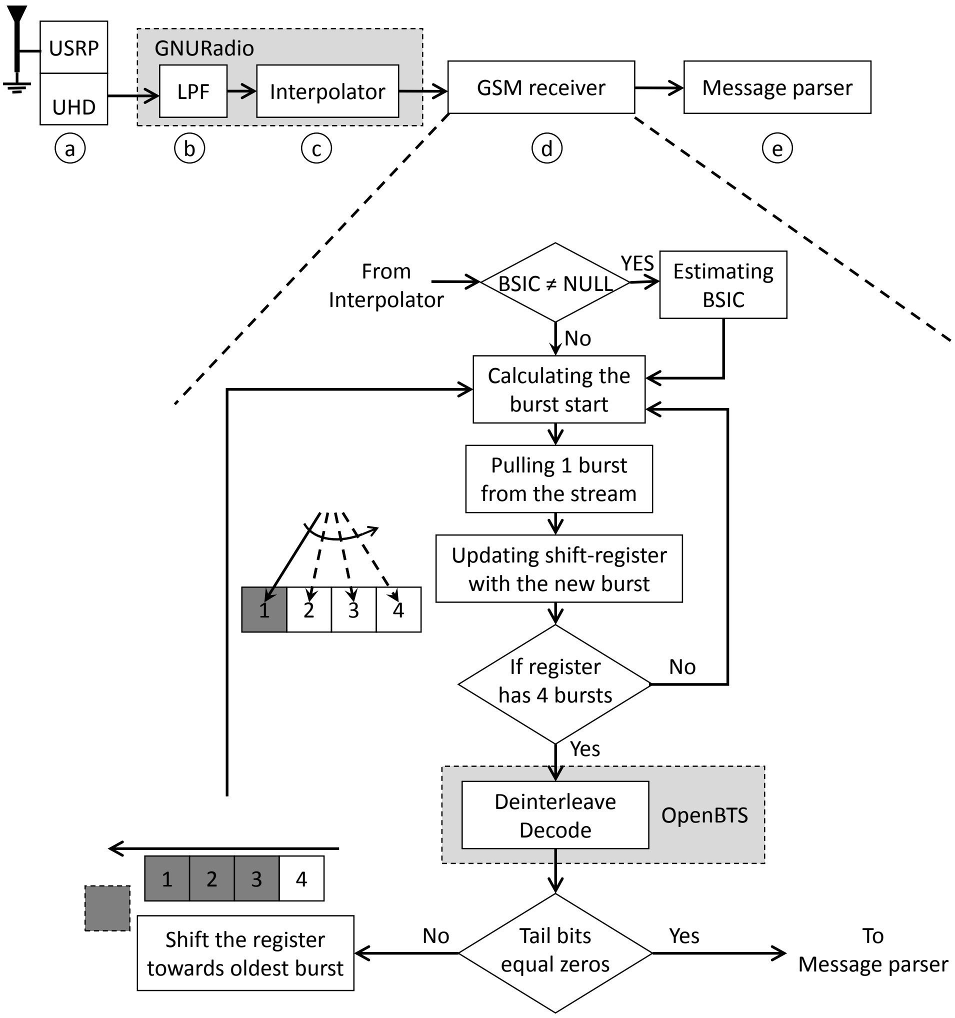


Figure 4.5: Algorithm for passive synchronization recovery for an uplink signal.

downlink, we can capture the SB and discover the TSC(NB). To avoid the need of re-tuning between downlink and uplink, we propose an alternative, namely cross-correlating (C_{xy}) the received signal stream S with the eight predefined TSC$_i$(NB), $i = 1...8$. The TSC(NB) that results in the highest peak-to-average ratio ($p2a$) of the corresponding correlated stream (and in a successfully decoded message) is chosen for burst recovery. The procedure can be expressed:

$$i = \text{BSIC} = \arg\max_i (p2a(C_{xy}(S, TSC_i))) \mid i = 1:8$$

. The $p2a$ ratio is calculated from the peak amplitude of the signal divided by the signal mean value as illustrated in Equation 4.2.

$$p2a(x) = \frac{\mid x \mid_{\text{max}}}{x_{\text{average}}} \tag{4.2}$$

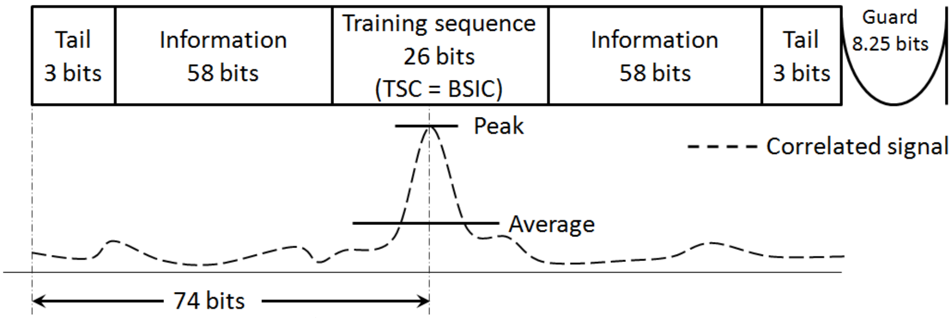


Figure 4.6: Normal burst structure

Finding the TSC(NB) needs to be done only once per cell since the BSIC (the TSC(NB) code) does not change over time.

Detecting NB After determining the TSC(NB) used in the cell, we can detect its appearance in the uplink signal and thus identify individual NBs. Knowing the TSC(NB) position within the NB's structure can help us to determine further the burst start. As illustrated in Figure 4.6, the burst start is an integer shift from the peak of the cross-correlated signal. Once recovered, a complete burst is passed to a demodulation module, and the resulting bits are entered into a shift register for message reconstruction. Recall that a single message is carried over four NBs. To align the 1/4 bit at the end of a NB (each NB is 156.25 bits long) we are running a 157-156-156-156 pattern shift as recommended [68].

The 184 bits (23 bytes) represent L2 information. Abis and Bbis message types do not have layer 2 headers. These two types of messages are transmitted on Downlink channels. Messages transmitted on uplink channels contain additional Layer 2 header. This important note should be considered while parsing received messages on uplink channels. To capture the correct order of the four NBs, our proposed solution is placed on two states: synchronized and unsynchronized.

- The system starts in the unsynchronized state by searching for the first normal burst, using the peak to average approach and registering it as (time slot number 0) TS_0.
- After successful detection of a NB, the system adjusts its starting pointer to the burst start and moves to the synchronized state.
- Different counters start to track different MDs assigned over timeslots:
  - The current burst number.
  - The current timeslot number.
  - The number of successful consecutive captured normal burst.
  - The burst number of the last failed captured normal bursts.
- The bits of successfully demodulated bursts will be sent to the decoder.

Message reconstruction As described in Section 4.2.1, messages from the data link layer are distributed over four NBs within four consecutive TDMA frames. Typically, a MD knows the correct burst sequence in the GSM multiframe from downlink broadcast information. To overcome the lack of this information in the uplink, we implemented a shift register with the size of four NBs (c.f. Figure 4.5-d).

- For each time slot, a Shift Register Memory (SRM) with length = 4 bursts is established, to save the consecutive received normal bursts on that timeslot.
- The system starts by filling the SRM with four consecutive normal bursts (from four consecutive TDMA frames).
- The four bursts are sent for deinterleaving and decoding modules[66]. Note that **demodulation**, **deinterleaving**, and **decoding** modules *are reused components* from the OpenBTS project [141].
- If the decoded message of the current SRM combination return false (by checking the parity bits = 0's). The system update the SRM with a new burst.
- Only if an actual message is recovered, it is passed on to the parser.

Detecting Access Bursts: The following procedure illustrates the proposed algorithm (c.f. Figure 4.7) for detecting access bursts in an unsynchronized manner.

- One burst (156 symbols) from the received signal stream will be sent to the RACH detection algorithm.
- The burst will be correlated with the RACH training sequence. The access burst will be considered as detected if a peak to average ratio of the correlation output is greater than a detection threshold (threshold = 5 based on empirical data)
- If an access burst is detected, then calculate its timing advance compared to the burst start (defined by the internal counter).
- If TA $>$ 76 (156-36-41-3), the burst will not be decoded because the data content of the burst will not be completed in the extracted burst. As a result, adjust the starting pointer.
- If the TA $<$ 76, demodulate the burst, decode it and check the parity check.
- If tail bits are zeros, send information bits for layer 2 decoder. If not, discard information bits.

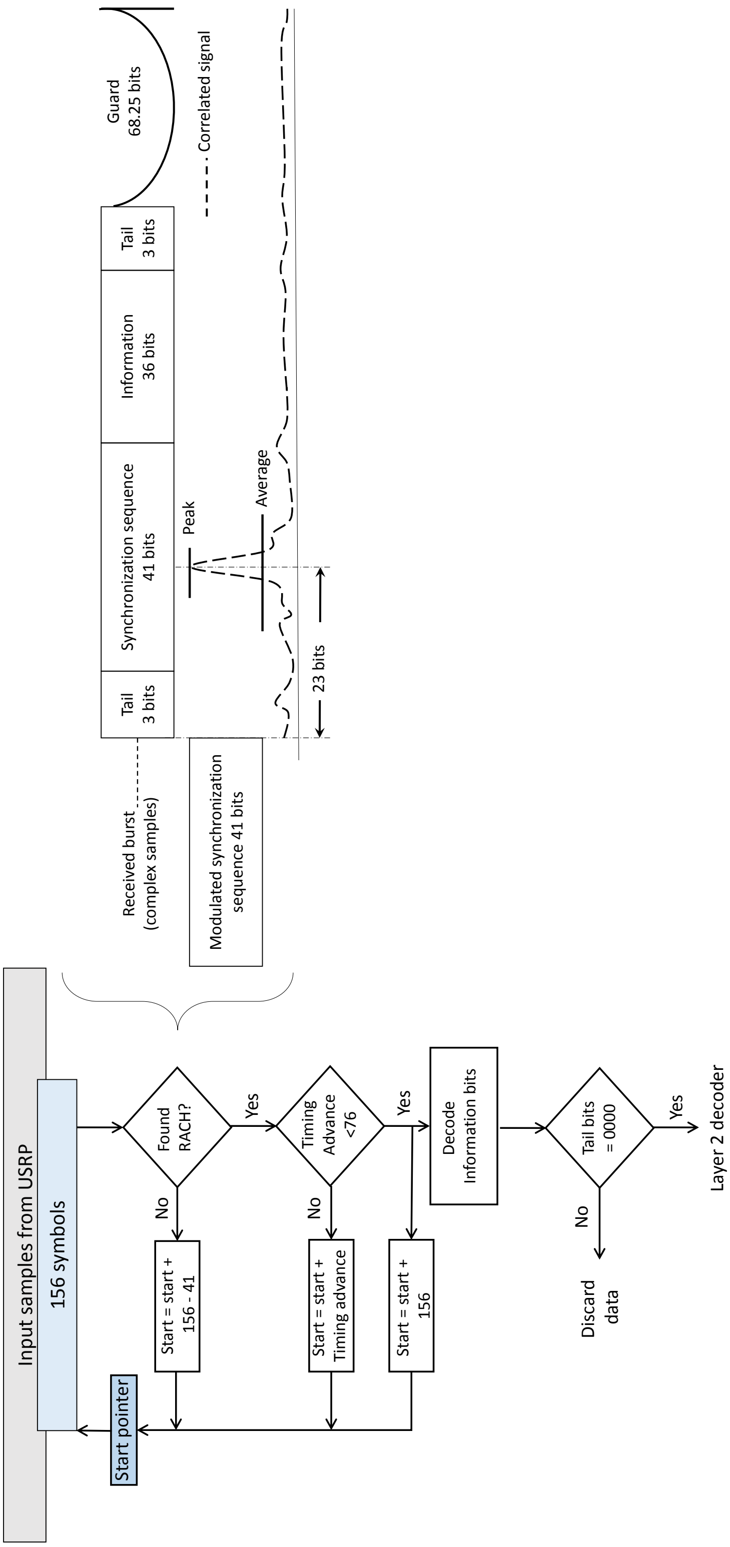


Figure 4.7: Detection algorithm for RACH bursts

Message parsing Finally, we implemented a message parser to enable identification of uplink messages (step e in Figure 4.5). We need to reconstruct GSM messages to find MD identifiers. The parser follows the 3GPP specification on GSM message formats.

Our proposed GSM receiver reconstructs messages on uplink and downlink channels and is agnostic to the message content (whether a message is encrypted or not). However, our message parser interprets correctly unencrypted messages. Our final target for identification and localization services is unencrypted uplink messages with a MD identity, such as Paging Response and Location Update Request messages [69].

4.2.3 Experimental Setup

The implementation and evaluation work was performed using the USRP SDR platform. In particular, we used an N210 motherboard, which runs the SDR software on an external notebook (Lenovo T520), and an SBX daughterboard, which contains one receiver front end in the 400-4400 MHz frequency range. The SDR software includes GR blocks for the necessary signal processing and our modules are written in C++.

The processing chain in Figure 4.5 is tuned as follows: (a) USRP filter bandwidth of 200 KHz; (b) the GR LPF operates at 135 KHz cut-off frequency and 10 KHz transition bandwidth; (c) the fractional interpolator adapts the sampling rate (300 KHz) to achieve the exact desired GSM signal rate.

Experiments on capturing GSM messages were performed in the city of Bern, Switzerland, on uplink and downlink channels administered by the Sunrise network as shown in Figure 4.8. Our target BTS has a broadcast channel (C0) at 941.8 MHz and paired uplink channel, where MDs initiate their access and service requests at 896.8. Its Location Area Code (LAC) is 4000 and Cell ID (CID) is 18321. In this first version of the code, we focused on single channel capturing. Hopping channels are not relevant to our current study.

Power Threshold for Decoding GSM Messages

The SNIR is an important parameter defining the MD capturing rate at an AN. Since GSM uses TDMA, a MD will only face interference from MDs of other cells and inter-symbol interference. The target MD and the AN were separated by a concrete wall as illustrated in Figure 4.9. Achieved results in Figure 4.10 shows the RSSI variation at an AN when a MD initiated outgoing calls. Figure 4.10 is divided into three phases: (i) noise phase: the noise power captured at the AN, (ii) signal phase: the signal power when the target MD initializes a call, and (iii)

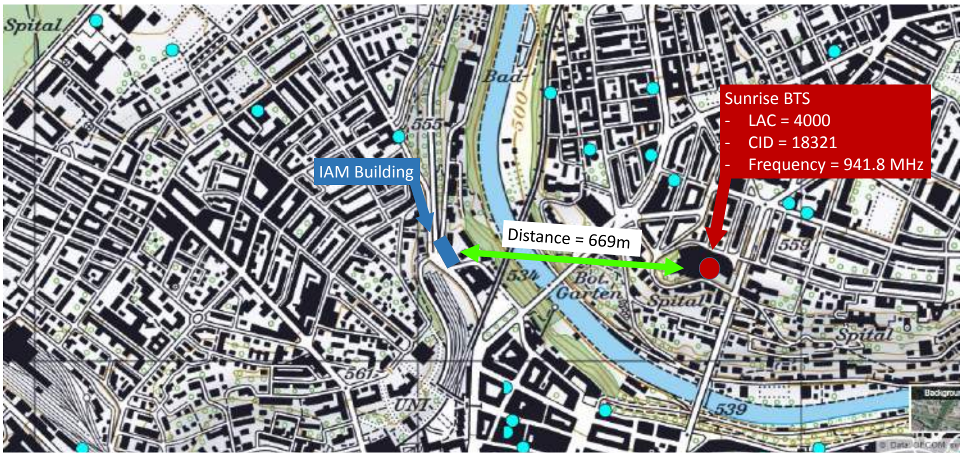


Figure 4.8: A city map showing information about IAM Institute and a Sunrise BTS

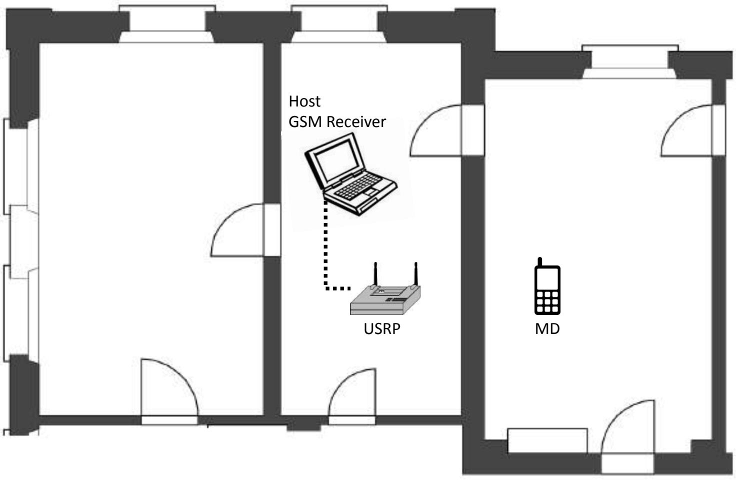


Figure 4.9: Experimental setup for decoding GSM uplink messages

interference phase: the interference power from other MDs using the same TDMA frame. Given the GSM sampling rate, Figure 4.10 illustrates a sample duration of seven seconds. During the MD's transmission, we notice that interference from neighboring cells is negligible. The estimated SNIR in Figure 4.10 is close to 25 dB, which is enough for the *gsm-receiver* module to decode MD messages. Typical GSM receivers define a SNIR threshold of 7-10 dB for success decoding [64]. Recall that we need SNIR above this threshold at each of the eight TSs (equivalent to eight different MDs) inside the TDMA frame.

Downlink Capturing Evaluation

In order to evaluate the ability of the passive uplink receiver to synchronize with GSM MDs, we need to compare its performance to a benchmark. We are not aware

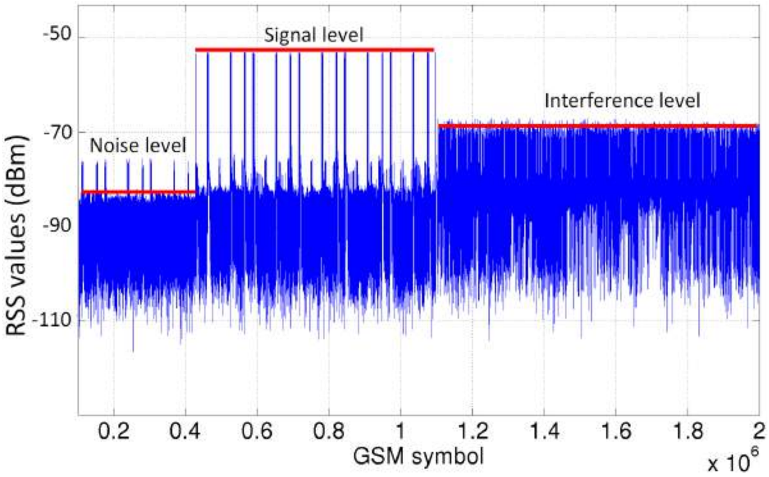


Figure 4.10: Signal, noise, and interference power levels in a GSM uplink channel

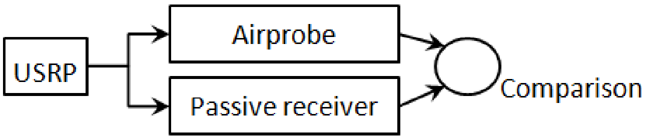


Figure 4.11: Downlink evaluation experiment setup.

of any passive receivers for GSM uplink capturing. OpenBTS cannot be used as an uplink benchmark due to spectrum license issues. Hence, we decided to test our method on a downlink channel with Airprobe as a benchmark. Airprobe made use of FB and SB synchronization and was optimized for downlink capturing, reaching message recovery rates of 95%. Our method uses NBs in the downlink for synchronization. We tuned one USRP, with two different streams, one of them running Airprobe, the other one our passive receiver, on the same downlink Sunrise channel and conducted outdoor measurements for two hours as shown in Figure 4.11. Table 4.1 shows that the passive receiver recovers in best case performance 99% of the messages decoded by Airprobe, with a mean rate of around 98.8%. Combined with the recovery rate of Airprobe, this means our method recovers 94% of the downlink traffic[1].

These imperfections in the observed performance can be caused by imperfect synchronization or weak signal strength. To determine the impact of the latter we compared the received power distribution of both decoded and non-decoded downlink messages. As Figure 4.12 shows, the two empirical histograms are significantly overlapping, excluding poor signal strength as a cause for missed messages. This result leaves imperfect synchronization. First, as shown in Section 4.2.1, TSC(NB) was not designed for synchronization purposes and thus may oc-

[1] Our proposed solution has been tested in a different environment by DFRC company [58]. The achieved performance by DFRC is in the range between 60-70% compared to Airprobe.

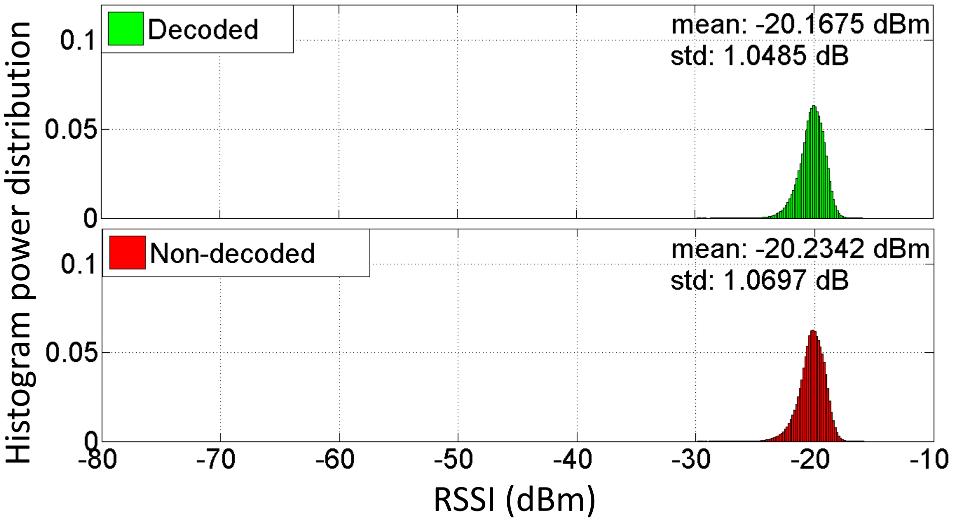


Figure 4.12: Distributions of decoded and non-decoded downlink messages.

casionally lead to wrong calculations of the burst start. Second, bad shift-register initialization, i.e., missing the first NB of a message, may occur. Although, the performance of the passive receiver could be improved, it already offers a reliable base for GSM signal capturing. There is, however, a distinct advantage with the proposed passive receiver over Airprobe to capture uplink messages.

Uplink Capturing Evaluation

Based on the downlink tests, we expected an equally good message recovery in the uplink. However, our in-house tests on uplink channels showed wider RSSI variation than on downlink channels. Our proposed receiver captures nearly 70% of generated packets by a MD Android application. To investigate the impact of this variation, we conducted more detailed evaluations on a Sunrise uplink channel. We present the findings of an eight-hour-long experiment.

Table 4.1: Success rate of passive receiver

Downlink message type	Passive receiver	Airprobe	Ratio (%)
Paging request type 1	61911	62581	98.8
Paging request type 2	19083	19262	99.0
Paging request type 3	17032	17206	99.0
RR System Info1	1477	1493	98.9
RR System Info3C	2955	2987	98.9
RR System Info4	2957	2986	99.0
RR immediate Assignment	6881	6973	98.6

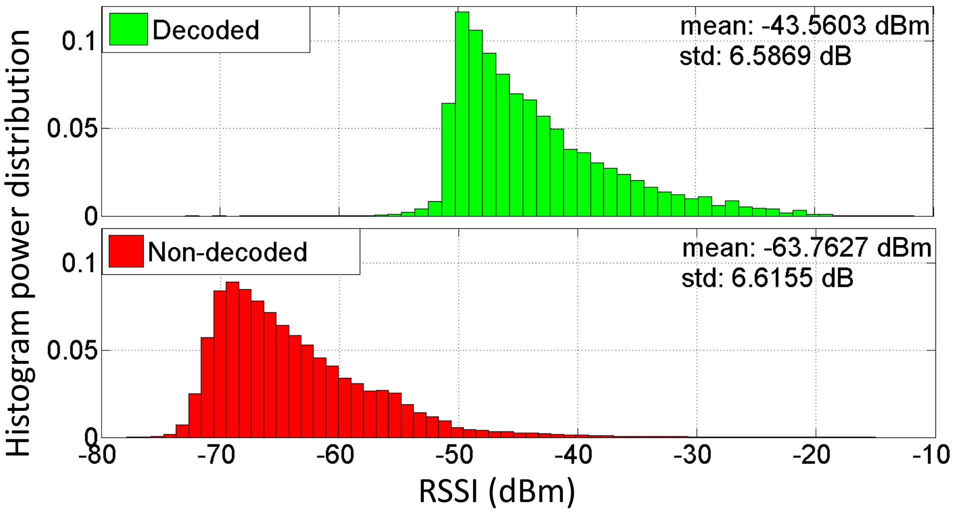


Figure 4.13: Distributions of decoded and non-decoded uplink messages.

Figure 4.13 plots the RSSI empirical histograms for decoded and non-decoded messages. These RSSI values belong to all MDs transmitting on one uplink channel (C0 - 45 MHz). Note that both empirical histograms in Figure 4.13 are normalized to the number of captured messages in each distribution. Figure 4.14 shows the combined distribution for both decoded and undecoded messages. From Figures 4.13 and 4.14, there is a threshold (around -55dBm, see decoded distribution) below which messages are rarely decoded. Note that the performance of our proposed system considers only messages with RSSI value above -63 dBm as illustrated in Figure 4.14. Moreover, we can indeed see that the RSSI variation in each uplink distribution is much larger compared to the downlink in Figure 4.12. On the uplink, the transmitters (MDs) have different distances to the passive receiver. In combination with MD mobility and changing multipath propagation, the effect is a variation in RSSI. Lower RSSI values compared to the downlink measurements are explained by the typically lower transmit power used by MDs in contrast to an electricity-powered BTS. Furthermore, a passive receiver has less radio visibility to the MDs than a BTS and experiences worse propagation conditions. In addition to the RSSI effect, the different distances of the MDs may occasionally cause time overlapping of their messages at the receiver, thus affecting the success of message decoding. Such an overlap is inherited to the system and cannot be compensated.

From the above analysis of power variation between captured downlink and uplink messages, we can explain the different capturing success rates of downlink and uplink channels. The captured downlink broadcast channels (called C0) transmit equal power over all TSs. Hence, the amount of received power at USRP terminal will almost be equal over all TSs. If the SNIR threshold of one TS is higher than the minimum decoding threshold, we guarantee to decode all transmitted messages over all TSs. However, due to the high variation of received power

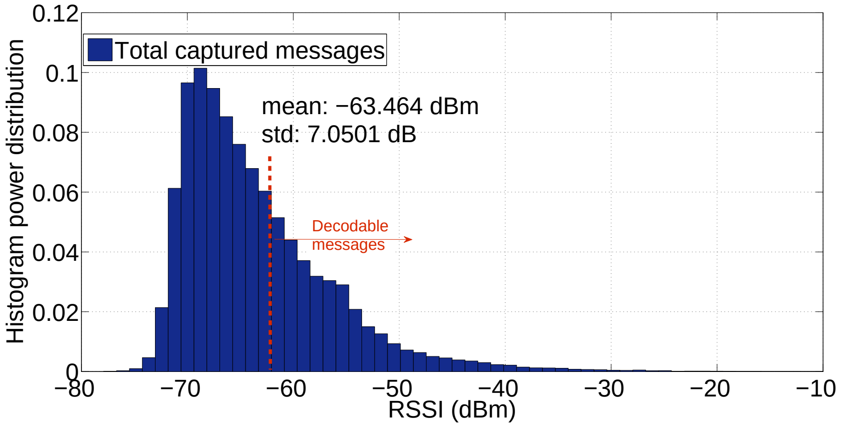


Figure 4.14: Combined distributions of decoded and non-decoded uplink messages.

at different TSs within one uplink TDMA frame, the interference level increases with the increase of instantaneous power difference between the maximum and the minimum received power on the eight TSs. Hence, we expect to have high capturing rates if MDs are located at an equal distance with respect to the USRP, which will decrease the interference level to the minimum. More detailed explanation of the problem is presented in Section 4.4.

Impact of Power Gain on Captured Messages

In an attempt to improve the message capturing rate, we conducted investigations with changing receiver power gain. The total power gain can be seen as the composition of two gain components: analog gain from the hardware (USRP) and digital gain from the software (GNURadio). First, we gradually increased the analog gain from 0 dB to 36 dB using the GR command *usrp->set_gain (gain, channel = 0)*, where *usrp* is a GR module containing the USRP object. Next, with 32dB analog gain, we increased the digital gain up to 17.7 dB. Figure 4.15 shows the averaged number of received messages as the total power gain increases. Although increasing the power gain benefits to message recovery, it should be carefully used because operating the USRP at maximum power gain for long periods may cause hardware failures. Therefore, we recommend selecting the analog gain close to 90% of the maximum and gradually adjusting the digital gain.

Unencrypted Message Distribution

In addition to analyzing power levels, we also evaluated the type of messages that we can recover. We built a GSM parser for most interesting uplink messages (specifically, Paging Response and Location Update Request messages). The parser can also detect encrypted messages without decrypting them. We aim to

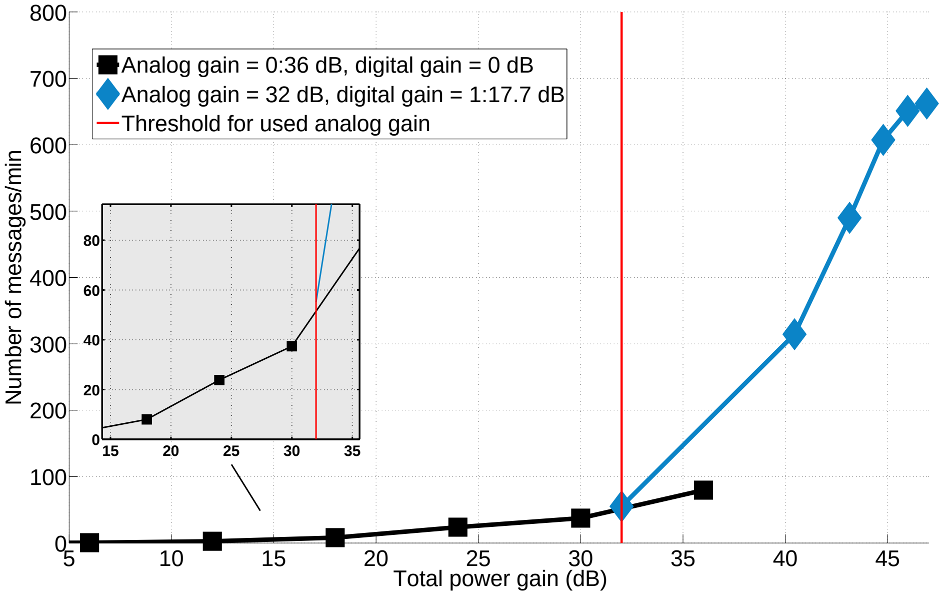


Figure 4.15: Power gain relation to the number of decoded messages per cell.

gain insight into the number of unencrypted identified messages, i.e., the messages containing the MD identity. Figure 4.16 shows the defined messages distribution over a working day period. Note that each bar in Figure 4.16 represents measurements for three hours. The distribution reflects identified MD's activity between day and night. Note that these measurements were performed in an uncontrolled real environment.

> The distribution of identified MD's messages depends completely on users' activities in a certain geographical area. Moreover, the ratio between encrypted and unencrypted messages depends on the network configuration *and* users' activities.

From the previous set of experiments, our proposed passive receiver showed reliable performance in capturing uplink and downlink messages. The receiver is also able to parse unencrypted messages and extract their MD identity for any further service development. The next research step seeks a wideband capturing solution that allows us to retrieve later the individual single-band channels.

4.3 Wideband GSM Cannelizer

As mentioned in Section 2.3.2, GSM is a FDM and TDM-based technology. RF channels are reused in a particular predefined pattern among cells of a certain geographical area. Moreover, within the same cell, communication between MDs and the serving BTS takes place over multiple RF channels. To expand the business model of our identification and localization services, it is an efficient approach to

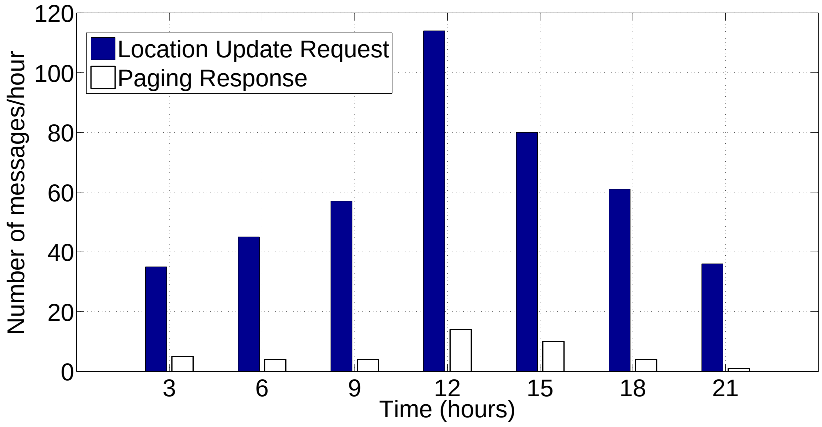


Figure 4.16: Unencrypted messages distribution over a working day period.

overhear radio signals over different RF channels using a single SDR system, what we call now, a wideband receiver. While a wideband SDR receiver offers many advantages in terms of costs and reconfigurability, it sets several challenges to signal capturing. It is required first to use a channelizer, which splits in real-time the wideband spectrum into a set of independent channels for later processing as illustrated in Figure 4.17. Note that the design of our wideband system is the same for both downlink and uplink bands. In this section, we investigate the performance of a wideband channelizer as a GR module. We discuss in Section 4.4 possible scenarios for a wideband GSM receiver and how to distribute the processing workload of the channelized streams.

The contribution of this subsection is optimizing an existing GR CPU-based channelizer using advanced CPU instructions and implementing a new GPU-based channelizer suited for GR (c.f., Sections 4.3.1 and 4.3.2). An evaluation of the proposed solutions as separate GR modules is presented in Section 4.3.3 [20].

Wideband Channelizer

To channelize a signal at a very high sampling rate and in real-time, we require an efficient implementation of a Polyphase Filterbank (PFB) channelizer. PFBs are a very powerful set of GR tools that (i) split the wideband signal into equispaced channels and (ii) filter the split channels with a given filter. GR includes a CPU-based channelizer implementation, which can take advantage of CPU architecture specific SIMD operations. Given the high-throughput and low-latency requirements of the channelizer, its implementation cannot be efficiently distributed to different processing machines over an IP network. At very high sampling rate, current GR implementation of the PFB channelizer might not support real-time processing of data. Hence, the need becomes evident for alternatively optimized (or newly implemented) PFB solutions within the same processing platform.

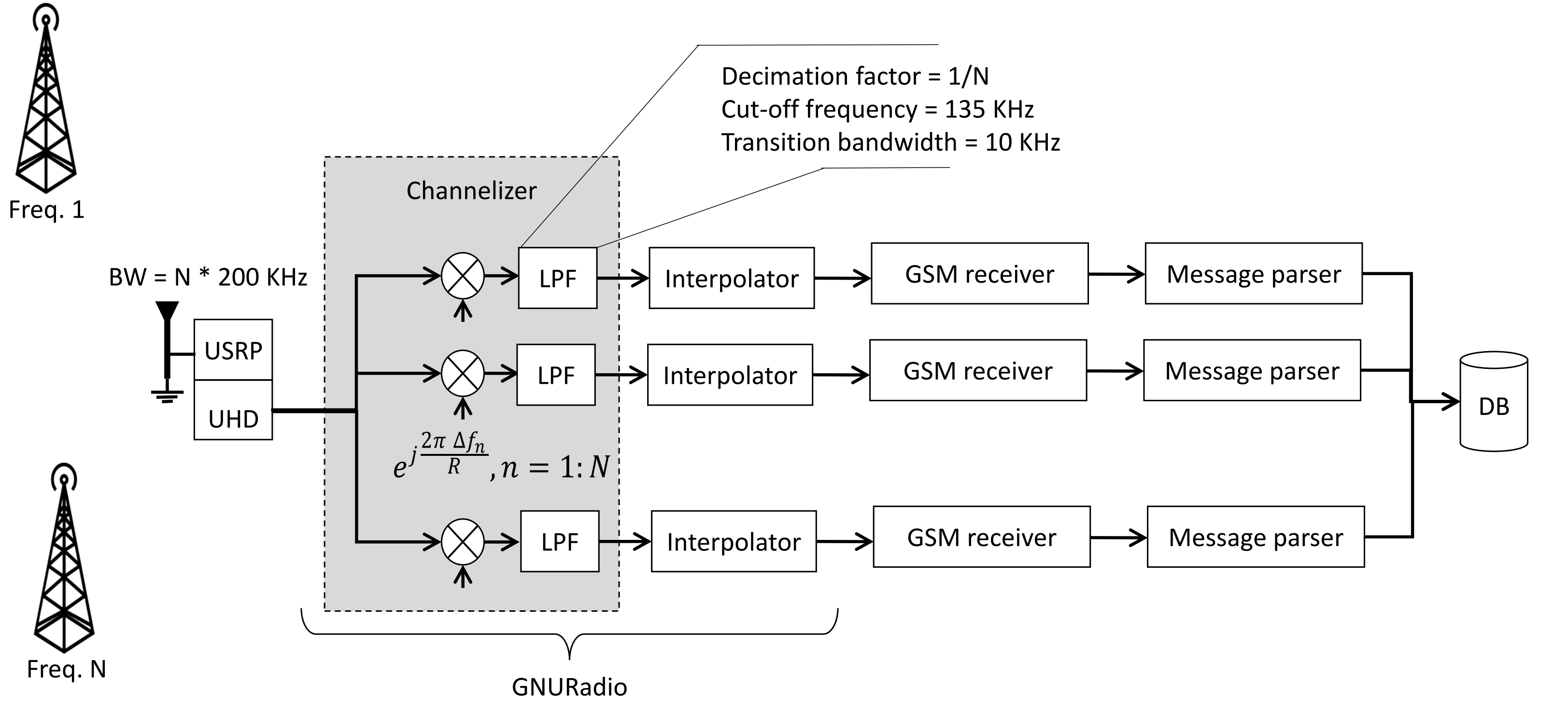


Figure 4.17: Schematic diagram of a wideband GSM receiver.

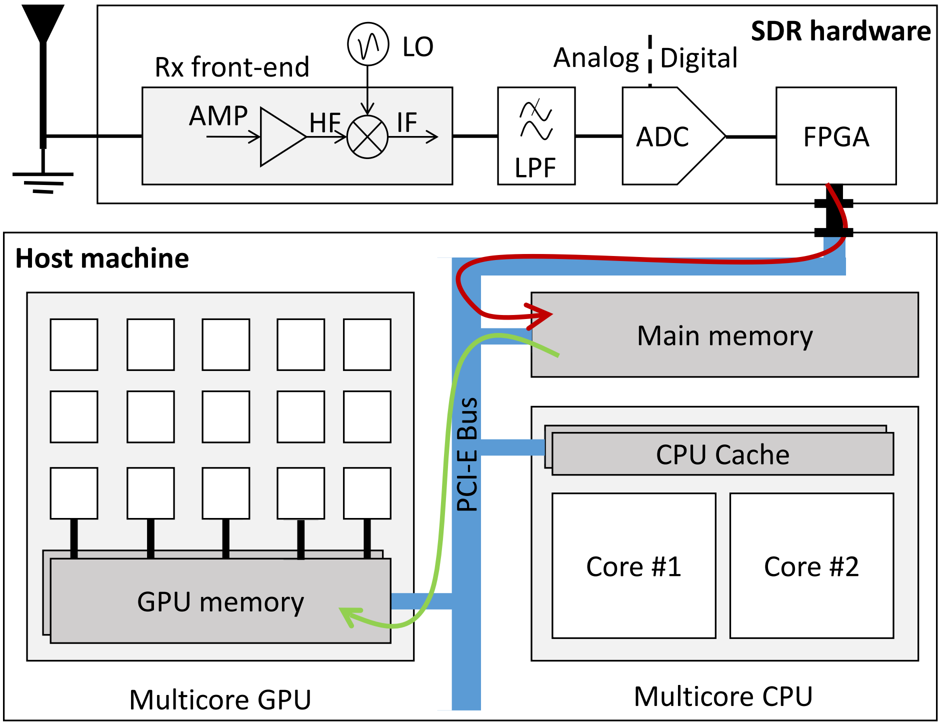


Figure 4.18: USRP receiver front-end architecture.

GPUs are fine grain SIMD processing units. They contain a large number of computing units. Hence, they can be used to implement a high-performance channelizer. However, to integrate such solutions with the GR framework, it is mandatory to implement the channelizer as a GR module. An essential requirement is the ability to transfer data between CPU (where GR works) and the GPU (where the channelizer can be implemented) at high bandwidth and low latency.

As shown in Figure 4.18, existing communication methods that transfer data through CPU memory to the GPU might be used (green line). However, data must cross the PCI Express (PCIe) switch twice between main memory and GPU memory.

4.3.1 Optimized CPU-based PFB

Pass filters (in the frequency-domain) are typically implemented using FIR filters [87]. FIR filters are faster when having a small number N of taps. As N increases (until a certain length), the signal-to-sampling-noise ratio also increases and the bit-error rate decreases. We can analyze the complexity of different channelizing implementations using an example of 5 GSM channels sampled at 1.35 Msps.

The first channelizing method is an N-taps FIR filter, which performs 2 + 4(N - 1) floating point operations per second (FLOPs) [177]. FLOPs do not measure the overall performance accurately as there are other types of operations involved; mainly data transfers and integer arithmetic instructions (loop(s) indices, pointers arithmetic). However, floating-point operations are the most computationally intensive operations. In case of wideband FIR filter, we have to handle $R * M$(2 + 4(N - 1)) FLOPs. Note that we have to use more taps when dealing with more channels due to the wider input bandwidth.

The second channelizing method is PFB. In case of PFB, each FIR filter y_m, where m = 0:M-1, has $N_m = N/M$ taps. The cumulated complexity of the PFB filters is hence $R * M$(2 + 4(N/M - 1)) = R (4N - 2M) FLOPs [177]. The number of FLOPs performed by an FFT is significantly lower than the filterbank, especially for a large number of taps [57]. Thus, PFB filter is more computationally efficient than FIR filter and will be considered in our proposed solution for wideband signal channelizing and filtering operation.

The FFT operations are computed through the fftw3 library [74]. This open source library takes advantage of modern architectures by making use of specific SIMD operations. FIR filters are accumulators that perform a lot of multiply and add operations. Thus, the PFB channelizer relies heavily on the VOLK multiply-and-add routine and most of the CPU cycles are spent inside this function [159]. The fastest multiply-and-add routine available in the VOLK machine for our CPU (Intel Haswell) is `volk_32fc_32f_dot_prod_32fc_a_avx` using the Advanced Vector Extensions (AVX) instruction set [124]. In the following, we present a set of optimizations by the available AVX PFB implementation.

Memory Alignment: CPUs access memory in chunks. Depending on the memory access granularity, a CPU may retrieve data in 2, 4, 8, 16, 32-bytes chunks or even more. Depending on the particular instruction, it may or may not be allowed to load data from memory into registers using addresses not being multiple of the memory access granularity [3]. Indeed, if allowed, loading unaligned data requires more operations as the register will be a compound of two different merged memory chunks. AVX are extensions to the x86 instruction set. They allow operations on 256-bit registers, enabling eight single-precision floating point numbers to be stored side-by-side in the same register and processed simultaneously. AVX uses relaxed memory alignment requirements. It means that using unaligned data is allowed for most instructions, but this comes with a performance penalty. When allocating memory using `malloc()`, compilers are in charge of the alignment, and we cannot assume that the returned address is aligned. However, it is possible to force alignment when desirable. GR provides such facility through the `volk_malloc()` function [81]. Such a function allocates a slightly bigger amount of memory (desired size + granularity - 1) and moves the pointer to the next aligned address. However, the FIR filters from the PFB channelizer

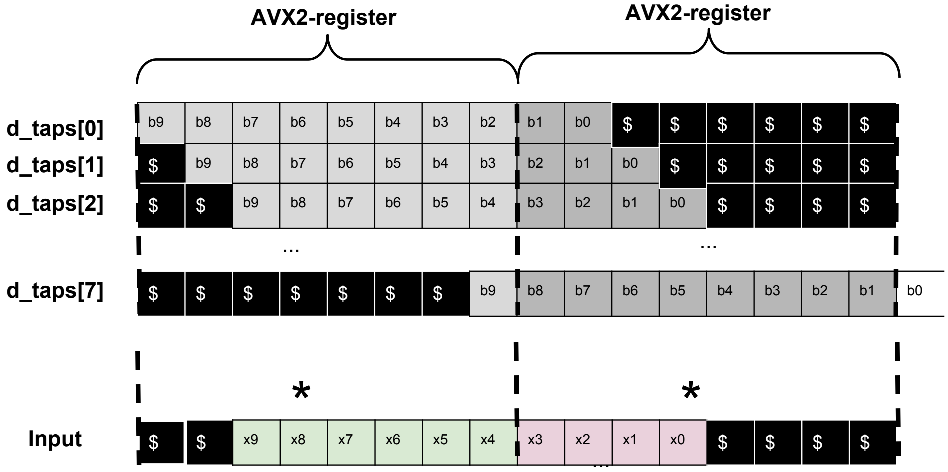

AVX2-register
AVX2-register
d_taps[0]
b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 $ $ $ $ $ $
d_taps[1]
$ b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 $ $ $ $ $
d_taps[2]
$ $ b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 $ $ $ $
...
...
d_taps[7]
$ $ $ $ $ $ $ b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
*
*
Input
$ $ x9 x8 x7 x6 x5 x4 x3 x2 x1 x0 $ $ $ $

Figure 4.19: Aligned filter taps in GR implementation.

cannot use this function because GR buffers are not necessarily aligned. Instead, GR generates a set of aligned taps d_taps for each possible architecture alignment as shown in Figure 4.19. The number of taps in Figure 4.19 is just an example for illustration. By measuring the alignment of the input samples, we can select the correctly aligned taps and always use aligned data. For example, if the input register is aligned with two floats, GR will choose to multiply with d_taps[2] (a set of taps aligned with 2 floats). GR performs SIMD operations on aligned data and non-SIMD operations on the remaining input samples. To avoid non-SIMD operations, one solution would be to make the number of filter taps a multiple of 8. Then, we could fill taps corresponding to unaligned data with zeros as shown in Figure 4.19. We could then consider the data aligned in GR buffers because zeros would discard unaligned data input values.

Fused Multiply-Add (FMA) operations: FMA operations are an AVX2 extension to 128 and 256-bit Streaming SIMD Extensions (SSE) instructions [103]. AVX2 FMA are floating point operations similar to c ← c + (a * b) that are performed in one step as shown in Figure 4.20. These operations are compatible with the GNU compiler collection. However, AVX performs (a * b) + c in two steps (first a * b, then + c). Benchmarks of pure multiply and add operations on Intel Haswell CPU show a double increase of FLOPs per cycle (double performance) using AVX2 FMA compared to AVX multiply then add.

The PFB implementation uses the product of complex and floats vectors. This implementation requires some further operations like deinterleaving an input block of complex samples into M outputs. However, when compiling fftw3 using `---enable-fma` option, we experience an adverse impact on performance. It means that this library is not yet optimized for AVX2 FMA instructions. Hence, we will use the library without the AVX2 FMA optimizations.

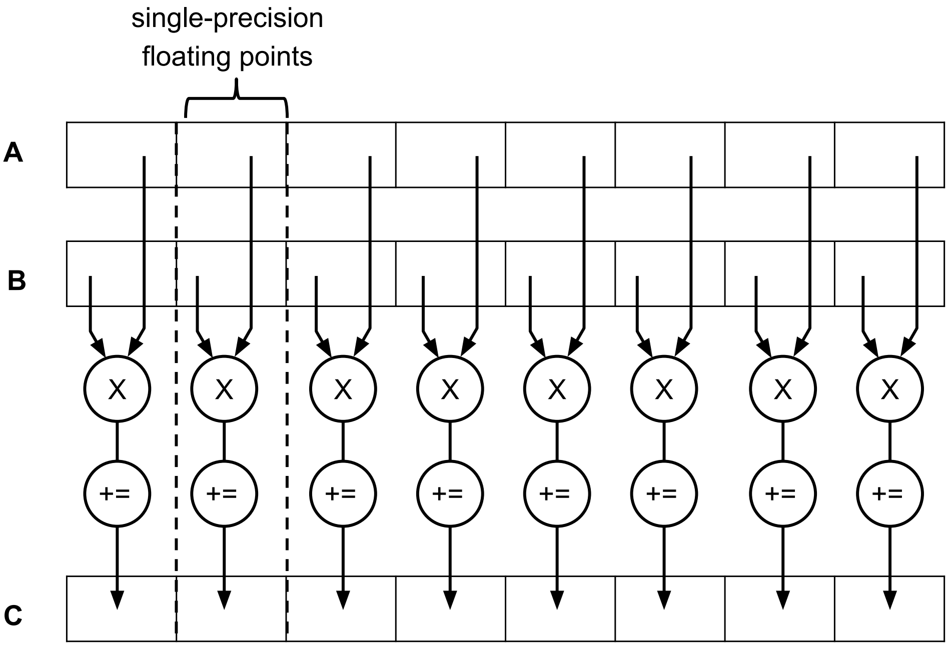


Figure 4.20: GR FIR AVX dot product.

Better Pipelining: complex numbers are just two consecutive floats as defined in `complex.h` of the C standard library. From the previous example c ← c + (a * b), a is a complex number representing an input sample and b is an integer representing a FIR filter coefficient. Consider that a is a tuple of (r , qi): r is the in-phase part of a, q is the quadrature part of a, and i is the complex number $\sqrt{-1}$. Complex arithmetic rules define a*b = (r + qi) * b = (r + qi) * (b + 0i) = (rb + qbi). As the GR FIR filters are not specifically designed for AVX operations, the coefficients are just stored as consecutive floats. Unfortunately, feeding 256-bit registers with consecutive floats is a costly process because it requires a for-loop overall input sample to perform the (r + qi) * b instructions. The best we can do is to create a more compatible layout of filter taps to benefit from AVX2-FMA operations. To do this we load (`_mm256_loadu_ps`), unpack (`_mm256_unpackXY_ps`) and then permute (`_mm256_permute2f128`) FIR coefficients before operations. The behavior of unpack does not allow to permute coefficients across 128-bit boundaries. This behavior is inherited directly from the legacy SSE unpack operation. However, by re-shuffling the taps during the creation of the PFB as illustrated in Figure 4.21, we can create a layout that enables the taps to be in the right order when unpacked (without permutation). Then, we can perform four (r + qi) * b instructions in one step.

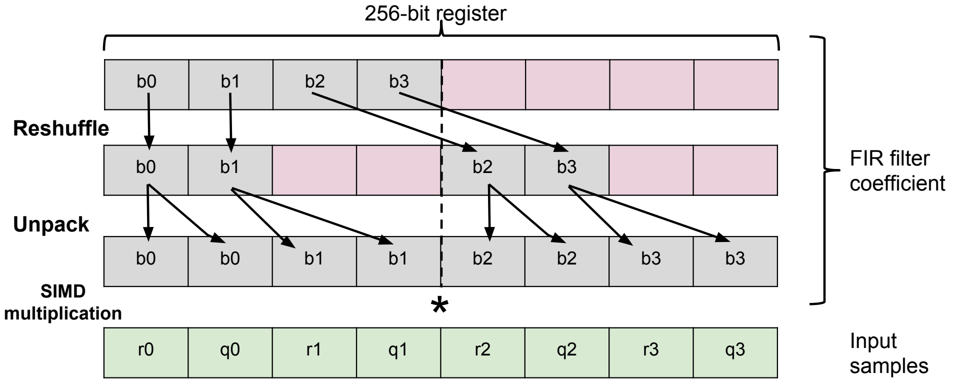


Figure 4.21: AVX unpack operation and multiplication.

4.3.2 GPU-based PFB

We can make PFB processing faster by taking advantage of massively parallel architectures, such as GPUs [155]. GPUs are not only able to perform a fixed set of graphics-related operations, but also general-purpose processing as we would expect from a GPP like a CPU. There are several technologies available to perform GPU processing. OpenCL is an open framework for heterogeneous computing launched in 2009 by Khronos and co-designed by Apple [175]. It is open by design and, therefore, best suited to the philosophy of GR. For this reason, we will implement a PFB channelizer using OpenCL.

In our experimental setup, we are using a discrete graphics card, meaning that the CPU and GPU do not share the same memory. There are several techniques to transfer data between the CPU and GPU, with various transfer speeds [136]. However, when using physically shared memory using fused architectures, e.g., using Intel IGP, we are not required to transfer data between the GPU and CPU. For discrete GPU, the data could be copied using the CPU or using GPU direct memory access (DMA). We want to be able to process data in real-time. Therefore, we have strict requirements for transfer speeds. Since GR has no built-in support for coprocessors, buffers could only be transferred to/from the GPU inside the `work()` function. Every time `work()` is called, we should (i) transfer data from the main memory to the GPU, (ii) perform the computation on the device and (iii) transfer the result from the device to the main memory. This situation is not optimal as the (i) and (iii) data transfers have high latency but are used to transfer few data. For real-time processing, data transfer between CPU and GPU is a time critical action. The time spent transferring data is time lost for processing. It is important to obtain a balance between transfer delay and processing delay. As a solution to this situation, we propose the following: First, to avoid inefficient data transfer to a single port, we deinterleave the input block of complex samples in front of the PFB channelizer inside `work()`. This is simply done using `stream_to_streams` (s2ss) function, which converts a stream of M items

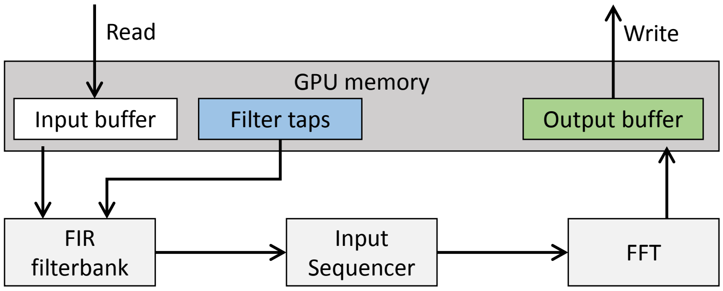


Figure 4.22: PFB Channelizer on GPU.

into M streams of 1 item. Hence, we get the inputs of the FIR filters on different ports with different buffers and transfer data more efficiently. Second, to avoid inefficient data transfer (small amount of data with high transferring latency), we batch the s2ss transferring buffers using `clEnqueueWriteBuffer()` with a non-blocking successive calls. With the non-blocking write transfer, control is returned immediately to the host thread, allowing operations to occur concurrently in the host thread during the transfer between host and device proceeds. On the GPU device, as illustrated in Figure 4.22, our filter taps are copied to the GPU memory once at the startup of our PFB channelizer. Each GPU FIR filter (inside the PFB channelizer) runs in parallel inside an independent GPU computing unit. Once we get the output of the FIR filters, we pass it to the sequencer so that we can perform in-place the M-point FFT. There are several OpenCL FFT libraries available like AMD clFFT (an open source library). clFFT usage is straightforward. Here, we take advantage of the functions `bakePlan()` to apply device optimizations and `setPlanBatchSize()` to handle several plans concurrently, thus minimizing OpenCL API calls.

4.3.3 Experimental Setup

In this section, we consider the GR CPU-based implementation of the PFB channelizer as a reference. We are using GR version 3.7.5. All experiments are performed on a Linux-based machine with Intel Core i7-4790 @ 3.60 GHz CPU, AMD R9 290 (Tahiti) GPU (4 GB video memory), and 32 GB RAM. We use a simple benchmark illustrated in Figure 4.23. The signal source is a GR block used to generate *continuously* a certain type of a signal, such as sine, cosine or constant. The input to the signal source is the signal type, sampling rate and amplitude. The output of the signal source is defined by its type. In this experiment, we used *gr_complex* defined in GR as an output to emulate the behaviour of a USRP (c.f. Section 2.7.2). A signal source has an advantage compared to a USRP N210 because it can generate signals with sampling rate more than what a GbE cable can carry. Note that a signal source does not define the number of generated samples N. To define this parameter, we used a GR head block. The head block stops passing samples from the signal source to the next block when N items of the gr_complex

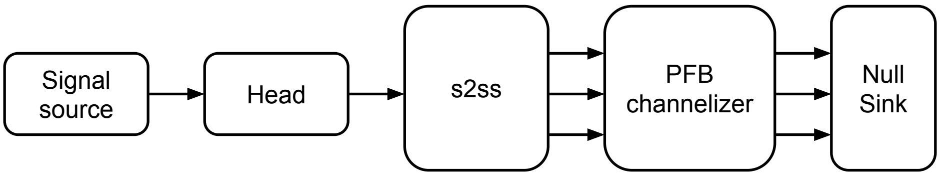


Figure 4.23: PFB channelizer benchmark overview.

type are processed. Hence, we can configure control experiments duration through N. The s2ss block performs the FDM operation in Figure 2.19. We used three implementations for the PFB channelizer: (i) original GR with AVX instructions, (ii) our optimized GR implementation with AVX2-FMA instructions, and (iii) our GPU implementation. The channelized streams are sent to a null sink because our target is to evaluate the performance of the PFB channelizer only.

A Running Example

To measure the performance of our proposed wideband channelizer in Sections 4.3.1-4.3.2, we will use a running example based on our proposed Passive GSM receiver presented in Section 4.2.2. Each GSM receiver requires a precise sample rate R for each GSM channel: $R = 1625000/6 \simeq 271\text{x}10^3$ *complex* samples per second (sps), which is equivalent to the data rate of the GSM signal itself. From these numbers, we can calculate a few elements for a wideband GSM receiver:

- We need a wideband sampling rate $R_b = R * M_{\text{ch}}$, where M_{ch} is the number of GSM channels.

- We can capture up to M_{ch} = 25x$10^6/R$ = 92 channels using N210 with 16 bit precision.

AVX vs AVX2-FMA vs GPU

In these experiments, we want to measure the improvement provided by the optimized AVX2-FMA PFB (presented in Section 4.3.1) and the proposed GPU-based PFB (shown in Section 4.3.2) compared to the default GR AVX implementation. We use a pre-designed filter of 55 taps, and we want to test the performance for different numbers of channels. We generate a signal equivalent to 60s of data capture for an appropriate number of channels. Our benchmarks output the total processing time *measured* to channelize the 60s of data. The presented results in Figure 4.24 are the arithmetic mean of 5 experiments for each number of channels and PFB implementation. If the processing time exceeds the 60s, it means that the PFB channelizer cannot process data in real-time. As illustrated in Figure 4.24, the improvement using our optimized AVX2-FMA ranges from 30% to 33%. Our optimized filter can process up to 76 GSM channels with processing time less than the 60s. This means that the maximum number of channels that an AVX2-FMA

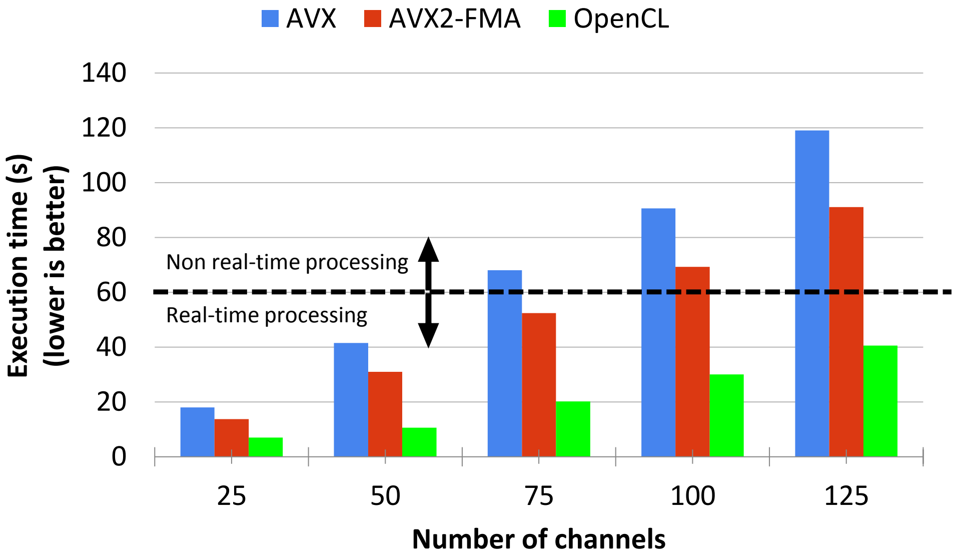


Figure 4.24: PFB channelizer performance vs. number of channels.

PFB channelizer can support in real-time is 76. It is, however still, lower than what a USRP N210 device can support (92 channels). Compared to the reference channelizer, there is an average improvement of 3.2-fold by the GPU-based PFB implementation. That means we can channelize more than 125 GSM channels, more than what the USRP is able to capture.

Taps per Filter

In this experiment, we run the same benchmark as in Section 4.3.3 with a constant number of channels equal to 25, but varying the number of taps per FIR filter (channel). As illustrated in Figure 4.25, the execution time increases with the increasing number of FIR taps. However, we see the zig-zag behavior for AVX and AVX2-FMA implementations. This is because AVX2 registers are double the size of AVX registers. Extra samples in both implementations are processed using standard non-SIMD operations. This behavior is not noticed in the OpenCL implementation due to the different mechanisms for data transfer from/to GPU. Note that it is important to use a multiple of 2^N so that the compiler can optimize the costly division operation to a bit shifting operation.

Single-Machine Performance

To validate our proposed filters using real-time experimentation, we considered our proposed passive GSM receiver with a wideband signal. Using the same setup expressed in Figure 4.17, we used our GPU-based implementation of the PFB channelizer. The channelized streams were connected to set of GSM passive receivers (CPU-based). The USRP N210 device was tuned to the Sunrise channel at 949

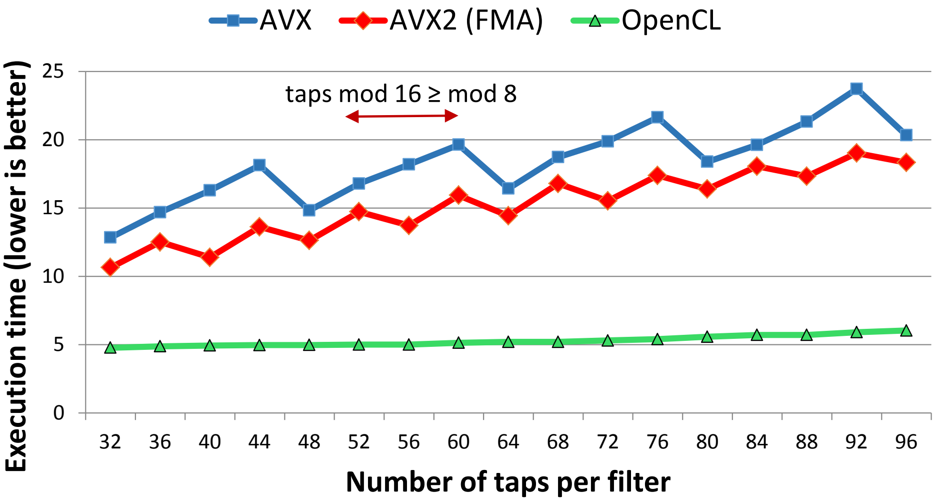


Figure 4.25: PFB channelizer performance vs. number of taps.

MHz as expressed in Section 4.2.3. We would like to analyze here the maximum number of channels by the CPU. Note that the PFB channelizer deals with a set of channels next to each other in the frequency domain. Hence, the number of captured messages depends highly on the availability of downlink channels around 949 MHz. The result expressed in Figure 4.26 shows the CPU usage as a number of channels. The maximum number of channels in raw that our machine (equipped with Intel Core i7-4790 @ 3.60 GHz CPU) can process in real-time is 15 channels. However, the USRP can support up to 92 channels. Hence, analyzing channels with the maximum USRP load has to distribute the load over an IP network to a set of processing machines.

4.3.4 Conclusions

GR is the framework of choice for many SDR projects. We demonstrated that it was possible to create high performance, reusable components for wideband applications. The channelizer technique is applicable for any distributed system that compose a set of independent channels. The polyphase filterbank channelizer is not only useful for FDD-based technologies but is a critical component in many SDR systems. Our AVX2 optimizations enhance the performance with up to 30 % using recent CPU features, namely fused multiply-add operation. Moreover, the GPU implementation opens up opportunities for applications were expensive specialized hardware was previously required. The GPU-based implementation is a GR-like module with 3.2-fold faster with respect to the PFB processing time when compared to the original GR implementation. Our proposed solutions are in the form of independent GR modules [38].

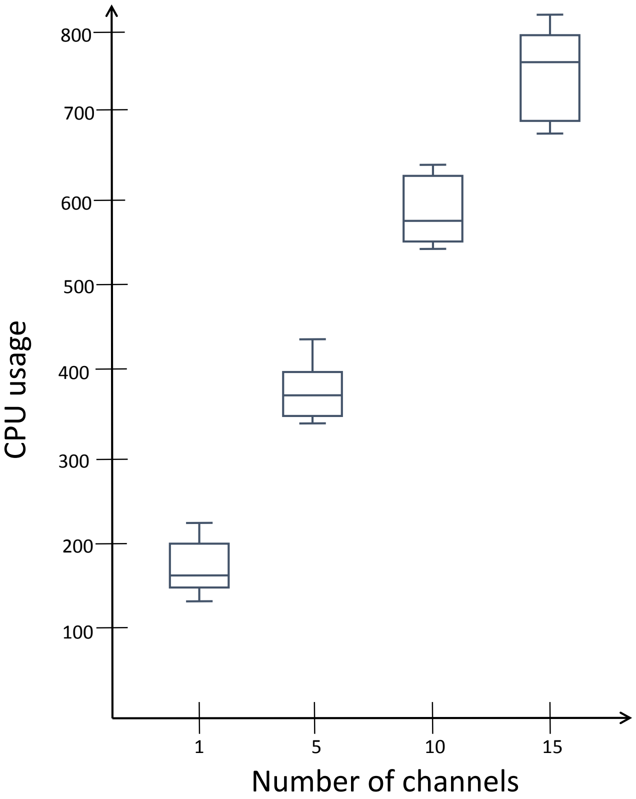


Figure 4.26: CPU performance of a single machine using PFB GPU channelizer.

4.4 Distributed Wideband Receiver

Up to this point, we have presented our proposed solutions for a passive GSM receiver (uplink and downlink) as presented in Section 4.2. We have also presented an efficient and optimized channelizer module that can isolate a wideband captured spectrum into equally spaced independent channels as presented in Section 4.3. However, as presented in Figure 4.24, the processing power of the machine hosting the PFB AVX2-FMA channelizer is completely occupied after 76 channels. It means that the hosting machine does not have any processing power to process the channelized streams. This section moves on and discusses challenges and possible scenarios for successful *decoding* of a channelized wideband spectrum using our proposed passive GSM receiver. Then, it moves to propose a distributed system that is capable of handling the process of high number of channelized signals that exceed the processing power of a single machine.

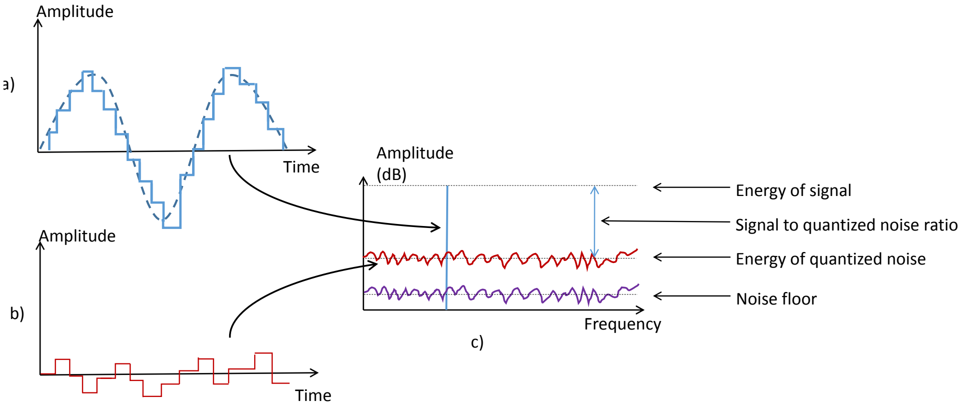


Figure 4.27: a) Analog and quantized signal in the time domain. b) Quantized noise in the time domain. c) Signal and quantized noise in the frequency domain.

4.4.1 Challenges in Wideband Capturing

We discussed briefly in Section 4.2.3 how uncontrolled variation of received power within TSs of one TDMA frame increases the interference and decreases the decoding rate. This problem did not appear in downlink capturing (of a single channel) because a GSM BTS transmits the equal amount of power over all time slots of one TDMA frame. Similar to the uplink problem of a single channel, downlink interference will increase in wideband capturing when the ratio between the highest signal channel and the lowest signal channel increases. The interference problem has two main reasons: (i) lack of control on transmitted and received power of captured messages and (ii) the design of current SDR systems.

First, let us understand a bit the power control algorithm inside the GSM network. A GSM BTS controls transmitted power of all served MDs such that SNIR at the BTS from all MDs exceed the minimum decoding threshold. This operation is done per TDMA frame inside a particular RF channel. Hence, the ADC inside the BTS deals with received signals of comparable power strength. Figure 4.27-a shows a quantized signal at the time. The rounding error introduced by the quantization process is called the quantization noise as illustrated in Figure 4.27-b. Figure 4.27-c shows the energy of the target signal and the quantized noise in the frequency domain. The ratio between the energy of the target signal and the energy of the quantized noise is called Signal-to-Quantized-Noise Ratio (SQNR). In theory, SQNR in dB of a radio signal is given by Equation 4.3 [151].

$$\mathrm{SNR} = 6.02N + 1.76 \tag{4.3}$$

N is the number of bits (resolution) of the ADC. The USRP N210 has 14-bit ADC, but only 12 bits are effectively used for quantization [71]. From Equation 4.27, the USRP N210 ADC has a Signal-to-Noise Ratio (SNR) equal to 74 dB. This means that the energy of the quantized signal is 74 dB higher than the energy of quan-

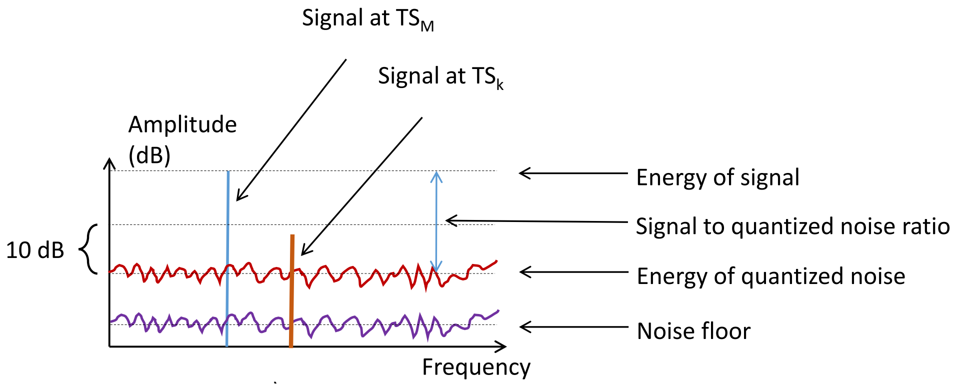


Figure 4.28: High interference for low power signals.

tized noise. Recall that we need SNIR of around 10 dB for a success decoding of GSM messages. This means when capturing multiple signals (over multiple TSs of one TDMA frame) that RF signals with energy of 74 - 10 = 64 dB lower than the highest energy signal will not be decoded as explained in more details in Figure 4.28. This is because the quantized noise of the highest signal power dominates. In other words, for our proposed passive receiver without control on the transmitted power, high power signals will contaminate low power signals. For example, the signal at TS$_K$ might be located below the quantized noise of the signal at TS$_M$ and hence, the lower signal is not decodable. Moreover, quantized noise for a signal is considered as interference for another signal. The more signals we include in the ADC, the more interference we have and the lower decoding rate we get. From this observation, we come to the problem of a wideband passive receiver. The highest BTS signal power will contaminate those signals with the difference in their power more than 64 dB with respect to the maximum signal power. In the city of Bern, Switzerland, we conducted real experiments to observe the difference of received power in the surrounding area. Figure 4.29 shows a set of captured downlink channels (different network operator). The results were obtained by scanning a wideband spectrum of 40 GSM channels that were passed later to our optimized channelizer. Figure 4.29 shows a difference of received power in the range of 90 dB. However, these results depend strongly on the location and configuration of surrounding BTSs. For example, we might observe this variation by scanning two GSM channels only.

Based on the above, we bring two points for discussion. First, do we still need the wideband GSM receiver for indoor localization? The answer is yes. The wideband passive GSM receiver will allow us to capture messages of GSM MDs regardless to which of the surrounding BTSs they will connect. For reasonable deployment density, e.g., 20m between USRPs, we expect the difference in power to be less than 64 dB. Second, where is the best location to gain fully from the wideband passive GSM receiver? As long as the system works in the passive mode

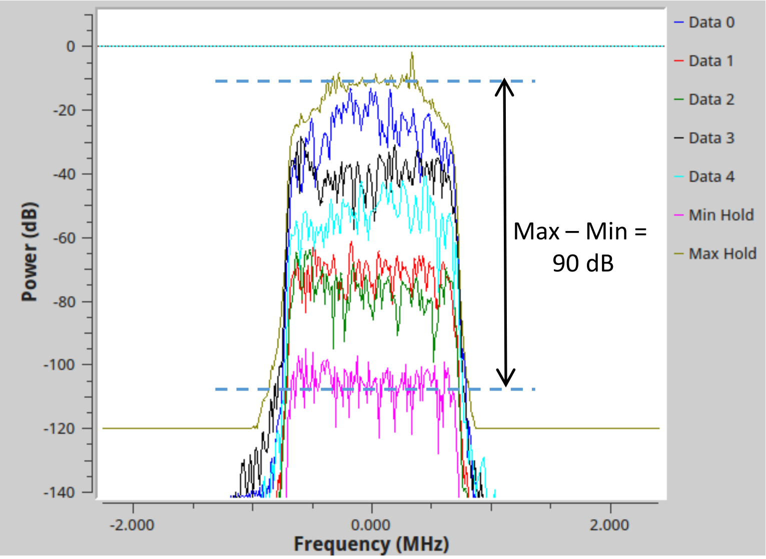


Figure 4.29: Downlink captured power at of a wideband receiver.

without control over GSM MDs' transmitted power, the best is to place our USRPs colocated (or as close as possible) with real GSM BTSs. In this case, we expect to capture all uplink messages of frequencies supported by each BTS individually. In the coming sections, we will discuss possibilities of analyzing channelized signals without the focus on constraints of geographical deployments.

4.4.2 Proposed Architecture for Distributed Signal Processing

In this section, we propose and implement a solution to distribute the workload of a broadband SDR receiver using an on-demand load-balancer dynamically. More specifically, we implement a load balancing system that distributes and balances the channelized spectrum according to available heterogeneous processing machines over an IP network. Moreover, we include on-the-fly signal compression for further optimization of the link capacity (c.f., Section 4.4.2).

In this section, we present our proposed architecture to optimize the GSM receiver and distribute the workload on other machines to support the maximum number of channels limited by the N210 specification. The simplest proposed solution is to broadcast the captured signals to a set of processing machines (workers). We call this solution the *broadcast approach* as illustrated in Figure 4.30. Every worker in the broadcast approach contains a channel filtering module and the proposed GSM receiver. Regardless the implementation of the filtering module (CPU-based or GPU-based), the broadcast approach has some drawbacks:

- The wideband signal (with high data rate) will pass through the switch connecting the USRP to the set of workers. To prevent the network from excessive traffic, which degrades the network performance, most switches im-

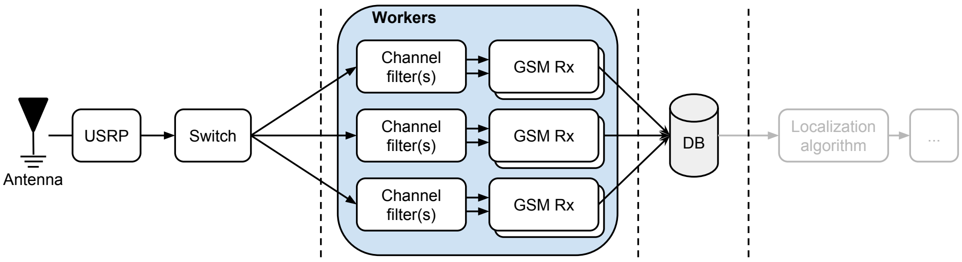


Figure 4.30: Broadcast approach for a wideband receiver.

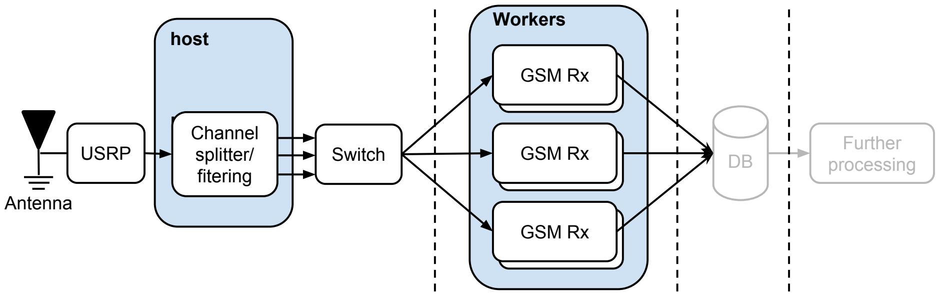


Figure 4.31: Unicast approach for a wideband receiver.

plement a special feature called storm-control [45]. Storm-control allows switches to monitor traffic level and to drop broadcast packets when the traffic level exceeds a certain predefined limit, called storm-control-level. Hence, the broadcast approach requires first to disable the storm feature to avoid dropping packets at high datarate.

- The received stream at each worker contains the wideband signal and each worker has to extract its own signal from it. This added load on each machine grows with captured signal bandwidth.

The other proposed solution is to channelize and filter the wideband signal before the switch at a central processing unit. We call this solution the *unicast approach* as illustrated in Figure 4.31. Every worker in the unicast approach contains only the proposed GSM receiver. However, the machine hosting the channelizer (host) needs to be fast enough to process samples faster than the R_b sampling rate. Otherwise, the UHD buffer will overflow, and the UHD will drop samples. The unicast approach expressed in Figure 4.31 removes the dependency on the network infrastructure compared to the broadcast approach. In both approaches, workers can be placed on one or multiple machines over an IP network.

4.4.3 Real-Time Load Balancer

To avoid a preconfigured approach between the channelizer and the set of workers (e.g., each worker is set to a specific channel), we chose to use a token-based approach: workers send a token when they are ready to process more channels to a central stable point inside the network. For the signaling between workers and the channelizer, we use the ZeroMQ messaging library and a broker to manage the tokens (c.f., Figure 4.32).

Broker: The broker, is a load balancer that distributes tokens. It is a stable point in the network, known as a device in the ZMQ terminology. When the broker is down, queues and items are lost. We implemented the broker in C++ similar to the ZMQ load balancer implementation. However, there is a fundamental difference between our proposed broker and the ZMQ load balancer: the latter uses Least Recently Used (LRU) queue to select a worker. This mechanism applies only to limited jobs in time. Upon task completion, the balancer takes a new job. However, our system is different as tasks are channel processing, which are not limited in time. We can fix this by turning the LRU-based queue into a token mechanism: requests do not represent jobs anymore, but session tokens.

TCP blocks: The most important part of workload distribution in real-time signal processing software is how to distribute the radio signal itself. Our application requires reliable transmissions, sessions as well as congestion control, so TCP is the natural fit over UDP. For our networking blocks, we decided to observe strictly the original API design of the GR TCP (Python) block so that it can be used on its own as a full-featured alternative. Here, the TCP source is acting as a server, and the TCP sink is acting as a client. Workers and clients respectively use them. Note that we give names based on GR terminology. Hence, the output of the channalizer goes to a sink (TCP sink) and the input of our workers is a source (TCP source).

Client: The client, is a GR block that manages the transition between the host machine and the workers by connecting TCP sinks to the appropriate worker endpoint. The client is directly instantiated with the number of input ports (channels) and the host machine(s) IP address. Note that the machine hosting the channelizer and clients have the UHD driver.

Worker: The worker role is different to that of a client. It has to distribute the tokens and to manage a set of sessions. In the GSM receiver, the session token is linked with session data, such as channel center frequency. We want our worker to have only one listening port, so we do not need a whole range of listening ports to be available and configured. Once a client gets a response from a worker, instead of connecting directly to a TCP source block, it connects to an endpoint running on a dedicated thread (listening thread). The stream always begins with a 32-bit number in TCP/IP network byte order that represents a token ID. Note that the set of machines hosting our workers do not have UHD driver. Instead, we are using TCP to transfer our channelized spectrum. Hence, the delay introduced by GbE cable *will not* interfere processing the streamed signal.

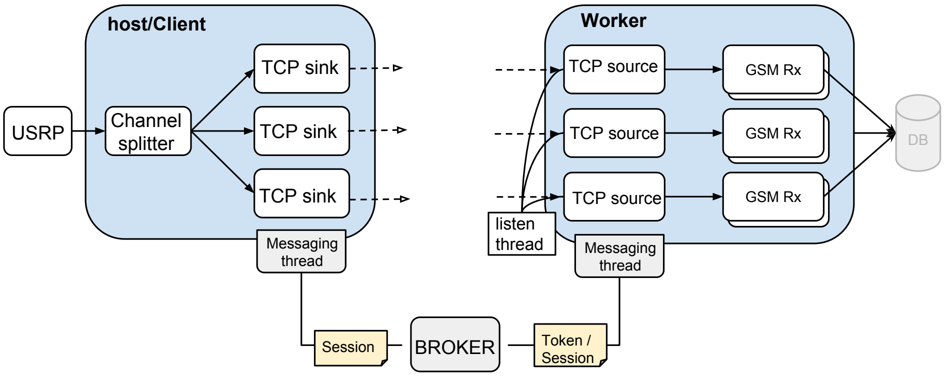


Figure 4.32: Client/Worker communication overview.

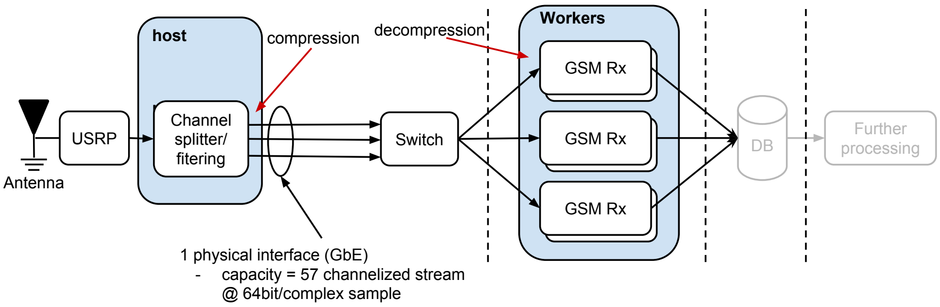


Figure 4.33: Compression scenario to increase link efficiency.

4.4.4 Signal Compression

GR does not provide any facility to handle data types of different precision. Most architectures have FPU for single-precision floats and converting data would introduce a massive performance penalty. However, when using an efficient channelizer, the system is now limited by the link capacity from the host machine (64-bit per complex sample) as shown in Figure 4.33. When using GbE, we are bounded to distribute roughly 57 GSM channels. This means that if we can channelize more than 57 channels, we cannot distribute them for processing. What we can do to improve the system without upgrading to a 10 GbE connectivity is signal compression. In our scenario described in Figure 4.33, we would apply the compression process after each channelized stream inside the host machine and apply the decompression step before the GSM-receiver at each worker. We implemented some optional (at compile-time) data compression inside our TCP source/sink blocks. Therefore, we expect to achieve significant saving space inside the GbE cable.

4.4.5 Experimental Setup

To validate our distributed architecture expressed in Figure 4.32 with the wideband passive GSM receiver system, we build a synthetic test using a uniform set of workers in a cloud environment (an Openstack server) as illustrated in Figure

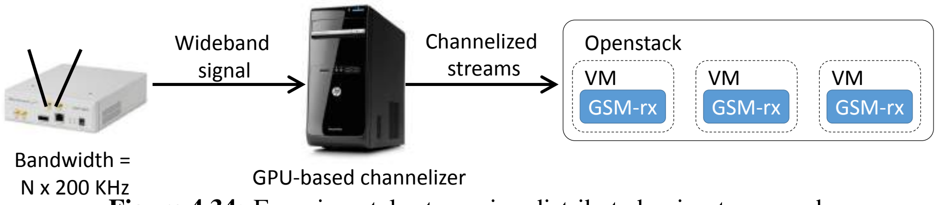


Figure 4.34: Experimental setup using distributed unicast approach.

4.34. We connect the USRP using a GbE connection to a host machine responsible to channelize and filter (GPU-based implementation) the wideband signal. The host machine is a Linux-based machine with Intel Core i7-4790 @ 3.60 GHz CPU, AMD R9 290 (Tahiti) GPU (4 GB video memory), and 32 GB RAM. The host machine is connected using another GbE connection to the same IP network of the Openstack server holding the GSM workers (VMs). Each worker (VM) inside the Openstack server has 1 vCPU (virtual CPU) and 1 GB RAM. Note that the Openstack server illustrated in Figure 4.34 contains 32 Intel(R) Xeon(R) CPU E5-2450 v2 @ 2.50GHz and 190 GB RAM.

However, due to the limited link capacity (up to 100 Mbps) between the host machine and the Openstack server, only three channelized streams (3 GSM channels) can be transferred from the channelizer host to the Openstanck server. Hence, we build another synthetic test using a uniform set of workers in another cloud environment (Amazon Elastic Compute Cloud [25]). We use a GR signal sources with *gr_complex* samples as an input source instead of the USRP. We created a set of 8 Amazon EC2 m3 large instance (2 vCPU) that are responsible to capture GSM messages out of the channelized signal. We call them Amazon workers for simplicity. Amazing workers rely on underlying Intel Xeon E5-2666 v3 operating at 2.6 GHz. For the host (client), we use one Amazon EC2 c4.4x large instance (16 vCPU with AVX2-FMA support). We call it Amazon client for simplicity. Amazon clients rely on Intel Xeon E5-2670 v2 operating at 2.5 GHz. Amazon provides instances for GPU compute applications, but they do not have enough dedicated throughput for our application.

4.4.6 Experimental Results

For the first experimental setup using a USRP and the Openstack server, we configured our system to channelize 3 GSM channels. Although the GPU-based channelizer can split more than 125 GSM channels, the link capacity cannot transmit more than 3. Note that workers in both setups do not use the UHD driver, messages are passed to them using TCP protocol. Hence, delays inside the cable will not affect the processing performance of workers. We ran an end-to-end experiment for one hour. Results illustrated in Figure 4.35 show the CPU usage inside a VM while processing a GSM channel with data, such as C0 broadcast channel, and

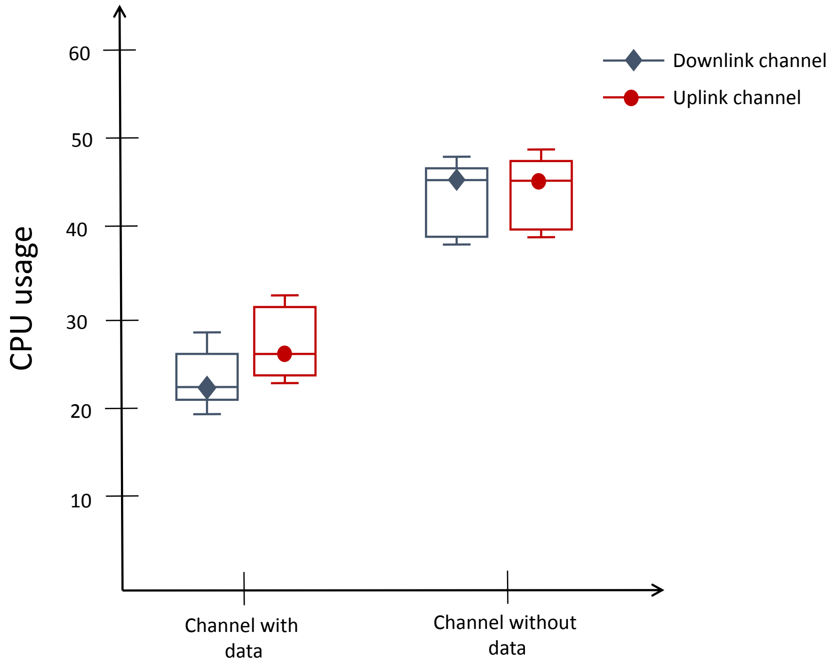


Figure 4.35: CPU usage of distributed unicast approach with a real-time GSM data from a USRP.

without data, such as an empty GSM channel. From Figure 4.35, uplink channels with data require slightly more processing power (27% of CPU usage) than downlink channels with data (22% of CPU usage). This is because the GSM receiver processing an uplink channel stays more in the unsynchronized state waiting for an uplink message, which is more processing consuming, than in the synchronized state. Whereas the GSM receiver stays more in the synchronized state than the unsynchronized while processing a downlink channel because bursts are transmitted on every time-slot. Moreover, processing uplink or downlink channels without data require more processing power (45% of CPU usage) than channels with data. This is because the GSM receiver correlates the input stream with 8 training sequences (instead of 1 training sequence if the channel has data).

For the second experimental setup using a signal source and Amazon VMs, we validated our integration system using the optimized AVX2-FMA channelizer only. Amazon client *can process up to 76 channels using the AVX2 channelizer*, which is lower than what a USRP N210 device can support, i.e., 92 channels. We present in Figure 4.36 the results from our evaluation scenario. The Amazon workers were able to analyze the digital signal from up to 11 channels (dynamically selected for each instance). We can see that the processed channels are growing linearly when we add workers until we reach 76 channels (the maximum number of channels that we can handle using one Amazon client). Concerning this scenario, the GSM

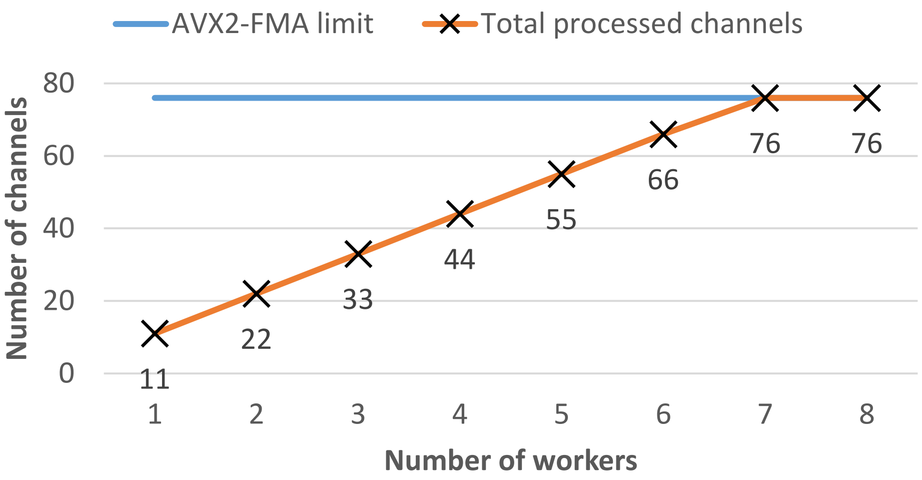


Figure 4.36: TCP blocks throughput chart.

receiver modules as discussed in Section 4.2.2 require more processing power to analyze random data than to analyze real data.

4.4.7 Compression Performance

At this point, we did not develop our own compressing algorithm but rather used some off-the-shelf implementations. Out ultimate target is to show the feasibility of more efficient distribution by improving the link efficiency. We used different generic compression algorithms with our TCP source/sink, namely, the Linux implementations of gzip, bzip2, lzma, and lz4 compressing algorithms. We performed end-to-end (over the GbE cable) experiments five times for each algorithm on an Intel Core i7-4790 @ 3.60 GHz CPU. Results are expressed in Table 4.2. We used the lowest compression level setting (= 1) for testing, which trades compression ratio for better speed. We can observe that the compression ratio is directly proportional to the compression time. The more the algorithm can compress data, the more time for compression it needs. It is clear that only the lz4 algorithm is fast enough for real-time data compression on $>$ 1 Gbps data transfers. lz4 can compress up to 1244.8 Mbps, which is higher than what the USRP N210 device can produce. With the 78.2 compression ratio for the lz4 algorithm, we can increase

Table 4.2: End-to-end compression measurements.

Algorithm name	Compression ratio	Compression speed-Mbps	Decompression speed-Mbps
gzip	48.4 %	313.6	629.6
bzip2	36.6 %	77.44	199.2
Lzma	44.4 %	45.92	234.4
lz4	78.2 %	1244.8	4213.6

the link capacity from 57 GSM channelized streams (without compression) up to 73 GSM channelized streams using the lz4 algorithm.

4.4.8 Conclusions

We presented in this section the challenges facing a wideband receiver using USRP N210 devices. Then, given the specification of the used ADC resolution, we calculated the maximum difference between highest and lowest received power in the wideband spectrum that will allow a success decoding for the lowest-power signal. These challenges produced by the uncontrolled environment can be avoided by operating at close distance to the serving BTS or an active mode, such as in the OpenBTS project [141]. However, the required processing power for the GSM receiver is high and the channelized streams cannot be processed on a single machine. While some basic building blocks to create a distributed system are already available, they are barely suited to create dynamic and reliable high-throughput distributed applications. We introduce a solution including a load-balancer based on the ZMQ library to create distributed GR applications. In this section, we demonstrated the usage of a GSM system. Our proposed solutions are in the form of independent GR modules [38]. Finally, to improve the efficiency of the link capacity (GbE connection), we propose the use of compression on both sides of this link, the channelizer side and the GSM receiver side. Real experimentation using off-the-shelf compression algorithms shows an improvement in link efficiency from 57 GSM channelized streams to 73 GSM channelized streams.

4.5 Localization Algorithms

In Sections 4.2, 4.3 and 4.4, we introduced the mechanisms needed to overhear GSM signals, capture and process them for the purpose of extracting information (MD identity and RSSI levels) for localization. In this section, we proceed towards the design and evaluation of proximity localization algorithms. We aim to deliver a solution that cope with passive system requirements and show high localization accuracy without any prior knowledge of the deployment environment. Note that our proposed localization algorithms can be performed agnostic to the data acquisition part, passive or active, GSM or WiFi.

4.5.1 Proposed Solutions

For a passive GSM-based system, we opted for the use of proximity-based localization due to its ease of deployment compared to other techniques. The next section presents the improvements we propose to proximity-based localization taking into account the invisibility requirement. We propose two novel proximity-based algorithms. The algorithms use RSSI information to adapt the weights for the ANs in the set C_{AN}. We consider two types of geometric methods: (1) the use of centroids

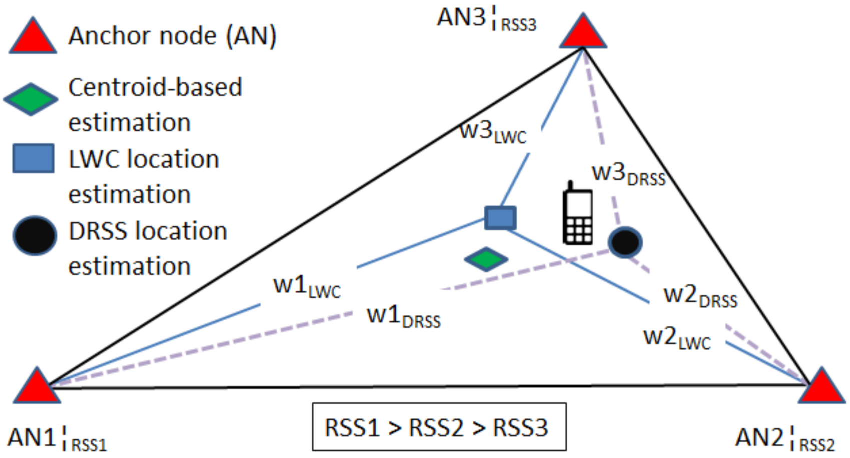


Figure 4.37: Operation of differential RSSI and LWC algorithms.

and (2) the use of circumcenters as the target position. We take LWC presented in Section 2.9.4 as a benchmark.

Combined Differential RSSI (CDRSS)

We developed several improved versions of the LWC method and compared their performance with each other. We present the most promising ones in terms of localization errors. CDRSS builds upon a differential RSSI (DRSS) approach, in which the calculation of the weights does not rely on the absolute RSSI values registered by each AN but on the difference in RSSI between the ANs [116]. Working with absolute RSSI has the drawback that different MDs may have different P_{tx}, making the organization of the RSSI levels in ranges and their mapping to weights challenging. Considering the lognormal shadowing model, and all ANs with equal gain settings, DRSSI omits the constant A from Equation 2.4 as follows:

$$\text{DRSSI}_{i,j} = P_{rx}(d_i) - P_{rx}(d_j) = 10\alpha\text{log}\left(\frac{d_j}{d_i}\right) - \psi'_{i,j} \qquad (4.4)$$

where $\psi'_{i,j}$ is the difference between the ψ_i and ψ_j shadowing effects.

The DRSSI method exploits the fact that RSSI decreases non-linearly with distance. Given the same inter-AN distance, ANs close to the target will have larger DRSSI than ANs that are more distant. Hence, nodes with larger DRSSI are given greater weight in the calculation of the target's coordinates. Figure 4.37 illustrates the conceptual presentation of the DRSSI and LWC algorithms. The weights of each anchor node are also shown. The DRSSI procedure is as follows:

1. Select the three ANs with the largest RSSI values.

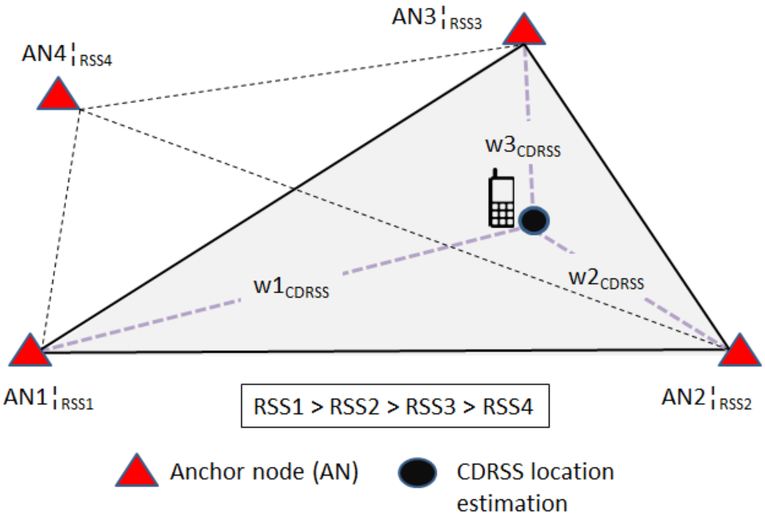


Figure 4.38: Operation of combined differential RSSI algorithm.

2. Calculate the relative DRSSI values between each two ANs using the following equations:

$$\left.\begin{aligned} X &= RSSI1 - RSSI2 \\ Y &= RSSI1 - RSSI3 \\ Z &= RSSI2 - RSSI3 \end{aligned}\right\} \Longrightarrow \left\{\begin{aligned} Q_1 &= Y/X \\ Q_2 &= Z/X \\ Q_3 &= Z/Y \end{aligned}\right. \tag{4.5}$$

   Note that the RSSI values are ordered using decreasing values so that X, Y, and Z are always positive and $Q_3 < 1$. Since a MD is separated by different numbers of walls and obstacles to each AN, relative DRSSI values are important to balance the weight calculations.

3. Calculate the weights for each contributing node by solving the set of equations:

$$\begin{gathered} w_{1_{\text{DRSSI}}} : w_{2_{\text{DRSSI}}} : w_{3_{\text{DRSSI}}} = Q_1 : Q_2 : Q_3 \\ w_{1_{\text{DRSSI}}} + w_{2_{\text{DRSSI}}} + w_{3_{\text{DRSSI}}} = 1 \end{gathered} \tag{4.6}$$

4. Estimate the MD coordinates by substituting each AN's weight and coordinates in Equation 2.3.

The combined DRSSI is further developed by modifying the DRSSI weights calculation with feedback on the RSSI level of all ANs and not only the three strongest ones. Figure 4.38 shows the CDRSS algorithm using four ANs. The following steps describe the CDRSS algorithm:

1. Form all possible K triangles of the C_{AN} set.

2. Calculate the weights $w_{ik_{\text{DRSS}}}|_{i=1,2,3/k=1:\text{K}}$ for all K triangles ANs according to Equation 4.6.

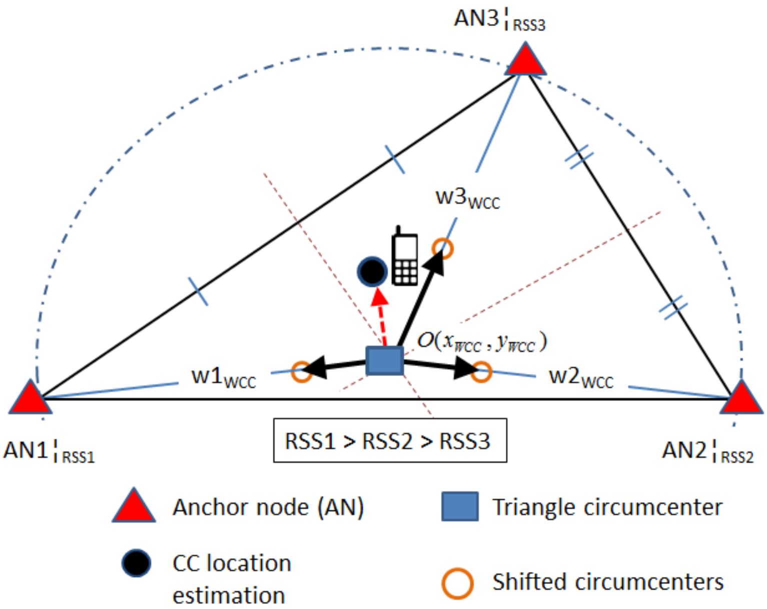


Figure 4.39: Operation of circumcenter algorithm.

3. Calculate the weights for the three ANs with the highest RSSI values as illustrated in Equation 4.7.

4. Substitute Equation 2.3 with weights and coordinates of the three highest ANs.

$$w_{i\text{CDRSS}} = \frac{1}{\text{K}}\sum_{k=1}^{\text{K}} w_{ik\text{DRSSI}} \quad \text{for } i = 1,2,3 \tag{4.7}$$

Weighted Circumcenter

The circumcenter is the center of the circumscribed circle around the polygon formed by the ANs such it is equidistant to all ANs. The concept is illustrated in Figure 4.39. Using the geometric circumcenter has the same disadvantage as using the centroid, i.e., the same estimated location is taken for multiple target areas as long as the same set of anchor nodes is used. Therefore, we introduce the WCC concept, which allows us to shift the triangle circumcenter closer to the target's actual position by integrating information on the registered RSSI levels. This concept is implemented using the following steps:

1. Form a triangle from three ANs with the largest RSSI values.

2. Calculate the triangle circumcenter $O(x_{wcc}, y_{wcc})$. The circumcenter is defined as the intersection of any two of the three perpendicular bisectors (a perpendicular bisector is a line that forms a right angle with one of the triangle's sides and intersects that side at its midpoint) as shown in Figure 4.39.

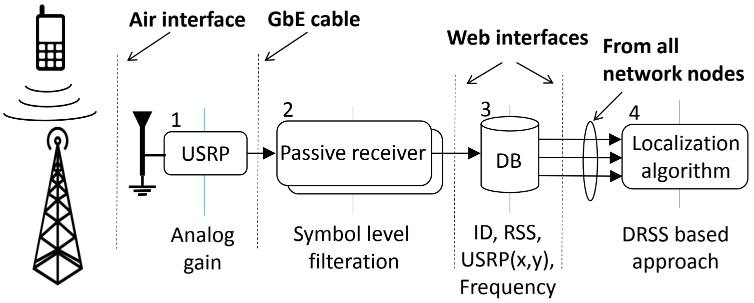


Figure 4.40: GSM SDR system.

3. Calculate the relative DRSSI values between each AN and its two neighbors using the following equations:

$$\left.\begin{aligned} X &= RSSI1 - RSSI2 \\ Y &= RSSI1 - RSSI3 \\ Z &= RSSI2 - RSSI3 \end{aligned}\right\} \implies \left\{\begin{aligned} h_1 &= X/Y \\ h_2 &= X/Z \\ h_3 &= Y/Z \end{aligned}\right. \tag{4.8}$$

   Relative DRSSI values are important to balance the weights calculation from different walls and obstacles distribution between ANs and the MD.

4. Move the circumcenter point towards each AN using the line equation between any two points in 2D space:

$$(x'_i, y'_i) = h_i * O(x_{wcc}, y_{wcc}) + (1 - h_i) * (x_i, y_i) \tag{4.9}$$

   For i = {1, 2, 3}, the result consists of three new potential positions of the circumcenter, i.e., (x'_i, y'_i). These three new points are used to form a triangle and to calculate its centroid.

5. Calculate the AN weights $w_{i_{WCC}}|_{i=1,2,3}$ for the newly formed triangle using Equation 4.6.

6. Estimate the MD coordinates by correcting the centroid coordinates using the following equation:

$$(x_{\text{est}}, y_{\text{est}}) = \left(\frac{\sum_{i=1}^{3} w_{i_{WCC}} * (x'_i, y'_i)}{\sum_{i=1}^{3} w_{i_{WCC}}} \right) \tag{4.10}$$

As illustrated in Figure 4.40, the gathered information, RSSI and MD identity, is stored in a remote database. For each captured message, the tool exports one MD identity and four RSSI values into the database. Collecting signal measurements is the first step in the localization process, the second step is feeding the collected measurements to a localization algorithm. The implementation of the WLC, CDRSS and WCC algorithms described in Section 4.5.1 has been done in Matlab.

4.5.2 Experimental Setup

In this section, we describe the experimental scenario and our findings. The indoor measurement setup is shown in Figure 4.41 and spreads over a single floor of a multi-floor office building. Bold black lines represent concrete walls, windows and doors openings are also indicated. We deployed four GSM ANs, marked by blue circles, at fixed positions throughout the floor. A first step in our experiments is frequency synchronization. This step is done by scanning the downlink channels to create a mapping between the serving cells and the uplink allocated channels in the area of interest. The individual GSM sensors were tuned to capture the same set of uplink channels administered by the Swiss mobile network of Sunrise simultaneously. Listening to several channels at once allows localizing MDs connected to different BTSs. Red rectangles mark measurement locations. We performed experiments at 27 locations for 1 hour each and collected 500 RSSI values per hour, as illustrated in Figure 4.41.

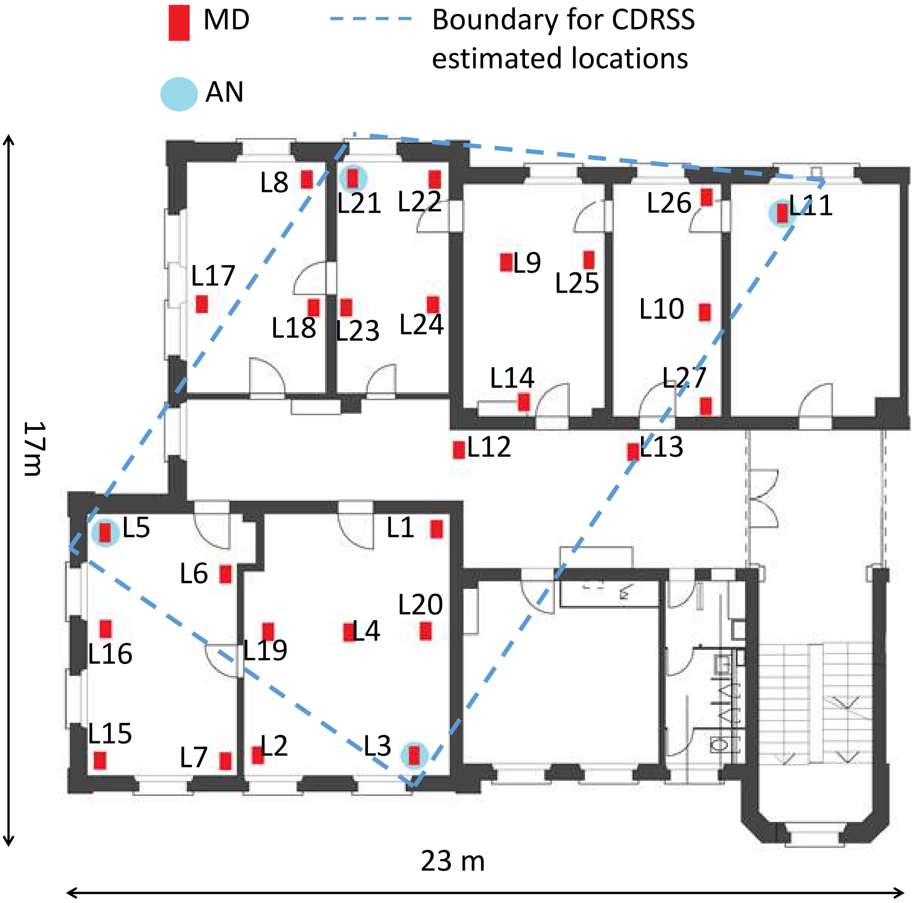


Figure 4.41: Indoor experiment environment setup.

4.5.3 Experimental Results

Mobile Device Identification

A mobile phone with GSM traffic, i.e., incoming or outgoing telephone calls, was used as localization target. For evaluation purposes, the target MD generates uplink traffic with high frequency using a self-developed Android application. In a first step, we identify the MD by its TMSI number. The TMSI is linked to the MD via RSSI measurements. We selected to work with the TMSI since it is used more often. 95% of all captured messages are using TMSI. In our observations, the TMSI in the Sunrise network is changed every 2 hours, which gives us enough time to run experiments.

Localization Accuracy

All experiments on localization accuracy were performed during working days when people were moving in the office, and doors were not closed all the time. Table 4.3 contains the localization error mean μ, error deviation σ and RMS for all conducted experiments using LWC, CDRSS and WCC[2]. Note that even points outside the triangle formed by the strongest three ANs (dashed blue line in Figure 4.41) will receive a location estimate but will result in relatively high localization error, e.g., L2, L7, and L8. The duration of each experiment is one hour. However, μ and σ are obtained over a period segment of 5 min. Based on the results in Table 4.3 we can observe that:

- The LWC method has the worst performance in all scenarios but three (L14, L23, and L24). In such cases, the good performance can be explained by the fact that the target's location was, in fact, the centroid of the LWC method.
- The performance of the WCC algorithm depends on the layout of the experiment. In most cases, WCC performs equally well or better than centroid techniques.
- For locations outside triangles (dashed blue line), the WCC algorithm shows a better performance compared to the CDRSS algorithm. This performance is because according to Equation 4.9, the node with the weakest RSSI pushes the estimated point away from itself as h_3 is always larger than one. In contrast, in case of CDRSS each node pulls the centroid towards itself.

We developed a Graphical User Interface (GUI) using Matlab to demonstrate localization performance. The GUI indicates the MD's estimated location by small dots, using one of the localization algorithms as described in Section 4.5.1. Figure 4.42 shows GUI outputs of two experiments at locations L4 and L9 using the WCC algorithm. A more detailed description of Figure 4.42 is given in Section 4.5.3.

[2] Note that these algorithms do not require any channel modelling or PLE estimation.

Impact of Indoor Environments

For all experiments illustrated in Figure 4.41, MDs were stationary over the period of each experiment. However, at some locations the MD has at least one LOS link to the GSM sensor, while at other locations it has only NLOS links since an obstruction is in the path. Doors offer an excellent opportunity to investigate the effects of obstacles, by simply opening/closing them. Opening a door might create stronger multipath components or a LOScommunication link. Figure 4.43 shows the RSSI measurements of the MD for LOS and NLOS conditions. The MD was static over 24 hours. The RSSI values were aggregated over periods of 5 minutes.

Table 4.3: Error comparison of Centroid and Circumcenter based approaches (meters).

MD Location	LWC			CDRSS			WCC		
	μ	σ	RMS	μ	σ	RMS	μ	σ	RMS
L1	3.36	0.28	3.37	2.52	0.93	2.68	2.72	1.18	2.96
L2	6.45	0.09	6.45	4.47	0.20	4.47	3.19	0.02	3.19
L3	6.68	0.21	6.68	4.21	0.91	4.30	4.38	0.87	4.46
L4	3.76	0.08	3.76	2.68	1.49	3.06	2.14	1.46	2.59
L5	5.17	0.12	5.17	3.67	0.19	3.67	1.83	0.21	1.84
L6	1.94	0.76	2.08	0.74	0.17	0.75	1.02	0.48	1.12
L7	6.52	0.19	6.52	4.88	0.77	4.94	3.54	1.29	3.76
L8	8.42	0.20	8.42	4.50	0.19	4.50	3.67	1.18	3.85
L9	4.52	0.10	4.52	1.15	0.11	1.15	0.97	0.13	0.97
L10	5.91	0.11	5.91	3.67	0.20	3.67	2.01	0.51	2.07
L11	6.37	0.27	6.37	4.85	0.11	4.85	3.10	0.40	3.12
L12	2.25	1.15	2.52	2.17	1.54	2.66	2.00	2.65	3.32
L13	4.58	0.83	4.65	2.50	0.41	2.53	1.25	1.01	1.60
L14	0.57	0.06	0.57	2.95	1.01	3.11	2.32	1.89	2.99
L15	8.36	0.15	8.36	4.52	0.83	4.59	3.42	1.68	3.81
L16	7.77	0.22	7.77	4.71	0.30	4.71	3.26	0.41	3.28
L17	7.41	0.14	7.41	2.30	0.92	2.47	2.36	0.74	2.47
L18	5.44	0.20	5.44	2.86	0.92	3.00	2.46	1.76	3.02
L19	3.45	0.12	3.45	2.14	0.59	2.21	3.54	0.62	3.59
L20	2.94	0.57	2.99	3.10	1.13	3.29	2.92	1.51	3.28
L21	5.85	0.15	5.85	2.54	1.01	2.73	2.74	1.30	3.03
L22	5.41	0.23	5.41	2.10	0.49	2.15	3.27	0.88	3.38
L23	1.12	0.31	1.16	3.32	0.81	3.41	2.99	1.03	3.16
L24	1.41	0.24	1.43	3.52	0.20	3.52	2.01	0.41	2.05
L25	3.73	0.17	3.73	2.85	0.33	2.86	2.70	0.40	2.72
L26	8.45	0.15	8.45	4.17	1.11	4.31	4.50	1.35	4.69
L27	7.52	0.83	7.56	4.14	0.51	4.17	4.87	0.61	4.90
Average	5.01	0.31	5.02	3.25	0.64	3.31	2.78	0.97	2.94

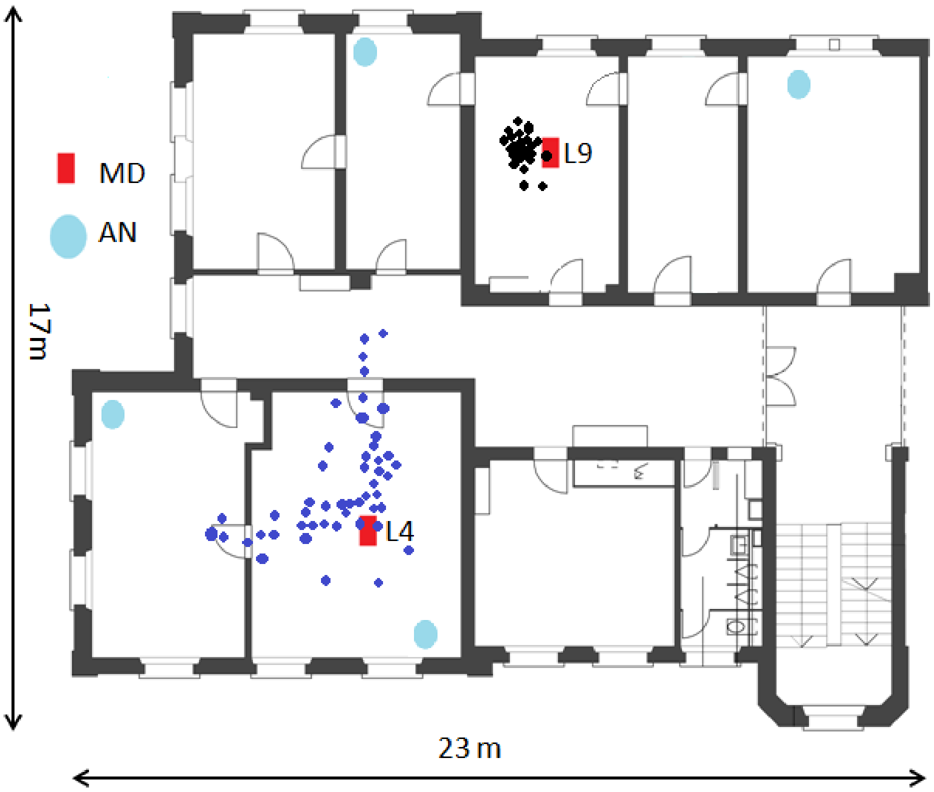


Figure 4.42: Estimated locations using WCC algorithm: L9 in case of closed doors and L4 in case of open doors.

From Figure 4.43, we draw the following conclusions:

- A NLOS environment degrades the RSSI value more than LOS. According to [60], we expect the amount of RSSI losses to be related with the type and number of obstructions, e.g., concrete walls have higher loss than wooden doors.
- Assuming a static indoor environment overnight, NLOS causes stronger RSSI fluctuation compared to LOS.
- Since multipath propagation is often created by moving objects [12], moving people during the daytime in addition to opening and closing of doors create a severe RSSI variation - up to 10 dB in NLOS conditions.

The effect of the environment is visible on Figure 4.42. Experiments with open doors, such as at location L4, have higher estimated position deviations than experiments with closed doors, such as at location L9. This performance is because opening a door causes additional multipath propagation components to appear and can lead to LOS to a previously ’hidden’ AN, causing the AN to pull the centroid stronger towards itself.

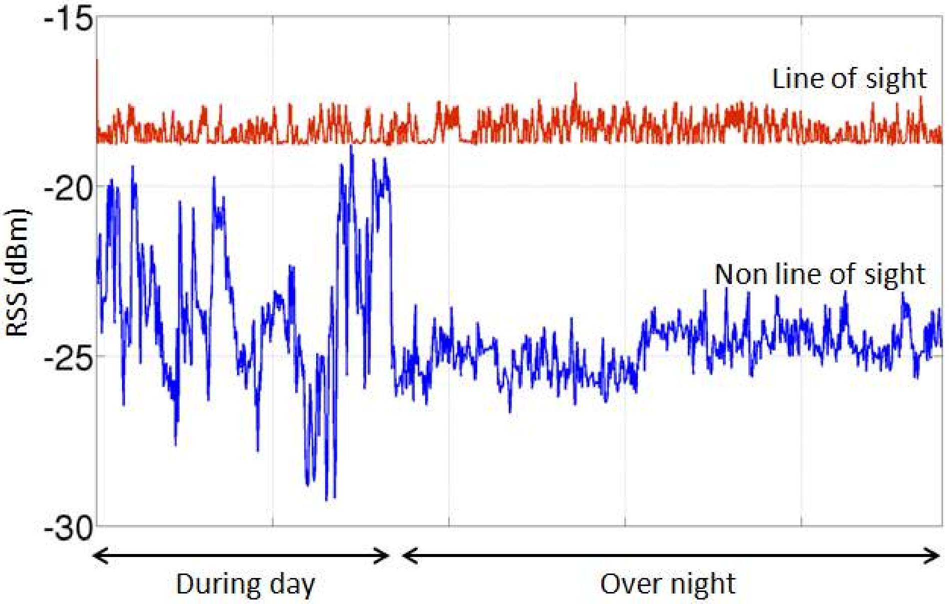


Figure 4.43: LOS and NLOS power relation during day and night.

4.6 Robust Radio Estimators

As it was clearly shown in Section 4.5.3 (Impact of Indoor Environments), the indoor environment generates contaminated signals, i.e., outliers, which behave in unpredictable ways and subsequently introduce localization errors. In this section, we propose a filtering approach against outliers without any prior knowledge of the observed signal and the MD radio settings.

4.6.1 Proposed Algorithm for Outlier Detection and Filtering

Outlier filtering techniques have been considered in different studies to protect localization parameters from the influence of outliers [41]. Most of these techniques fall into one of the two categories: (i) filtering outliers of the parameters fed into the localization algorithm, and (ii) filtering outliers in the localization output. We focus on filtering outliers of the parameters fed into the localization algorithm. Our proposed algorithm, as illustrated in Figure 4.44 should be implemented at the front of the localization system [17]. The input of the algorithm is the received signal strength of a MD measured by one AN. After all the signal measurements have been gathered, the algorithm applies the following post-processing steps to obtain a robust RSSI data set against outliers:

1. Create a set of RSSI measurements X_n over a period T fixed among the ANs.
2. Calculate the hat matrix using X_n and acquire each RSSI$_i$ leverage value, i.e., h_{ii}. Then, estimate the $E_n(X_n)$ parameter using the Huber function

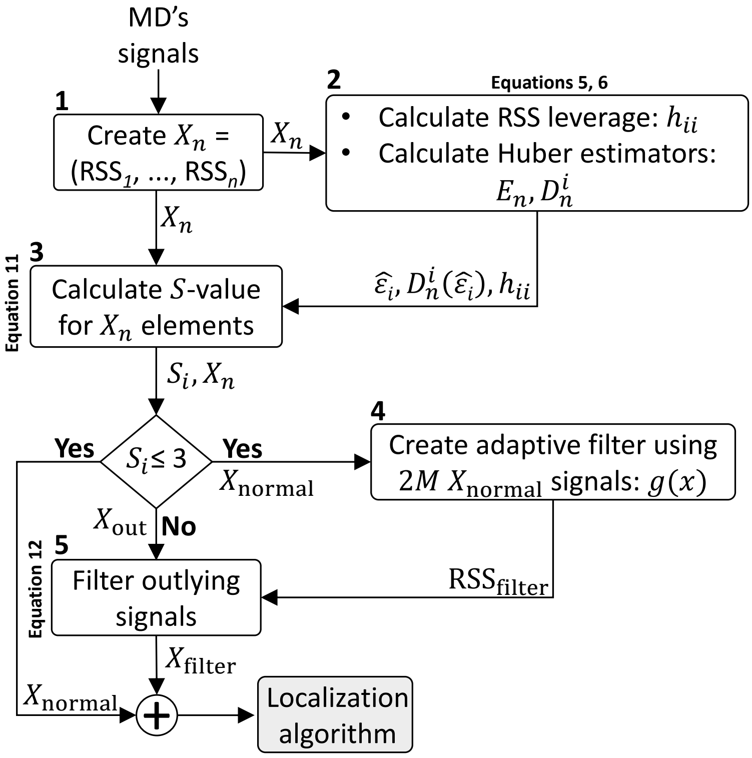

MD's signals
1
Create X_n = (RSS$_1$, ..., RSS$_n$)
X_n
2
Equations 5, 6
• Calculate RSS leverage: h_{ii}
• Calculate Huber estimators: E_n, D_n^i
X_n
3
Equation 11
Calculate S-value for X_n elements
$\hat{\varepsilon}_i, D_n^i(\hat{\varepsilon}_i), h_{ii}$
S_i, X_n
$S_i \leq 3$
Yes
Yes
X_{normal}
4
Create adaptive filter using $2M$ X_{normal} signals: $g(x)$
X_{out}
No
5
Equation 12
Filter outlying signals
RSS$_{\text{filter}}$
X_{filter}
X_{normal}
Localization algorithm

Figure 4.44: Outlier detection and filtering algorithm.

with the M-estimator expressed in equation 2.25. The resulting value is used to calculate the residual value $\widehat{\epsilon_i}$ for each RSSI$_i$ in X_n. The $D_n^i(\widehat{\epsilon})$ parameter is then calculated using the Huber function with the M-estimator expressed in equation 2.26. Huber argued [98] that b = 1.345 is a good choice that provides robustness against outliers and high performance against false detection. We also use b = 1.345.

3. Calculate the S-value for each RSSI$_i$ in X_n as expressed in equation 2.31. RSSI measurements that have a S-value outside the ϖ=3 range are considered as outliers, denoted as X_{out}. The difference between the X_n and X_{out} sets are the considered-normal signals, denoted as $X_{\text{normal}} = X_n \setminus X_{\text{out}}$.

4. The simplest outlier mitigation technique is achieved by discarding outliers from the original set of measurements. However, removing all detected outliers outputs a lower number of localization points, which is a limiting factor in most post-processing algorithms [190]. The following steps summarize our proposed dynamic filtering algorithm using a probabilistic approach to online measurements:

   - Select a number 2M of X_{normal} measurements around the detected outlier, denoted as X_{2M}.

- Create a Probability Mass Function (PMF) of the RSSI with probabilities $g(\text{RSSI}_i) = \Pr(x = \text{RSSI}_i)\ \forall\ x \in X_{2M}$ using signals from the previous step.
- Select the RSSI value that has the maximum probability inside the created PMF, denoted as $\text{RSSI}_{\text{filter}}$.

$$\text{RSSI}_{\text{filter}} = \underset{x \in X_{\text{normal}}}{\arg\max}\, g(x) \tag{4.11}$$

5. Replace the detected outliers by $\text{RSSI}_{\text{filter}}$.

4.6.2 Experimental Setup

In this section, we describe the experimental scenario of the indoor setup and AN deployment. To validate our work, we make use of the same wideband GSM receiver described in Section 4.5. As shown in Figure 4.45, the robust estimator module, built in Matlab, fetches RSSI records from the DB for a certain MD and applies classical and robust estimation of RSSI values over a particular aggregation period. This process is used with signal streams from all ANs independently and in parallel. To validate our proposed solution, the WCC localization algorithm described in Section 4.5.1 is used. Note that Figure 4.45 contains all modules in Figure 4.40 with the addition of the robust RSSI estimator (step 4).

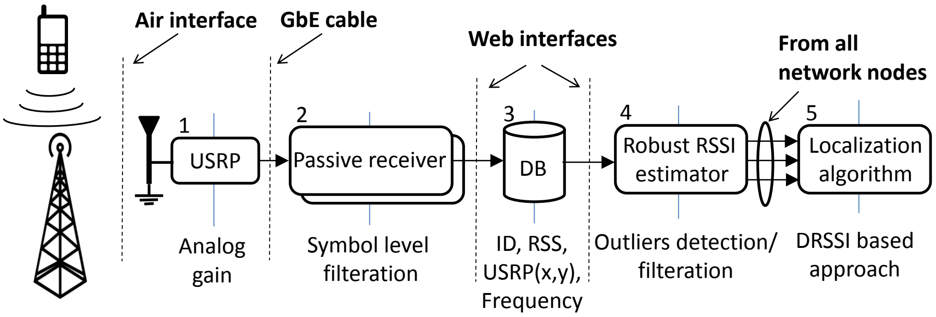


Figure 4.45: GSM sensor processing chain.

We deployed four GSM ANs, i.e., four USRPs, at fixed positions on a single floor of an office building as shown later in Figure 4.49. ANs have just some knowledge about the operating frequency range of MD. A first step in our experiments was frequency synchronization. This step was done by scanning the downlink channels to create a mapping between the serving cells and the uplink channels in the area of interest. The individual GSM sensors were then tuned to capture the same set of uplink channels administered by the Swiss mobile network operator Sunrise simultaneously. A mobile phone with signaling traffic, i.e., incoming or outgoing telephone calls, was used as a target for localization. For evaluation purposes, the target MD generates uplink traffic with high call frequency using a self-developed Android application.

4.6.3 Experimental Results and Discussion

In this section we present the results of experiments that show the power variation in the captured uplink GSM messages, the impact of power variation on localization accuracy, the improvements achieved using our proposed algorithm and further analysis in the detected outliers.

RSSI Location and Dispersion Estimators

A MD generated uplink traffic at a distance of 1m from an AN; the communication took place in a NLOS indoor environment, i.e. a metal cupboard was used as an obstruction. Figure 4.46 gives a zoom-in view for the RSSI measurements of one GSM burst inside the *gsm-receiver* module. The RSSI value of each symbol represents the combined signals from multiple paths at the receiver as illustrated in equation 2.19. The GSM burst contains slow and fast fading components around -49 dBm. The slow fading symbols in Figure 4.46, i.e., the considered-normal symbols in equation 2.20, were obtained using an averaging window size of 10 applied on the burst symbols. In our experiments, we observed that the RSSI of the symbols over one burst duration, 570 μsec, have a normal distribution. Hence, a classical estimator such as the mean can achieve results comparable to a robust estimator.

However, when combining multiple bursts of the *same MD and the same experiment setup*, as illustrated in Figure 4.47, we observed a RSSI variation between -48 dBm and -60 dBm. Every point in Figure 4.47 represents the RSSI value of a GSM symbol. Note that we have 156 symbol/GSM burst and hence, Figure 4.47 shows RSSI measurements of 51 GSM bursts. A large number of symbols has RSSI values around -49 dBm with considerable measurement dispersion towards the lower values. On the one hand, a classical estimator, such as the MLE estimator, is biased by outlying symbols below -49 dBm, i.e., the nondeterministic symbols in equation 2.20. The MLE amplitude estimation is around -51 dBm. On the other hand, a robust estimator, such as the Huber estimator, shows higher robustness to outlying

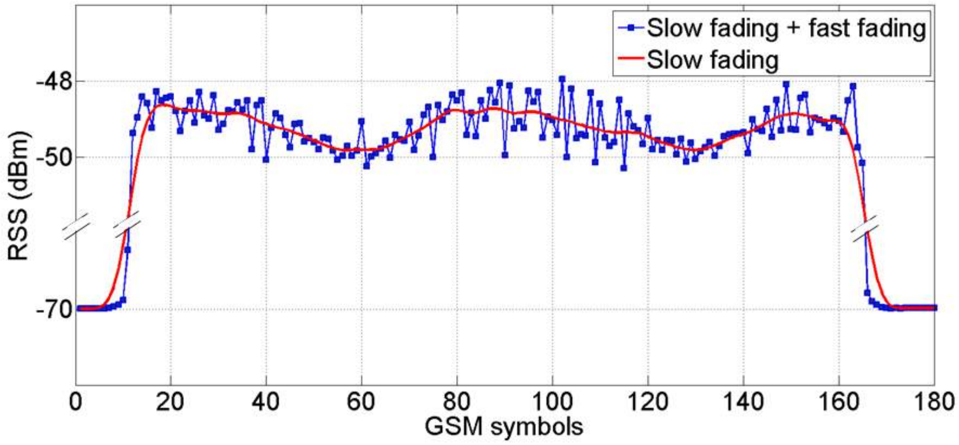


Figure 4.46: GSM burst power.

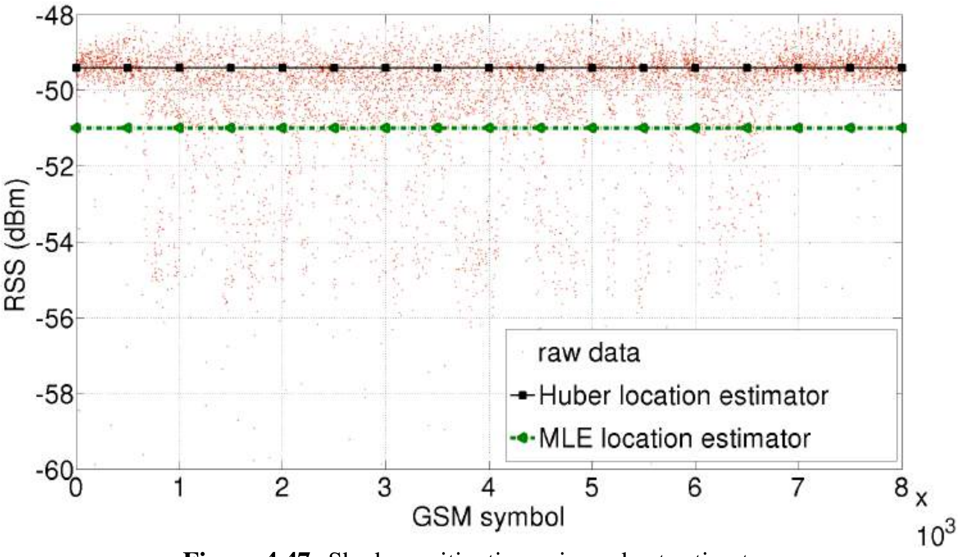


Figure 4.47: Shadow mitigation using robust estimator.

symbols. Its amplitude estimation is -49 dBm. To obtain robustness against outliers, we applied our proposed RSSI filter shown in Figure 4.44. Given the RSSI measurements in Figure 4.47 as the input parameter to the proposed algorithm, the measurement set X_n were created over periods of five minutes. The response time requirement can be reduced to several seconds in case of active localization as in LTE networks. However, our focus is not on the radio technology but to validate the filtering algorithm in general. The $E_n(X_n)$, $\widehat{\epsilon_i}$, and $D_n^i(\widehat{\epsilon})$ parameters (step 2 in Figure 4.44) were obtained over RSSI values using both the MLE and the Huber estimators. Using the proposed dynamic filtering approach with $M = 20$. Figure 4.48 shows the RSSI Probability Density Function (PDF) distribution of the algorithm's input raw data and output measurements. The processed measurements using the Huber estimator have five times smaller deviations than the original one.

The Huber estimator also shows better performance in detecting outliers than the MLE estimator. This result implies that using the Huber estimator in our proposed filtering algorithm enables us to obtain a robust RSSI distribution. This result is important for both robust channel modeling as well as giving a robust indication on the communication link quality, e.g., by measuring the number of outliers within a given period. Moreover, the proposed filtering algorithm achieved comparable performance compared to removing outlier while keeping the localization set size.

Localization Performance

In this section, we present the localization performance results obtained using our proposed system for detecting and filtering outliers. Given the experiment setup illustrated in Figure 4.49, four USRPs, marked by blue circles, overheard a static MD uplink traffic. Red rectangles mark measurement locations of a MD. We performed experiments at 10 locations for one hour each and collected 500 RSSI values per hour. Note that these location points are different that the ones presented in Section 4.5.2 and 2 years almost in between. Assuming that RSSI measurements are uncorrelated at different USRPs, each USRP performs independently the proposed filtering algorithm using the Huber and MLE estimators (step 4 in Figure 4.45). The following steps are applied to the localization algorithm to derive the target user location (step 5 in Figure 4.45):

- Aggregate the input RSSI measurements over periods of five minutes.
- Average every period of RSSI measurements using simple mean calculation.
- Feed the aggregated measurements into the WCC algorithm.

All experiments on localization accuracy were performed during working days when people were moving in the office, and doors were not closed all the time. Selected points are different that ones presented in Table 4.3. Table 4.3 contains the localization error mean μ, error deviation σ and RMS for all conducted experiments. μ and σ are obtained from multiple location estimates performed at each location of a MD. Based on the results in Table 4.3 we can observe:

- Filtered RSSI measurement sets show a general improvement in the localization error mean and standard deviation than raw data sets. However, the MLE estimator yields worse localization performance than non-filtered data at L3 and L4. This result can be explained by the fact that these two points have the highest non-filtered error deviation and challenge the outlier detection by the MLE estimator.
- For all locations, the proposed algorithm using the Huber estimator achieves greater accuracy compared to the MLE estimator. The improvements of the filtering algorithm using the Huber estimator are (4.72–2.7)/2.7 x 100 = 75%

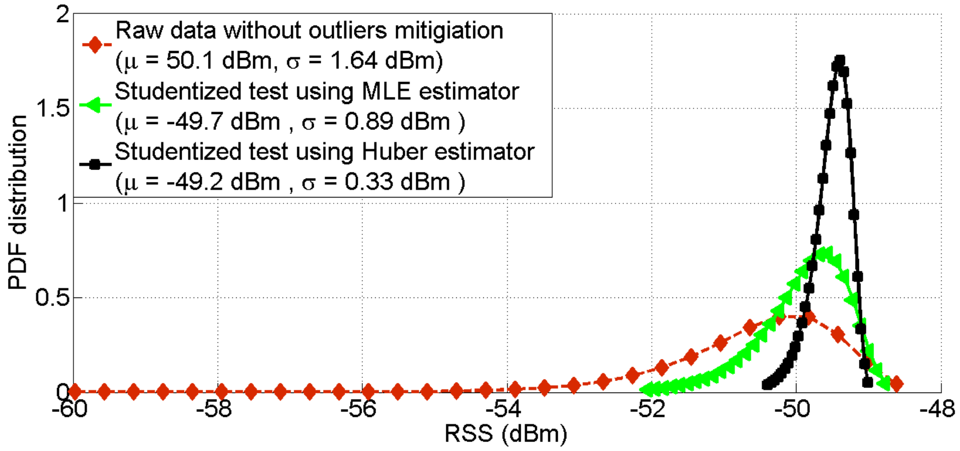


Figure 4.48: Shadow mitigation using robust estimator.

for the localization error mean, (2.61–0.5)/0.5 x 100 = 422% for the localization error deviation and (5.39–2.74)/2.74 x 100 = 96.7% compared to the raw data set.

However, we see a difference in the localization error mean between the WCC algorithm and the ones presented in Table 4.3. This is part of the natural characteristics of the RSSI in different environments, i.e., inconsistency of RSSI measurements in real-world indoor environments. In this set of experiments, we observed a higher variation in RSSI measurements, which results in a lower performance that the one presented in Table 4.3. We believe the reason is that these experimental locations are located in a high density of metal and furniture areas. The change in the environmental settings (and the time) makes it hard to compare. The importance from our perspective is the improvement that our proposed filtering solution adds to such locations.

The GUI illustrated in Figure 4.50 indicates the estimated location of the MD at location L2 using different aggregated points. We see that the proposed algorithm using the Huber estimator leads to estimate locations only inside the real target room while the other two cases show various estimated positions between different incorrect rooms.

Moreover, we studied the outlier detection performance with different numbers of measurements. We performed the experiment at L2 for a duration of eight hours. Figure 4.51 shows the error mean and standard deviation versus the number of measurements of a step size equal to 200. From Figure 4.51, we draw the following conclusions:

- Additional RSSI measurements improve the performance, but only with sets of filtered measurements. The random behavior (mean and STD) of non-filtered measurement sets is due to the randomness of RSSI measurements.

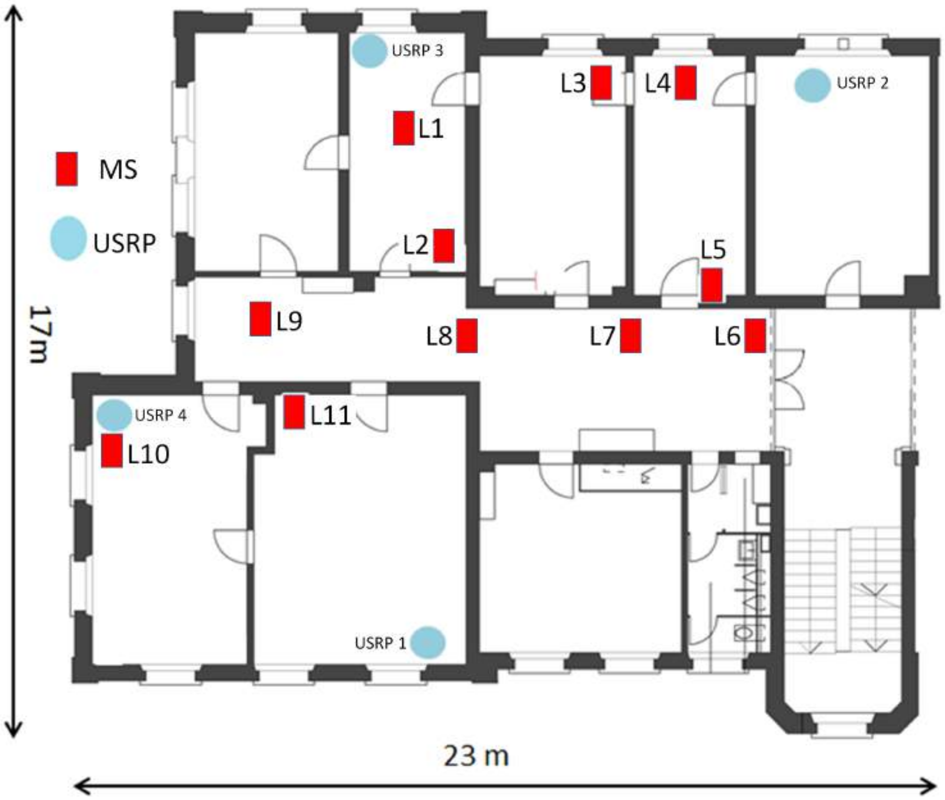


Figure 4.49: Experiment setup.

Table 4.4: Error comparison of WCC localization algorithm using raw and filtered based approaches (meters).

MD Location	No filter			MLE			Huber		
	μ	σ	RMS	μ	σ	RMS	μ	σ	RMS
L1	4.38	1.76	4.72	4.21	0.85	4.29	2.08	0.36	2.11
L2	2.17	1.58	2.68	1.81	0.36	1.84	1.70	0.01	1.70
L3	6.10	7.83	9.92	6.70	1.10	6.78	3.63	0.43	3.65
L4	5.97	4.39	7.41	6.42	2.39	6.85	2.77	0.68	2.85
L5	6.68	2.61	7.17	5.67	2.19	6.07	3.39	1.23	3.60
L6	6.56	3.75	7.55	7.07	4.64	8.45	3.84	0.16	3.84
L7	4.01	1.61	4.32	4.66	1.56	4.91	2.69	0.18	2.69
L8	3.16	2.62	4.10	3.05	1.12	3.24	1.83	0.90	2.03
L9	2.75	1.94	3.36	1.33	0.61	1.46	0.82	0.07	0.82
L10	5.42	2.06	5.79	4.28	1.55	4.55	4.28	1.05	4.40
Average	4.72	2.61	5.39	4.52	1.63	4.80	2.70	0.50	2.74

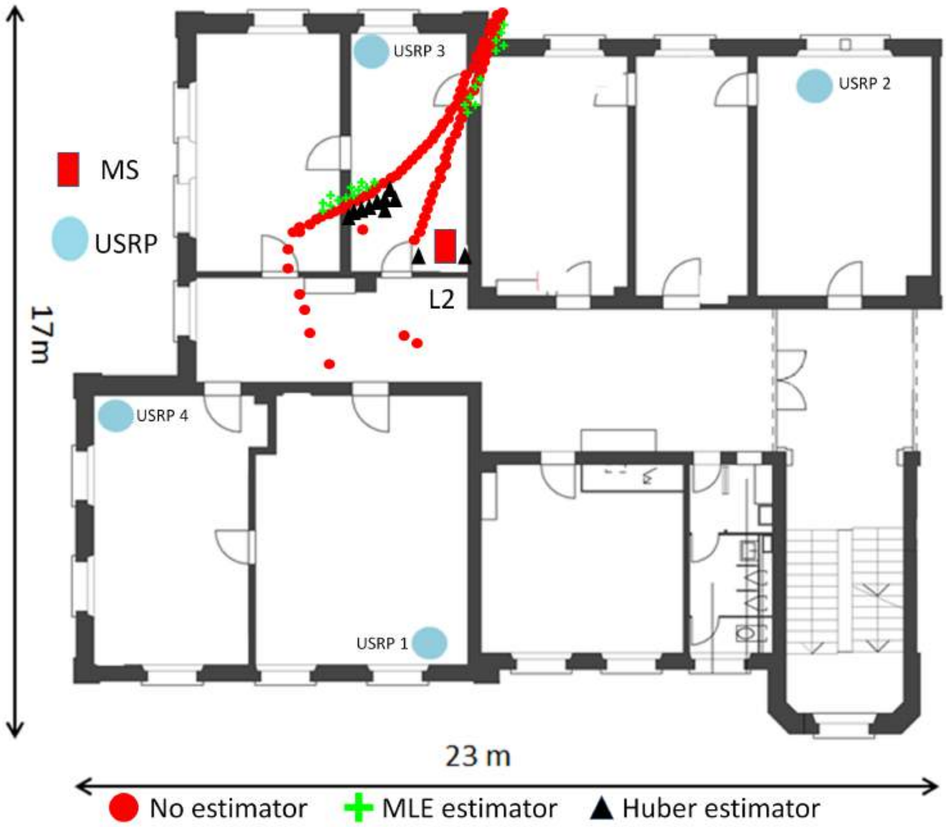


Figure 4.50: Estimated location using different aggregated points.

- The proposed algorithm using the Huber estimator gives the biggest improvement with a small number of RSSI measurements. Moreover, it shows lower localization errors than the MLE estimator with any number of input measurements.

- The proposed algorithm using the Huber estimator shows almost a constant localization error STD regardless of the size of the measurement set. Note that the smallest size of the measurement set equals to 200 bursts.

Residual Analysis

Most localization systems focus on modeling the considered-normal signals. However, if the system determines residuals $\widehat{\epsilon}$ using robust estimators, the residual distribution becomes valuable for localization (c.f. Section 2.11.3). Using the RSSI measurements at L2 as an example, we examined in Figure 4.52 quantile-quantile (Q-Q) plots of the residual using the Huber estimator. Q-Q plots are used to evaluate whether a dataset comes from a normal distribution. Note that a normal residual distribution occurs in LOS conditions. The R-squared goodness of fit (R^2)

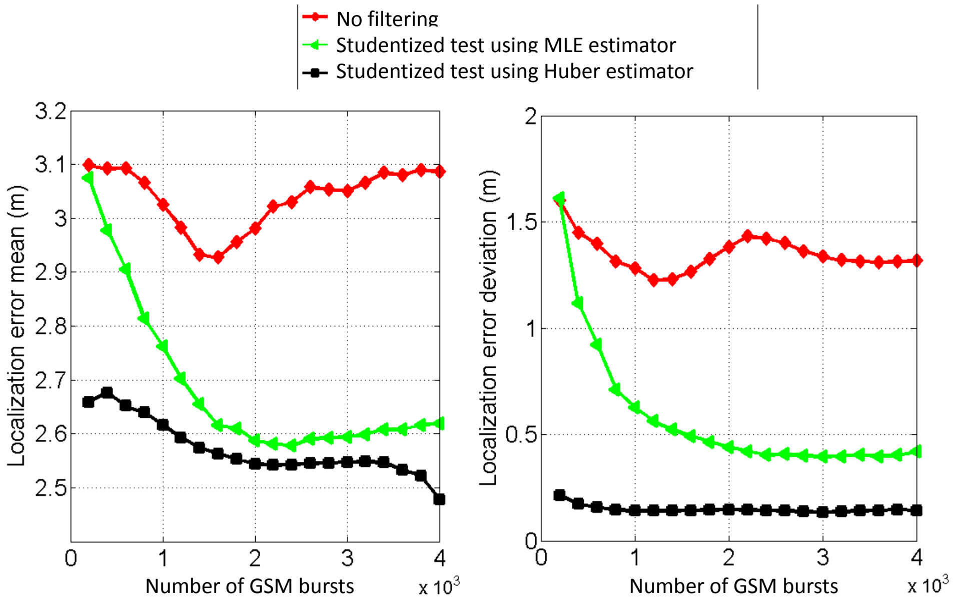


Figure 4.51: Localization error mean and standard deviation with and without outlier detection and filtering techniques.

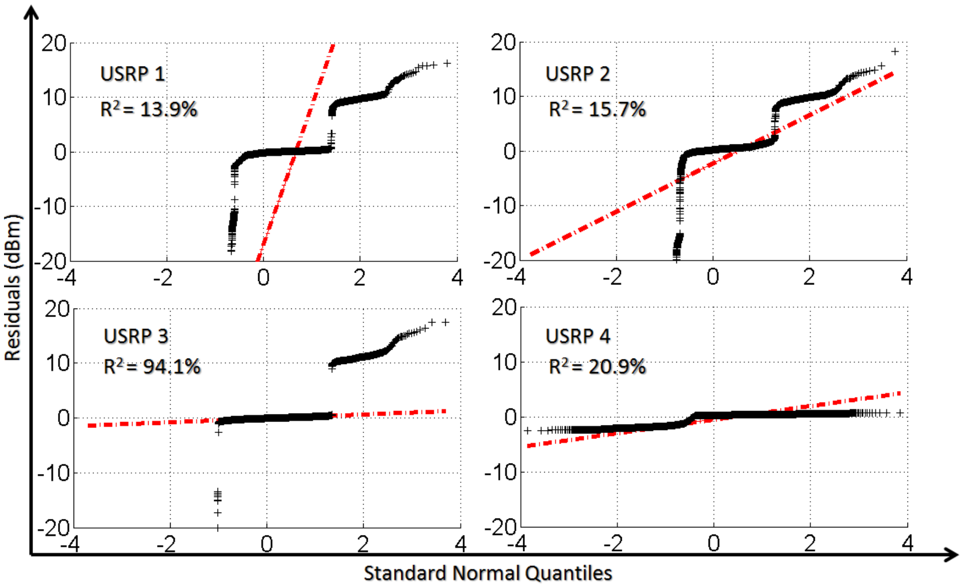


Figure 4.52: Signals residual of 4 GSM ANs using the Huber estimator

is a measure of how good the data is fitted to a curve [62]. In our work, R^2 is used to assess the robustness of the Huber estimator against outliers. We observe that the RSSI residuals at USRP$_3$ fit the normality test between -1 and +1 quantiles with 94.1% precision. The RSSI residuals at other USRPs did not meet the normality test since these USRPs have no LOS communication links with MD. The Q-Q test of residuals obtained from robust estimators can be used to give a physical inter-

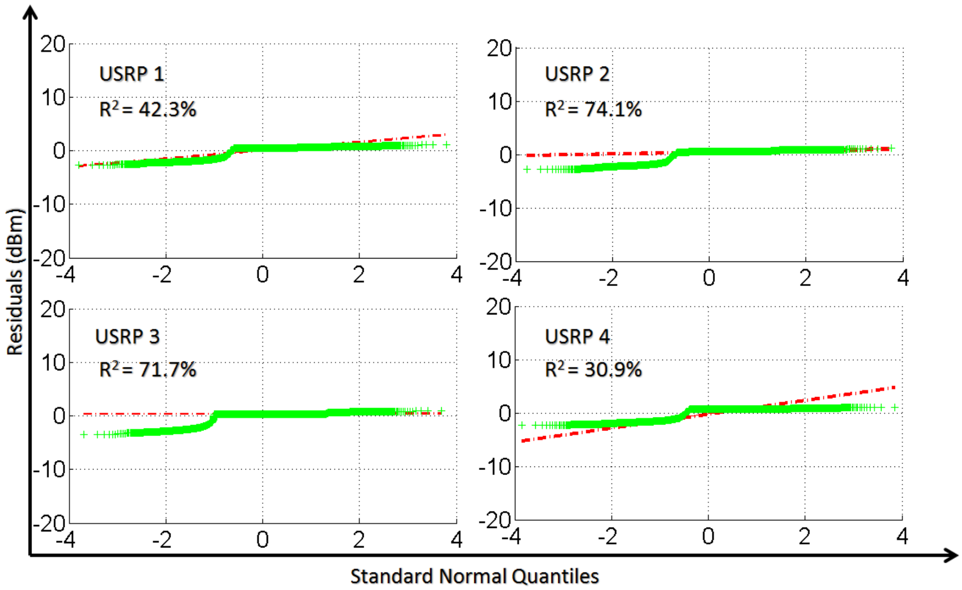


Figure 4.53: Signals residual of 4 GSM ANs using the MLE estimator

pretation of the obstructions, e.g., walls and floors, and communication conditions, e.g., LOS and NLOS. However, the Q-Q plot of residuals obtained from the MLE estimator in Figure 4.53 has similar R^2 values for USRP$_2$ and USRP$_3$. Decisions based on MLE residuals do not help in localization.

4.7 Conclusions

The single GSM receiver presented the problem of GSM uplink capturing by third-party devices from a theoretical and a realization perspective. We proposed a new time synchronization implementation that used the structure of uplink frames to achieve synchronization with the mobile device. The method can synchronize to multiple MDs simultaneously by only overhearing uplink traffic. The proposed passive receiver was implemented using SDR Platform. Its performance evaluation showed an average of 98.8% success rate of downlink message recovery and close to 70% of uplink messages. We also investigated the impact of received signal strength on the receiver's performance. The results indicate an RSSI threshold, below which successful message decoding is not possible. The limitation of low RSSI values was minimized using optimized receiver power gain values. Finally, we implemented an uplink message parser limited to unencrypted messages for later service development. Since the receiver operates independently from the network operator, it is attractive to third parties interested in GSM signal recovery, one of its applications being passive MD localization.

GR is the framework of choice for many SDR projects. We demonstrated that it was possible to create high-performance GSM channel processing by reusable components for wideband applications. The polyphase filterbank channelizer is not only useful for FDM-based technologies but is a critical component in many SDR systems. Our AVX2 optimizations improve the performance by up to 30 % using recent CPU instructions, namely fused multiply-add operations, compared to the AVX implementation. Moreover, the GPU-based implementation opens up opportunities for applications, where expensive specialized hardware was required previously. The GPU-based implementation is a GR-like module with 3.2-fold faster with respect to the PFB processing time when compared to the original GR implementation on CPU only.

Moreover, we focused on the development of localization algorithms for indoor spaces, which explore information on the deployment geometry and signal strength of radio (GSM) signals. In particular, our system is unique since it does not rely on the cooperation of the network or the MD but uses signal overhearing for localization. The system consists of several SDR sensors, which can capture GSM messages from the target object and process them for localization. Localization is based on CDRSS and WCC proximity-based localization algorithms. The first algorithm builds upon the centroid concept while the second algorithm is based on the circumcenter concept. Both algorithms explore information on received signal strength and particularly the difference in readings from sensors to derive a location estimate. The second algorithm allowed us to derive location estimates of targets outside the geometry of the deployment. To validate the effectiveness of our proposed algorithms, a set of measurements were conducted in a real indoor environment and an actual GSM MD as a target. The performance results showed that with zero network configurations, both algorithms outperformed the linear weighted centroid algorithm. Moreover, we observed the impact on localization accuracy by dynamic indoor environments.

To account for the complicated radio propagation and its consequences on localization accuracy in dense indoor environments, we proposed to extend outlier mitigation algorithms with a novel filtering algorithm. The approach aims to increase the quality of the radio parameters before being fed into the localization algorithm. This requirement can be achieved using robust radio parameter estimators such as the Huber estimator. The robustness of the implemented algorithms is shown to be effective in dealing with radio signals. A performance comparison using robust and classical radio parameter estimators is illustrated in the section. We validated our proposed algorithms using the RSSI parameter as it can easily be obtained from most off-the-shelf equipment. The experimental results indicate that the proposed algorithms can significantly decrease the absolute localization error mean and deviation using the minimum number of anchor nodes, i.e., three ANs for a 2-dimensional space. Moreover, we studied the characteristics of the outliers measured by the Huber- and the MLE- estimators. The results are promising and support the localization algorithms at the room level using the Huber estimator.

Chapter 5

Hybrid-Network Localization Systems

5.1 Introduction

Both localization systems presented in Chapter 3 (WiFi-based) and Chapter 4 (GSM-based) have their own limitations in indoor passive environments. On the one hand side, although the packet rate in WiFi is high, WiFi radio signals are more vulnerable to interference than GSM, because WiFi devices operate in an unlicensed frequency band. On the other hand, although GSM devices operate on a licensed band (less vulnerable to interference), a GSM passive localization system can identify target users using only three types of messages. To combine advantages from both systems and improve the overall localization performance, we propose in this chapter a hybrid-network indoor localization system using multiple Radio Interfaces (RIs). The proposed solution detects changes in signal quality and employs different weighting schemes to favor the signal with higher quality after filtering. The proposed solution relies on simultaneous active RIs of the target MD. A realistic indoor scenario from our daily life is the use of GSM RI for sending and receiving calls (not data), and the WiFi RI for sending and receiving data packets, such as email browsing and video streaming. For passive localization systems, we verify the proposed solution using a proximity-based localization algorithm, namely, the CDRSS method [18]. The proposed solution benefits from the varying uncorrelated characteristics of different radio signals and offers reliable and accurate localization performance in real indoor environments [21]. Our contributions to the current indoor localization state of the art include the following:

- A hybrid signal preprocessing for combining online radio signal information based on a probabilistic approach (c.f., Section 5.3.2).
- A hybrid location post-processing that combines online location output of radio signals based on a probabilistic approach (c.f. Section 5.3.3).
- A set of real indoor experiments and corresponding performance evaluations (c.f. Sections 5.4 and 5.5).

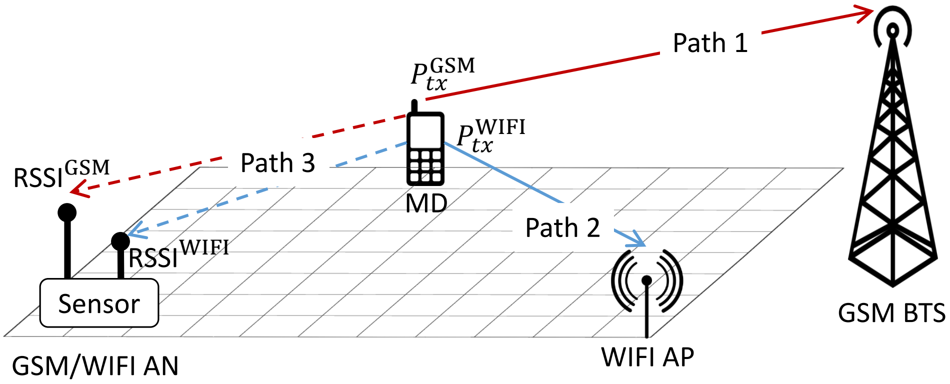


Figure 5.1: Signal overhearing setup

5.2 Characteristics of Captured Signals

To combine radio metrics from different RIs, an essential step is to understand the propagation characteristics of the combined metrics. As mentioned in Sections 2.2 and 4.2.1, the operating frequencies of both WiFi and GSM are different. Hence, a GSM/WiFi AN, as illustrated in Figure 5.1, will receive different RSSI values (abbreviated as $\mathcal{R}$) from a MD at a distance d, even if both WiFi and GSM RIs are transmitting the same amount of power P_{tx}. However, due to the power control algorithm in each radio technology, the transmitted power at each RI is not the same. As illustrated in Figure 5.1, the GSM BTS controls the MD to transmit with P_{tx}^{GSM} that allow successful signal decoding. Because GSM signals has to penetrate floors and walls and travel a relatively large distance (path 1) to reach the GSM BTS, P_{tx}^{GSM} implicitly takes these parameters into consideration. However, WiFi propagation distance and obstructions between the WiFi AP and the MD (path 2) are different than in GSM. WiFi APs are typically located within a relatively short distance ($<$ 100m) and mostly in the same building. Hence P_{tx}^{WiFi} is typically lower than P_{tx}^{GSM}. In this scenario, a passive localization AN is also separated by different propagation path and obstructions (path 3) with the target MD. Hence, $\mathcal{R}^{\text{WiFi}}$ and $\mathcal{R}^{\text{GSM}}$ are different because of different transmitted powers and operating frequency of their RIs. Without knowing the exact transmitted power at each RI, combining WiFi and GSM signals using only $\mathcal{R}$ measurements is a challenging task.

In theory, we cannot yet combine DRSSI measurements (abbreviated as $\mathcal{D}$) of WiFi and GSM RIs because $\mathcal{D}$ is a frequency-dependent variable (c.f. Equation 4.4). However, for short-range of indoor communication, we expect the environment layout, such as walls and furniture, to be a dominant factor in $\mathcal{D}$ measurements than the operating frequency. Moreover, since GSM service operates on a licensed band and the WiFi in the unlicensed band, we expected to obtain a higher quality of GSM $\mathcal{D}$ measurements compared to measurements over the WiFi RI.

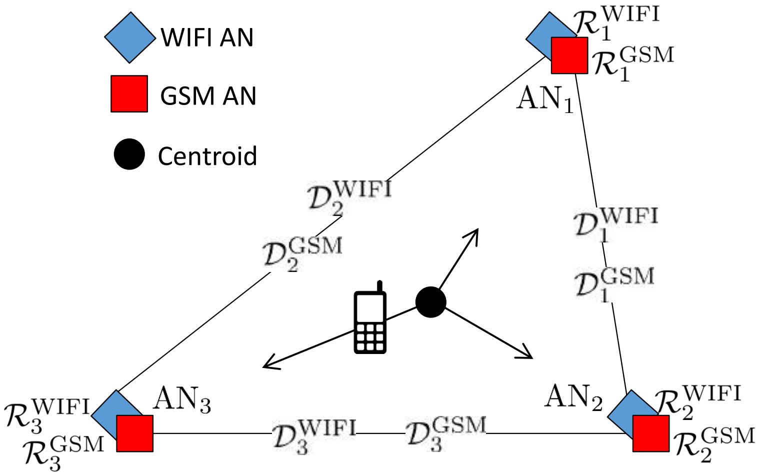

WIFI AN
GSM AN
Centroid
$\mathcal{R}_1^{\mathrm{WIFI}}$
$\mathcal{R}_1^{\mathrm{GSM}}$
AN_1
$\mathcal{D}_2^{\mathrm{WIFI}}$
$\mathcal{D}_2^{\mathrm{GSM}}$
$\mathcal{D}_1^{\mathrm{WIFI}}$
$\mathcal{D}_1^{\mathrm{GSM}}$
$\mathcal{R}_3^{\mathrm{WIFI}}$
$\mathcal{R}_3^{\mathrm{GSM}}$
AN_3
AN_2
$\mathcal{R}_2^{\mathrm{WIFI}}$
$\mathcal{R}_2^{\mathrm{GSM}}$
$\mathcal{D}_3^{\mathrm{WIFI}}$
$\mathcal{D}_3^{\mathrm{GSM}}$

Figure 5.2: Simplified illustration of CDRSS algorithm.

A RSSI-based (abbreviated as $\mathcal{R}$-based) localization system uses signal strength measurements to derive coordinates of a MD. $\mathcal{R}$ acquisition requires simpler hardware and lower processing resources compared to other radio metrics such as TDoA [123]. In existing $\mathcal{R}$-based localization algorithms, a typical approach is to estimate the distance d_i between a MD and an AN_i based on instantaneous $\mathcal{R}_i$ measurements. Estimated distances are then passed to the localization algorithm along with the AN's coordinates (x_i, y_i). Alternatively, the location of a MD is determined relative to the ANs in proximity-based algorithms [123], i.e., d_i is calculated using proximity relation to the AN's coordinates. We developed a multiple of proximity-based algorithms that combine the invisibility requirement and performance reliability in indoor environments. In this work, we chose the CDRSS localization algorithm to verify our proposed solutions [18]. The concept is illustrated in Figure 5.2 as expressed in Section 4.5.1.

5.3 Hybrid-Network Indoor Localization

The proposed hybrid-network solution aims to overcome challenges of indoor propagation in the localization process. As described in Section 5.2, we expect to have a small number of high-quality GSM measurements and a large number of low-quality WiFi measurements. The proposed solution benefits from advanced characteristics of both signals to produce more accurate location estimates. We have two main hybrid solutions: (i) a hybrid signal preprocessing, which deals RSSI measurements from different RIs before being fed to the localization algorithm, and (ii) a hybrid location post-processing, which deals with estimated locations (as an output of the localization algorithm) of multiple RIs. Given the deployment flexibility of our ANs, their capturing antennas (or RIs) can be colocated or dis-

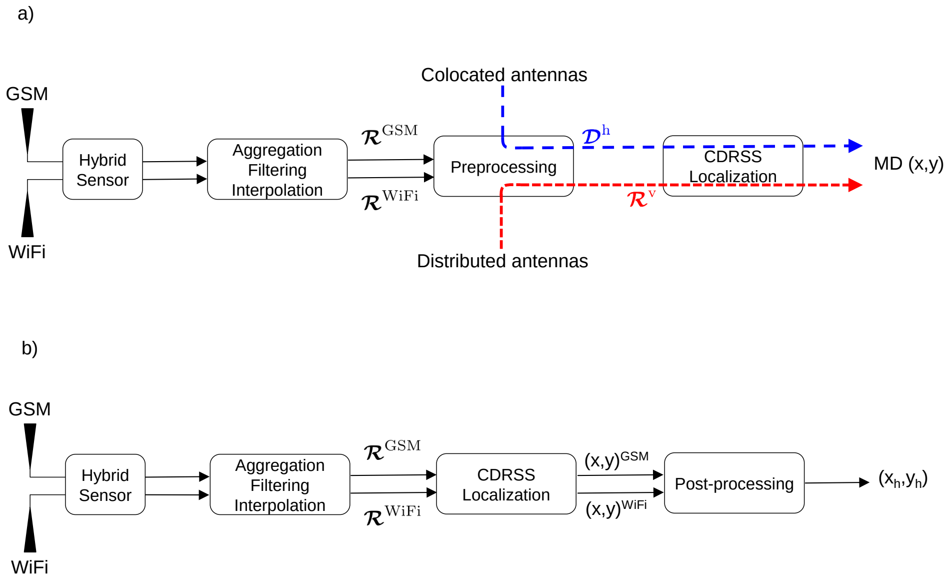


Figure 5.3: a) Preprocessing for colocated and distributed antenna setups. b) Post-processing for both antenna and distributed antenna setups.

tributed as shown in Figure 5.7. First, colocated antenna setup means that WiFi and GSM antennas are deployed at the same location. Second, distributed antenna setup means that WiFi and GSM antennas are deployed separated from each other at different locations. Figure 5.3 shows the processing chains for colocated and distributed antenna setups with signal preprocessing (Figure 5.3-a) and location post-processing (Figure 5.3-b). Note that the post-processing approach expressed in Figure 5.3-b is the same for colocated and distributed antenna setups. The follwoing subsections discuss Figure 5.3 in more details.

5.3.1 Aggregation, Filtering and Interpolation

The CDRSS algorithm sorts instantaneous $\mathcal{R}$ measurements so that their corresponding $\mathcal{D}$ values are always positive. We take our assumption in Section 5.2 that $\mathcal{R}$ and $\mathcal{D}$ measurements will be dominated by the environment layout more than the operating frequency. Hence, hybrid processing of radio metrics from both RIs is possible. However, in the hybrid approach over a period T, we have two main challenges: (i) How do we create synchronized data sets from both RIs? Moreover, (ii) which RI will dominate the CDRSSI sorting process?

To avoid the problem of unsynchronized datasets, we consider the aggregation of $\mathcal{R}$ measurements within a smaller period $t = T/N$, in which we collect measurements from both RIs. N is considered the number of aggregated measurements within T. We chose the median estimator to aggregate all $\mathcal{R}$ measurements within a period t into one measurement. From now on, we deal with aggregated $\mathcal{R}$ measurements. We call them $\mathcal{R}$ for simplicity. If the passive receiver did not catch any signal within a period $t_i;\ i = 1 : N$, we consider linear interpolation at t_i using collected $\mathcal{R}$ measurements surrounding the interpolation point. This happens typically in GSM scenario due:

- A small number of usable transmitted packets for localization (packets with identities).

- Imperfection of the passive receiver capturing process. The capturing rate is around 70%.

Up to this point, we consider having two datasets, $\boldsymbol{\mathcal{R}}^{\text{WiFi}}$ and $\boldsymbol{\mathcal{R}}^{\text{GSM}}$ with N measurements each.

For the CDRSS sorting requirement, we have the flexibility to choose which RI will dominate. For example, if we choose the GSM RI to control the sorting process, $\mathcal{D}_i^{\text{WiFi}},\ i \in \{1 : M\}$ might contain negative values. A negative $\mathcal{D}$ produces weights that push the centroid away from corresponding ANs [18].

5.3.2 Hybrid Signal Preprocessing

The preprocessing approach combines $\mathcal{D}$ measurements of the two RIs with a probability weighting scheme. The basic idea of the probabilistic approach is that radio measurements (within a period T) with higher probability are considered more accurate, i.e., less influenced by the indoor environment, than measurements with low probability.

Colocated Antennas

ANs with colocated antennas consider identical path obstructions (walls and furniture) of WiFi and GSM radio signals. A first step in the CDRSS algorithm is to order ANs based on their instantaneous $\mathcal{R}$ measurements such that $\mathcal{D}$ measurements are always positive. However, in the hybrid approach, we have two sets of $\mathcal{R}$ measurements: one for GSM and another for WiFi. Let's assume that WiFi $\mathcal{R}$ measurements control the sorting process in the CDRSS algorithm. The example in Figure 5.4 considers our processing for the branch between AN$_1$ and AN$_2$. However, this process is applied to all available branches as follows:

- Collect $\mathcal{D}$ measurements for a period T for both RIs.

- Quantize $\boldsymbol{\mathcal{D}}$ into N equally spaced intervals between the minimum and maximum value of $\boldsymbol{\mathcal{D}} = \{\mathcal{D}_1, ..., \mathcal{D}_N\}$. Each of the intervals is called a bin.

- Interpolate bins with no measurements linearly using measurements of surrounding bins.

- Count the number of measurements n_b inside each quantized bin.

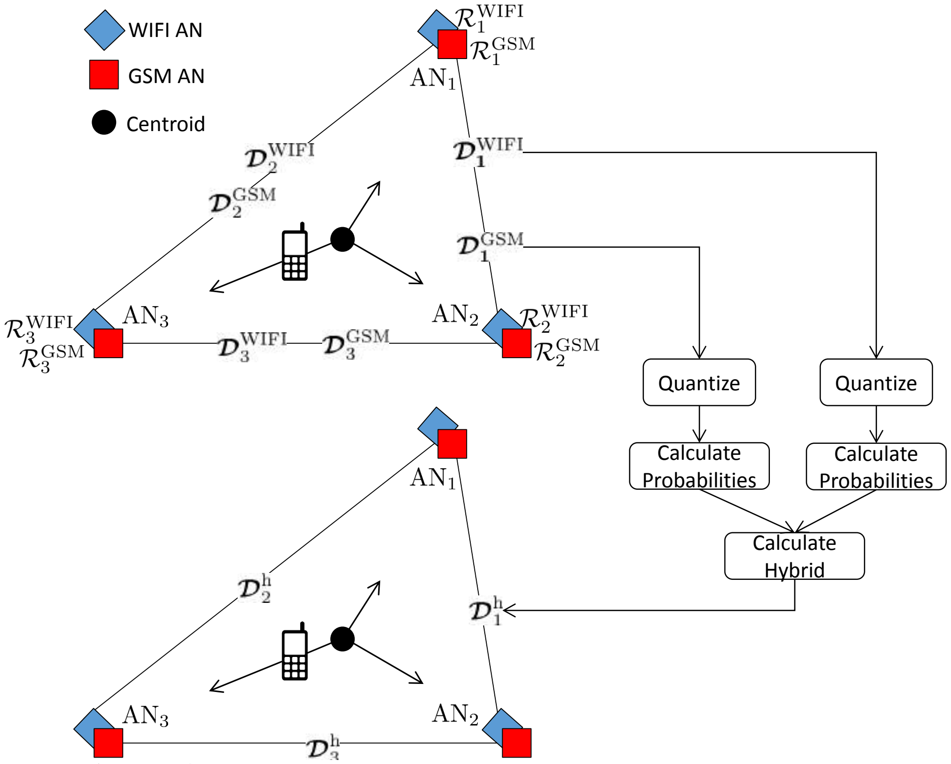

WIFI AN
GSM AN
Centroid
$\mathcal{R}_1^{\text{WIFI}}$
$\mathcal{R}_1^{\text{GSM}}$
AN$_1$
$\mathcal{D}_1^{\text{WIFI}}$
$\mathcal{D}_2^{\text{WIFI}}$
$\mathcal{D}_2^{\text{GSM}}$
$\mathcal{D}_1^{\text{GSM}}$
$\mathcal{R}_3^{\text{WIFI}}$
$\mathcal{R}_3^{\text{GSM}}$
AN$_3$
AN$_2$
$\mathcal{R}_2^{\text{WIFI}}$
$\mathcal{R}_2^{\text{GSM}}$
$\mathcal{D}_3^{\text{WIFI}}$
$\mathcal{D}_3^{\text{GSM}}$
Quantize
Quantize
Calculate Probabilities
Calculate Probabilities
Calculate Hybrid
AN$_1$
$\mathcal{D}_2^{\text{h}}$
$\mathcal{D}_1^{\text{h}}$
AN$_3$
AN$_2$
$\mathcal{D}_3^{\text{h}}$

Figure 5.4: Example of hybrid signal preprocessing of colocated antennas

- Calculate probability of each bin as $P_b = n_b / \sum_{b=1}^{N} n_b$. Figure 5.5 shows *a smoothed* example of obtained probabilities, where $\mathcal{D}_0/\mathcal{D}_1$ and P_0/P_1 are the differential RSSI (bin value) and corresponding probability for both GSM and WiFi measurements, respectively.

- Associate original $\mathcal{D}$ values with their corresponding probabilities.

- Calculate weights of original original $\mathcal{D}$ values as expressed in Equation 5.1. J is the number of RIs, i.e., two in our case.

$$w_b^{\mathrm{j}} = P_b^{\mathrm{j}} / \sum_{j=1}^{J} P_b^{\mathrm{j}},\ j \in \{1, ..., J\} \tag{5.1}$$

Over a period T, we will have a vector of weights $\boldsymbol{w}^{\mathrm{j}}$. Now, we can calculate the hybrid $\boldsymbol{\mathcal{D}}^h$ dataset as illustrated in Equation 5.2.

$$\boldsymbol{\mathcal{D}}^h = \sum_{j=1}^{J} \boldsymbol{w}^{\mathrm{j}}\ \boldsymbol{\mathcal{D}}^{\mathrm{j}} \tag{5.2}$$

When estimating the MD location, we use $\boldsymbol{\mathcal{D}}^h$ as a single dataset for all RIs.

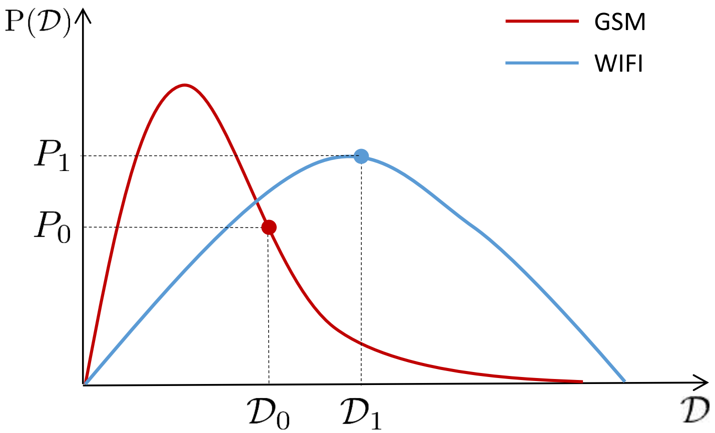


Figure 5.5: Probabilities for WiFi and GSM $\mathcal{D}$ values

Distributed Antennas

For distributed AN's antennas, as shown in Figure 5.3-b, antennas are deployed at different locations. Hence, captured signals over the distributed antennas are no longer sharing the same propagation path. To benefit from the diversity of radio measurements and improve the localization accuracy, we construct a new set of ANs called virtual ANs (VANs), which will be used to estimate the location of a target device (c.f., Figure 5.6-a). VANs have two main characteristics: (i) x- and y-coordinates and (ii) RSSI values. The location of a VAN lies on the path between the x- and y-coordinates of different RIs as illustrated in Figures 5.6-a and 5.6-c. For example, if we have M ANs, each GSM RI will have M VANs with corresponding WiFi RI and vice versa. In total, we will have M^2 VANs contributing to the localization process. Recall that the CDRSS algorithm selects the M ANs (now called VANs) with highest $\mathcal{R}$ measurements. Hence, the high number of VANs will not affect the complexity of the localization process. However, it will add some complexity to calculate VANs' locations and corresponding radio measurements. We consider synchronized $\boldsymbol{\mathcal{R}}^{\text{WiFi}}$ and $\boldsymbol{\mathcal{R}}^{\text{GSM}}$ datasets with N measurements each (c.f. Section 5.3.1). First, we followed the same procedure in Section 5.3.2 to calculated probabilities $\boldsymbol{w}^{\text{j}}$ of all RIs (c.f. Equation 5.1). The hybrid measurements dataset $\boldsymbol{\mathcal{R}}^{v}$ of a virtual AN is calculated based on a probabilistic approach as described in Equation 5.3 (c.f. Figure 5.3-a).

$$\boldsymbol{\mathcal{R}}^{v} = \sum_{j=1}^{J} \boldsymbol{w}^{\text{j}}\, \boldsymbol{\mathcal{R}}^{\text{j}} \tag{5.3}$$

But, the captured power at RIs are not the same. Equation 5.3 is valid under the assumption that P_{tx}^{GSM} and P_{tx}^{WiFi} are constant within a period t. Let A_i be an RI at position (x_i, y_i) for the i^{th} RI, and A_j be an RI at position (x_j, y_j) for the j^{th} RI. Calculated positions of a VAN over a period T $(X_v, Y_v)_b$ is shown in Equation 5.4.

$$(X_v, Y_v)_b = \boldsymbol{w}^{i}(x_i, y_i) + \boldsymbol{w}^{j}(x_j, y_j) \tag{5.4}$$

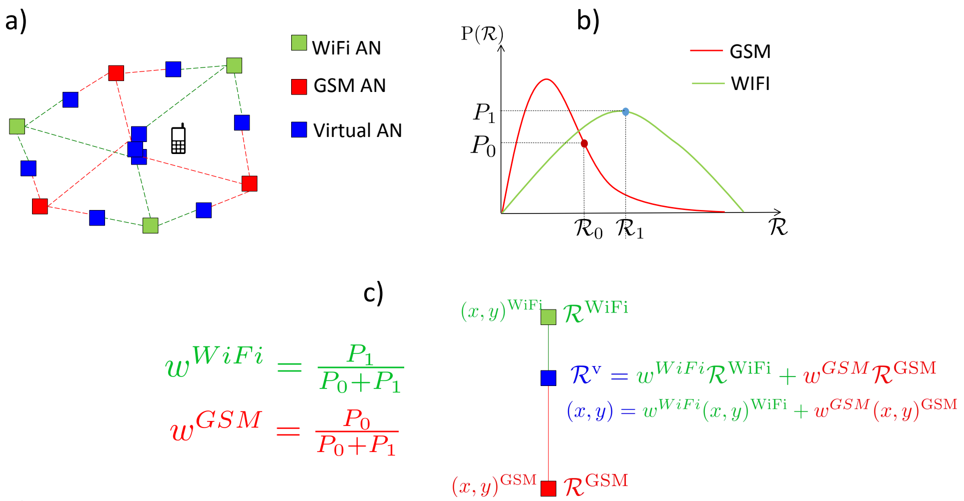

a)
🟩 WiFi AN
🟥 GSM AN
🟦 Virtual AN
b)
$P(\mathcal{R})$
P_1
P_0
$\mathcal{R}_0$
$\mathcal{R}_1$
$\mathcal{R}$
GSM
WIFI
c)
$w^{WiFi} = \frac{P_1}{P_0+P_1}$
$w^{GSM} = \frac{P_0}{P_0+P_1}$
$(x,y)^{\mathrm{WiFi}}$ $\mathcal{R}^{\mathrm{WiFi}}$
$\mathcal{R}^{\mathrm{v}} = w^{WiFi}\mathcal{R}^{\mathrm{WiFi}} + w^{GSM}\mathcal{R}^{\mathrm{GSM}}$
$(x,y) = w^{WiFi}(x,y)^{\mathrm{WiFi}} + w^{GSM}(x,y)^{\mathrm{GSM}}$
$(x,y)^{\mathrm{GSM}}$ $\mathcal{R}^{\mathrm{GSM}}$

Figure 5.6: a) Virtual antenna setup. b) Probability calculation of RSSI measurements. c) RSSI and coordinate calculations for VANs.

We use coordinates and corresponding $\boldsymbol{\mathcal{R}}^v$ of VANs as input parameters to the CDRSS localization algorithm.

5.3.3 Hybrid Location post-processing

With post-processing, we refer to analyzing the output of the localization algorithm. In a first step, MD location estimates of all RIs are calculated independently (c.f. Figure 5.3-b). The localization process is performed using the CDRSS proximity localization algorithm described in Section 4.5.1 for both colocated and distributed antenna deployment setups. We used aggregated $\boldsymbol{\mathcal{R}}$ datasets with N measurements, which then served as input to the localization algorithm. However, the hybrid solution does not depend on the localization algorithm, i.e., is algorithm agnostic. Besides CDRSS, several other algorithms that meet the passive requirement of the system can be used. After gathering the set $\boldsymbol{\mathcal{L}}^{\mathrm{j}} = \{\mathcal{L}_1, ..., \mathcal{L}_N\}$ of N location estimates for each the j^{th} RI, we measure the linear correlation coefficient between the X-axis and the Y-axis distribution of the location estimates. Let X^{j} and Y^{j} contain the x- and y-coordinates of location estimates $\boldsymbol{\mathcal{L}}^{\mathrm{j}}$. The correlation coefficient ρ^{j} of both coordinates is described in Equation 5.5.

$$\rho^{\mathrm{j}} = \rho(X^{\mathrm{j}}, Y^{\mathrm{j}}) = \frac{cov(X^{\mathrm{j}}, Y^{\mathrm{j}})}{\sigma_X^{\mathrm{j}} \sigma_Y^{\mathrm{j}}} \tag{5.5}$$

cov is the covariance measure, σ_X^{j} and σ_Y^{j} are the standard deviations of X^{j} and Y^{j}. ρ^{j} is the pairwise correlation coefficient between each pair of columns in the N-by-1 vectors X^{j} and Y^{j} We define the coordinate weights over a period T as

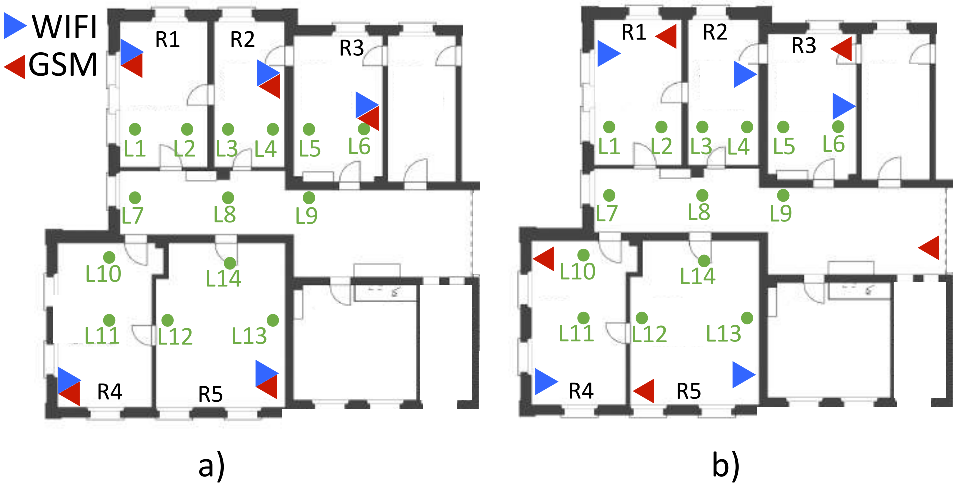


Figure 5.7: a) colocated antennas setup. b) distributed antennas setup

shown in Equation 5.6.

$$w^{\mathrm{j}} = \rho^{\mathrm{j}}/\sum_{j=1}^{J}\rho^{\mathrm{j}} \tag{5.6}$$

Finally, the hybrid estimation of the target MD location (X_h, Y_h) is expressed in Equation 5.7.

$$(X_h, X_h) = \sum_{j=1}^{J} w^{\mathrm{j}}(X^{\mathrm{j}}, Y^{\mathrm{j}}) \tag{5.7}$$

5.4 Experimental Setup

To validate the advantages of our proposed solutions, we set up a testbed with WiFi and GSM ANs. Our experiments were conducted in the office space of the Communication and Distributed Systems (CDS) research group.Figure 5.7 shows the two different antenna setups that were installed for testing colocated and distributed antenna configurations. For both setups, two Google Nexus Smartphones were fixed at seven different locations. At each location, both MDs continuously generated WiFi and GSM traffic over a period of 45 minutes. The continuous WiFi packet transmission was guaranteed by streaming a video over a WiFi VPN connection. Moreover, we used a self-developed Android application that generates continuous uplink GSM MOC messages by making fake calls.

For ANs' deployment, we use six open-mesh product OM2P devices for WiFi signal overhearing and five USRP N210 devices for GSM signal overhearing. The OM2P devices are equipped with a WiFi driver to scan channels and report timestamp, $\mathcal{R}$ and MAC address of overheard packets to a central database. USRPs are connected over an IP network to a central processing machine that host the GSM

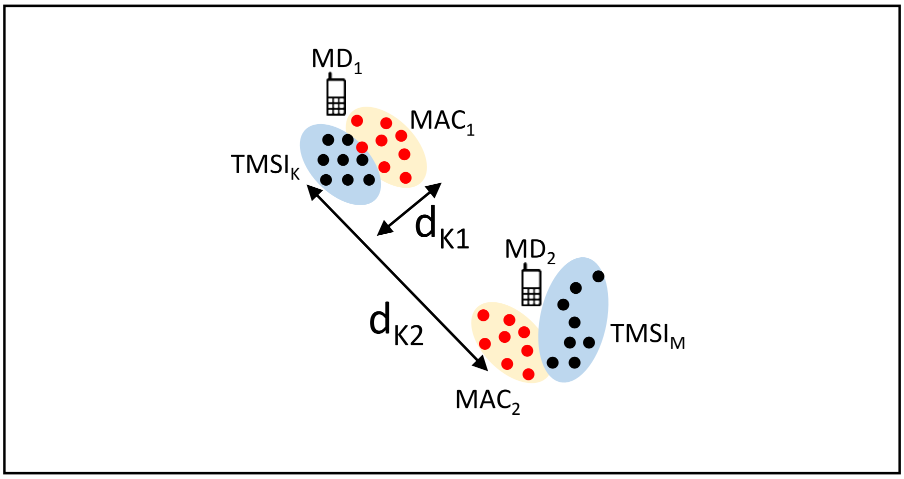


Figure 5.8: Distance between WiFi and GSM measurements.

passive receiver tool [18]. USRPs are tuned to capture a set of uplink frequencies of surrounding GSM BTSs, where the target MD might connect. The GSM receiver tool reports a timestamp, $\mathcal{R}$, and TMSI to a central database.

In GSM networks, a MD's TMSI is controlled by the network and changes over time (sometimes within one experiment). Hence, there is no direct relationship between the MD's MAC address over the WiFi RI and TMSI over the GSM RI. This causes a problem to extract and process radio measurements of multiple MDs simultaneously. Figure 5.8 shows the estimated locations of two MDs using both WiFi and GSM RIs. Inside the localization system, there are four sets of estimated locations and corresponding identities. However, our hybrid solution requires a mapping between identities and corresponding location estimated for each particular MD. For example in Figure 5.8, we have to map accurately estimated locations of MAC_1 and TMSI_k to one MD_1. To overcome this issue, we built the following algorithm in our hybrid system:

- Localize MDs using WiFi and GSM signals independently and tag estimated locations with their RI identity (MAC or TMSI).

- Measure distances between estimated locations (centre of mass) of WiFi and GSM signals, e.g., $\mathrm{d}_{\mathrm{K}1}$ is the distance between GSM estimated locations with TMSI_K identity and WiFi measurements with MAC_1 identity.

- Correlate identities of short distances to a MD, e.g., $\mathrm{d}_{\mathrm{K}1} < \mathrm{d}_{\mathrm{K}2}$ and hence, TMSI_K and MAC_1 are identities for one MD.

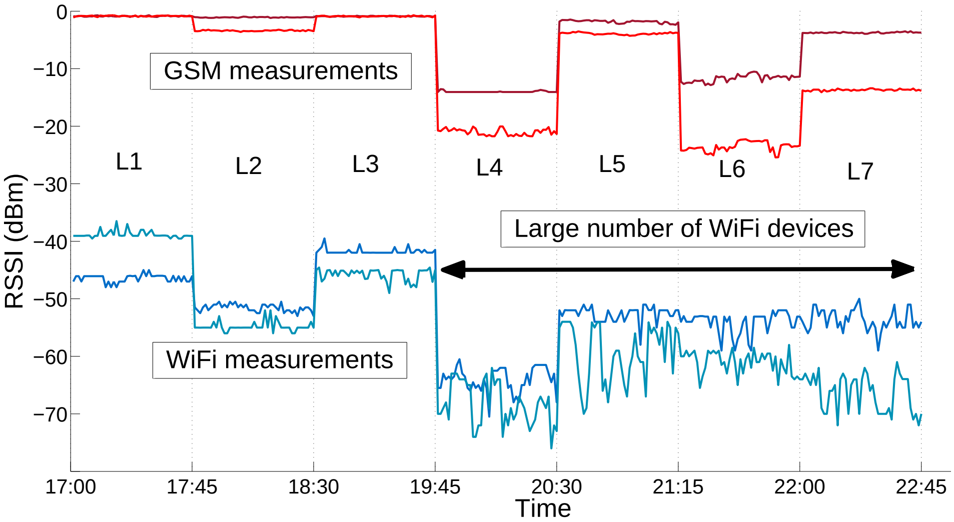


Figure 5.9: Comparing Signal Interference on colocated Antennas

5.5 Experimental Results

5.5.1 Signal Quality

In the first experiment, we compare the behavior of WiFi and GSM signals indoors. We set a MD with two active RIs at seven different locations (L1: L7). Results in Figure 5.9 show the received power of WiFi and GSM signals at ANs in rooms R4 ad R5. We summarize our observations as follows:

- There is a clear difference in $\mathcal{R}$ measurement levels: WiFi measurements are in the range of -35 dBm to -75 dBm and GSM measurements are in the range of 0 dBm and -25 dBm.

- Both WiFi and GSM signals react in a similar way to changes of the MD location.

- GSM signals are more stable than WiFi. This is because GSM operates at a lower frequency and in a licensed band.

- Evening activities, such as crowd events, imply a large number of active WiFi devices. Hence, we observe a big influence on WiFi signal quality (high fluctuations) during the period of crowd activities.

However, since GSM messages are less frequent when compared to WiFi messages, we nominate the WiFi RI to be dominant in sorting ANs for the CDRSS localization process. The high fluctuations in WiFi measurements can be limited using outliers detection and mitigation algorithms.

5.5.2 Received Power vs. Distance

Our proposed solution is based on the assumption that the indoor environment, such as walls and furniture, dominate radio measurements more than the operating frequency. To justify this assumption, we collected measurements from 7 different locations (L1: L7) as shown in Figure 5.7-a for both WiFi and GSM RIs. In these experiments, we used our knowledge of the exact separation distance between the MD and ANs with colocated antennas. Note that at each location (L1: L7), there is an unequal number of obstacles and walls separating the MD from ANs. The duration of each experiment at locations L1: L7 is 45 min using a static MD. The MD generates continuously GSM and WiFi uplink messages. Figure 5.10 shows the relationship between path loss (dBm) and separation distance. The axis of WiFi measurements is shifted 10 dBm to make the visual comparison easier. We define the path loss as the difference between P_{tx} and $\mathcal{R}$. However, in passive systems, we do not know the instantaneous value of P_{tx}. To calculate the path loss of each radio signal, we approximate P_{tx} to be the maximum observed $\mathcal{R}$ over a period T. From Figure 5.10, we draw the following conclusions:

- Fluctuations in path loss measurements vary at different locations. This is because different directions contain different numbers and types of obstacles that downgrade the radio signals with different values.

- The path loss of WiFi and GSM signals is different at fixed locations. This is because our algorithm lacks the knowledge of instantaneous P_{tx} values. It only estimates P_{tx} based on the maximum received $\mathcal{R}$, which causes this difference.

- The best line fit of measurements in Figure 5.10 represent the PLE α. The PLE of WiFi measurements is higher than GSM signals. This is because of the higher operating frequency of WiFi.

From these conclusions, we confirm our assumption in Section 5.2 that the indoor environment dominates the signal behavior more than the operating frequency.

5.5.3 Hybrid Localization

To verify our hybrid localization solutions proposed in Section 5.3, we conducted experiments at 14 locations using two static MDs with active WiFi and GSM RIs. The duration of each experiment was 45 min in each setup. We created quantized and aggregated $\boldsymbol{\mathcal{R}}^{\text{WiFi}}$ and $\boldsymbol{\mathcal{R}}^{\text{GSM}}$ datasets with $N = 6$ measurements each over a period T = 1 min and t = 10 seconds. Probabilities of WiFi and GSM signals are calculated every 10 seconds. Therefore, we consider the MD static and transmitting at a constant power during this period. Tables 5.1 and 5.1 show the localization error at 14 locations and using different localization approaches and performance metrics. A graphical representation of Tables 5.1 and 5.1 is shown in Figure 5.12. From Tables 5.1 and 5.1, we draw the following conclusions:

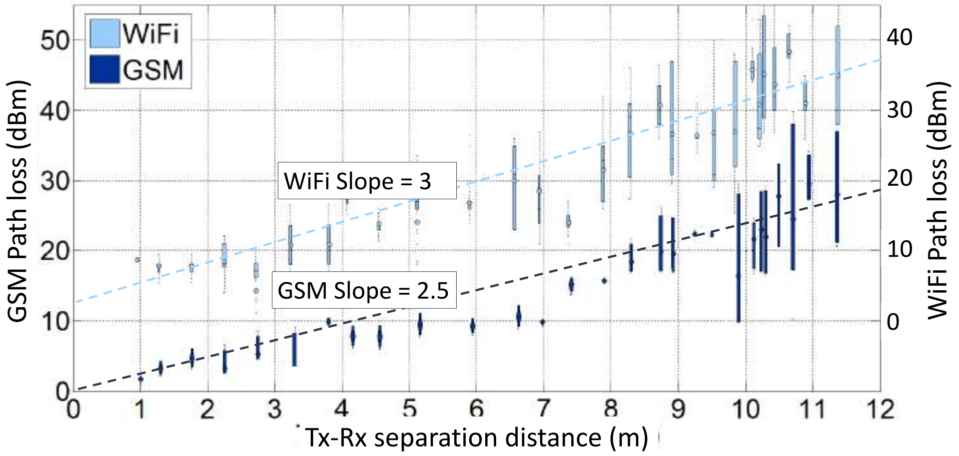


Figure 5.10: Comparing WiFi and GSM path loss

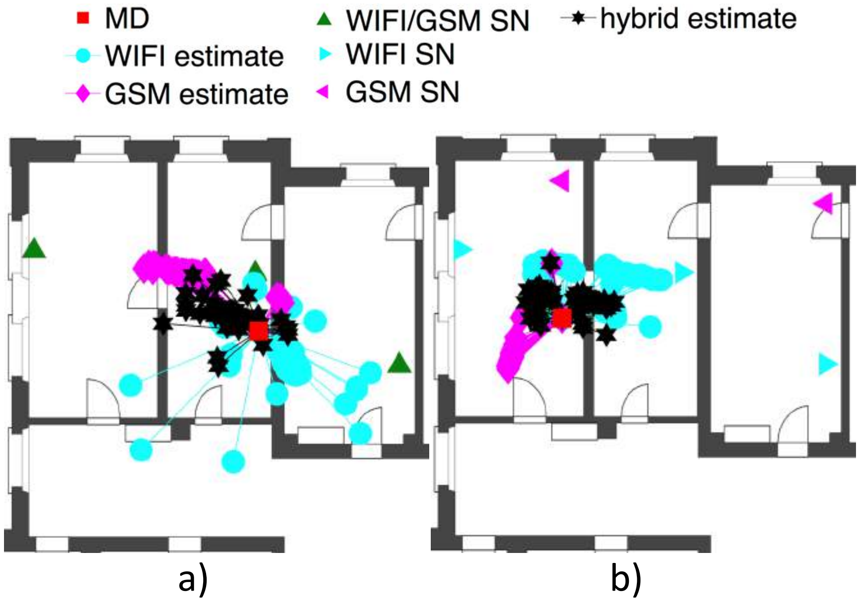


Figure 5.11: Estimated location of hybrid localization system, a) colocated antenna setup, b) distributed antenna setup.

- GSM-based localization shows comparable localization mean error mean of 3.41m and 3.20m for colocated and distributed antenna setups. The location of GSM antennas in both setup is explained in Figure 5.7. These results show the reliability of CDRSS algorithm with different setups (locations) of ANs.

- The WiFi antenna location did not change in both setups. Hence, we have one set of results for WiFi-based localization. Moreover, WiFi-based localization shows better localization accuracy of 2.61m (error mean) than GSM-based localization. This is because the mean estimator of WiFi signals over

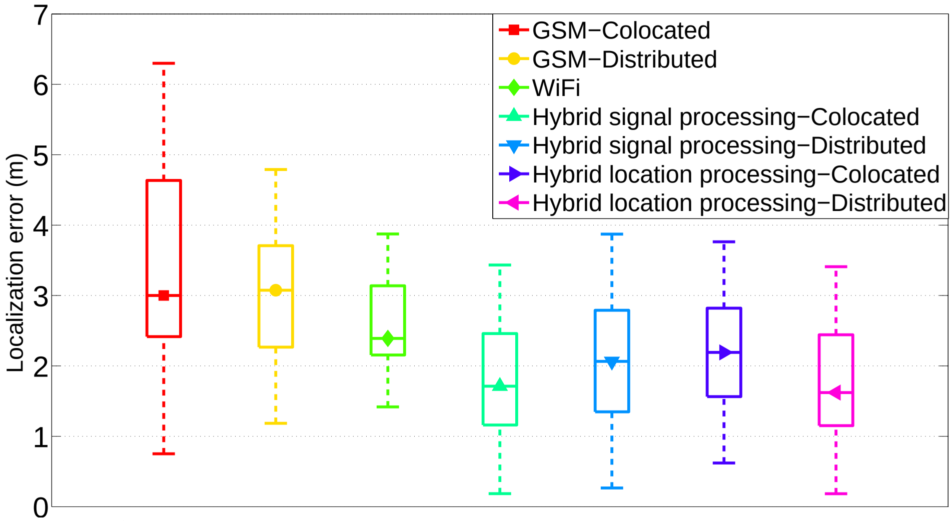


Figure 5.12: Comparison between hybrid and non-hybrid algorithms.

a period of one minute and relatively large number of captured packets is more robust than GSM signals with relatively few packets.

- Results of hybrid localization solutions show better performance than original non-hybrid solutions. The hybrid location post-processing solution with distributed antenna setup shows the best localization error mean of 1.71m. However, this setup might look like doubling the amount of ANs. The hybrid signal preprocessing solution with colocated antennas leverages radio signals most efficiently and achieves comparable results of 1.95m.

Table 5.1: Performance evaluation of WiFi and GSM localization systems

Location	**GSM**						**WiFi**		
	Colocated			Distributed					
	μ	σ	RMS	μ	σ	RMS	μ	σ	RMS
L1	3.86	1.15	4.02	3.07	0.99	3.22	2.23	1.72	2.81
L2	1.56	0.30	1.58	2.01	1.03	2.25	1.87	0.61	1.96
L3	2.75	1.24	3.01	3.80	0.24	3.80	3.12	1.05	3.29
L4	2.78	0.05	2.78	2.92	0.63	2.98	1.75	0.40	1.79
L5	2.44	0.48	2.48	3.60	0.57	3.64	2.13	1.35	2.52
L6	5.63	0.08	5.63	4.73	0.84	4.80	3.15	0.45	3.18
L7	2.04	0.17	2.04	2.89	0.92	3.03	2.79	0.91	2.93
L8	5.97	1.44	6.14	3.67	0.98	3.79	4.36	1.25	4.53
L9	3.21	1.22	3.43	5.05	0.35	5.06	2.27	1.60	2.77
L10	2.38	0.55	2.44	2.26	0.88	2.42	2.17	1.72	2.76
L11	4.81	1.67	5.09	2.16	0.85	2.32	3.18	0.98	3.32
L12	4.45	0.06	4.45	2.26	0.21	2.26	2.50	0.24	2.51
L13	3.02	0.77	3.11	3.27	0.15	3.27	2.61	0.26	2.62
L14	3.14	0.67	3.21	3.12	0.64	3.18	2.42	0.46	2.46
Mean	3.43	0.70	3.50	3.20	0.66	3.26	2.61	0.92	2.76

Table 5.2: Performance evaluation of the proposed hybrid-network localization

Location	**Hybrid**											
	Colocated						Distributed					
	Signal			Location			Signal			Location		
		processing							processing			
	μ	σ	RMS	μ	σ	RMS	μ	σ	RMS	μ	σ	RMS
L1	2.48	1.00	2.67	2.48	1.28	2.79	2.06	0.81	2.21	2.67	0.32	2.68
L2	1.04	0.30	1.08	1.07	0.40	1.14	1.28	0.50	1.37	0.34	1.27	1.31
L3	1.48	0.97	1.76	1.34	1.06	1.70	1.96	0.24	1.97	1.62	0.21	1.63
L4	1.93	0.29	1.95	0.87	1.05	1.36	1.40	1.19	1.83	1.65	1.15	2.01
L5	2.43	1.11	2.67	1.96	0.53	2.03	2.74	0.46	2.77	0.37	0.75	0.83
L6	1.10	0.42	1.17	3.05	0.79	3.15	2.93	0.79	3.03	1.28	1.39	1.88
L7	1.21	0.23	1.23	2.41	0.10	2.41	1.36	0.24	1.38	3.60	0.10	3.60
L8	4.17	0.30	4.18	4.79	0.07	4.79	1.10	0.90	1.42	3.05	0.61	3.11
L9	2.24	0.73	2.35	2.58	0.74	2.68	2.30	0.39	2.33	0.97	0.14	0.98
L10	1.36	0.56	1.47	1.78	1.09	2.08	3.20	0.98	3.34	1.56	1.34	2.05
L11	2.77	0.42	2.80	3.69	1.30	3.91	2.45	1.03	2.65	2.15	0.00	2.15
L12	0.39	0.99	1.06	1.82	0.18	1.82	3.09	1.12	3.28	0.86	1.08	1.38
L13	0.82	0.70	1.07	2.43	0.79	2.55	1.15	0.67	1.33	2.21	1.14	2.48
L14	3.98	0.65	4.03	2.01	0.65	2.11	1.97	0.12	1.97	1.67	1.21	2.06
Mean	1.95	0.61	2.04	2.30	0.71	2.40	2.07	0.67	2.17	1.71	0.76	1.87

5.6 Conclusions

To improve the localization performance of passive localization systems, we proposed different hybrid solutions. The proposed solutions rely on leveraging radio information for multiple radio interfaces of the target MD. To evaluate the performance of the proposed hybrid solutions, we conducted real indoor experiments at 14 different locations. Compared to non-hybrid localization, the hybrid approach shows an improvement in localization accuracy. The preprocessing option with colocated and distributed antenna setups shows a mean error distance of 1.95m and 2.07m, respectively. When performing location analysis in a post-processing step, we achieve mean error distances of 2.30m and 1.71m for colocated and distributed antennas, respectively. The improvement compared to the non-hybrid options illustrates the validity of our approach. We attribute this improvement to the increased resistance against unpredictable signal behavior due to NLOS reception and multipath propagation. Further increase in accuracy can be expected if more radio interfaces are encountered in the hybrid solutions or the case of active localization with range-based localization solutions.

Part II

Network-Based Localization

Part II focuses on active localization systems; more specifically, network-based positioning in LTE network. Our target research is to provide positioning and tracking solutions at the eNB level without additional signaling to the backhaul of the LTE network. Network-based localization has several advantages over passive-based one being previously discussed in Part I. In terms of system control, network-based localization has access to fine-grained information, such as transmitted power or timing advance. For tracking applications, network-based solutions, especially in LTE, support higher packet rates, which is important for short response time and accurate positioning. Moreover, network-based localization requires less involvement of target UEs than UE-assisted localization (data acquisition and processing happen on the network side). However, network-based solutions require overhearing of UE uplink transmission at a set of neighboring eNBs. Running LTE eNBs in a cloud environment allows us to exchange required information for signal overhearing. We study in Chapter 6 the possibility to operate a LTE eNB in the cloud environment. Then we detect the minimum CPU processing power (GHz) required for real-time operation. Based on these results, we build our indoor setup for further real-time experimentation. Finally, we propose in Chapter 7 a LTE network-based localization and tracking algorithm. The proposed solution relies on accessing fine-grained information from LTE eNB to detect outliers and modeling the environment using online measurements at the network-side.

Chapter 6

Real-Time LTE RAN Validation for SDR

6.1 Introduction

The research of this chapter is considered as a preliminary study that covers the real-time operation of LTE eNB in a cloud environment. The possible advantages of operating LTE eNB in a cloud environment can allow us to overcome challenges related to network-based positioning as follows:

- Operating eNBs in a cloud environment (especially within the same processing frame) allows the serving eNB to exchange target UEs' spectrum allocation with neighbouring eNBs through fast connections with low latencies (faster than any physical X2 interface defined in Section 2.4).

- By knowing the spectrum allocation to a certain UE, neighboring eNBs will be able to overhear the UE transmissions as interference without having enough power to decode them. However, this is sufficient to provide the positioning algorithm with required RSSI measurements.

- For time-based algorithms, such as Uplink Time Difference of Arrival (UTDoA), the cloud environment offers precise synchronization between different eNB instances operating within the same processing frame because all instances are using the same CPU clock. Recent models of Intel CPUs have synchronized clocks across all cores.

Figure 6.1 shows our proposed model for network-based positioning and tracking, where different eNB instances are running in a virtualized environment within the same processing frame. eNBs exchanged scheduling information required for signal overhearing using a software implementation of X2-interface (called X2-like interface). Finally, different location based services can be implemented at the top of eNBs. We would like first to answer in this section the following questions. First, can we operate an LTE CRAN in real-time on a commodity hardware (as a

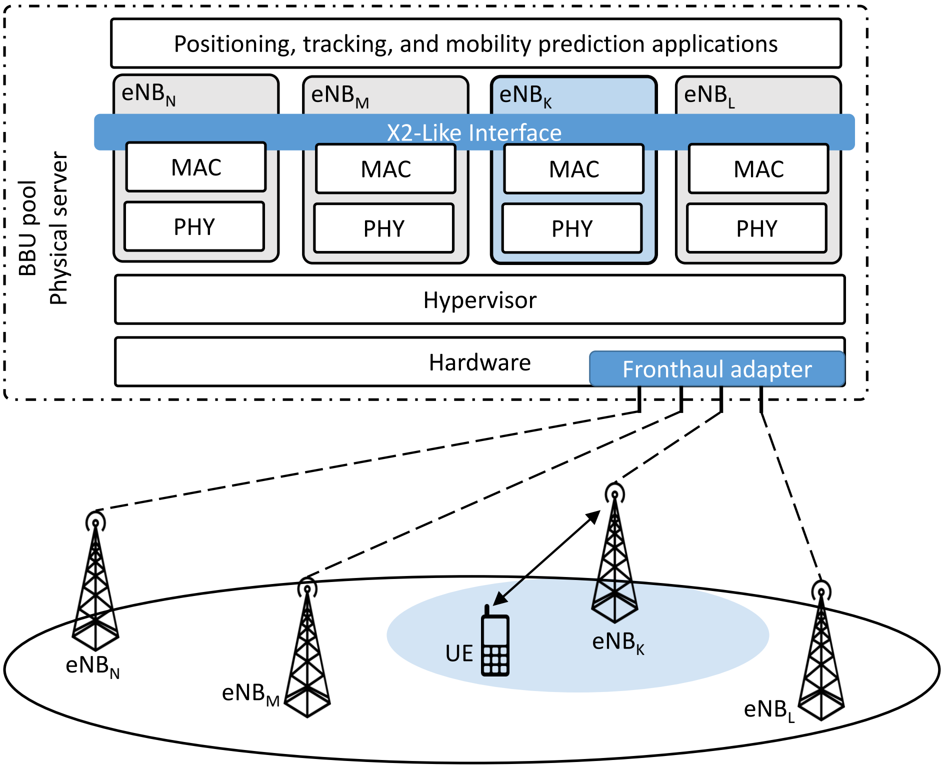


Figure 6.1: Proposed architecture for LTE network-based positioning.

SDR application)? Second, what are the minimum requirements for such an operation?

In this chapter, we evaluate the performance of OAI using GPPs and cloud environments to execute the LTE FDD PHY layer. Our results are based on running the OAI LTE-Release-8 on a cloud platform. This approach provides a precise method for the estimation of required processing resources to handle adequately traffic load by identifying processing bottlenecks. Our main contribution (as a group work in [24]) is a description of bottlenecks in cloud environments and a suggestion of a new execution model of the OAI-based LTE PHY layer.

6.2 Processing Budget in LTE FDD

From the functional perspective, every eNB consists of a signal processing BBU and an RRH [50], which handles transmitting and receiving; they both combined provide the RAN.

In this chapter, we put a particular focus on LTE FDD, which consists of the following layers: i) LTE PHY layer with symbol-level processing, ii) MAC, which supports wide-band multiuser scheduling and HARQ. The LTE PHY layer uses an asymmetric access scheme consisting of OFDMA on the downlink and SC-FDMA on the uplink. The effective data rate over the air interface (goodput) is controlled by the MCS. According to the 3GPP standards, the MCS index varies between 0 and 27 and describes the modulation such as QPSK, 16QAM, and 64QAM as well as the coding rate. The modulation decides on the number of bits per symbol while the code rate defines the amount of redundant information inserted into the data stream [50, 34]. In the LTE PHY layer, the smallest chunk of data transmitted by the LTE eNB is called PRB. The LTE technology uses radio channels of 1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz, and 20 MHz, which can allocate 6, 15, 25, 50, 75, and 100 PRBs respectively. Consequently, MCS index, the number of PRBs, and radio signal bandwidth have an impact on the generated workload.

6.2.1 RAN on a PC

As illustrated in Figure 2.23, we focus on the PHY layer, which is the signal processing bottleneck as mentioned in Sec. 6.2. To run a BBU on a signal processing pool equipped with typical GPP-based computers, we have to operate software-based equivalents of the functionality provided by dedicated hardware previously. The processing architecture consists of the Operating System (OS) equipped with drivers to the CPRI interface and the BBU functionalities implemented as software applications. In the following, we further describe the required properties of the OS and applications deployed.

A general purpose OS requires a kernel, which uses scheduling algorithms to provide processing time to applications. In theory, the kernel can instantaneously suspend every user-level task, but in practice, some parts of the kernel code are not preemptive and introduce unpredictable delays. Due to the fact that in CRAN, BBU is an application, it has to be provided with processing time within a short interrupt-response delay of 100 μs [176] and be able to uninterruptedly process the task within a given processing time window. Such a processing scheme cannot be secured by a typical OS, and therefore a Real-Time (RT) OS such as RTLinux is required at the BBU to provide appropriate timing and to avoid processing time fluctuations in our target scenarios. Also, the OAI process, which executes signal processing on the RTLinux operating system, has to be prioritized.

6.2.2 Cloud-based LTE RAN

Our starting point is OpenStack, a well-known cloud management system that orchestrates various resources such as compute, storage and networking resources to control and manage the execution of VMs on the physical server pool at a datacenter. The task of OpenStack is to configure VMs on physical host machines. As illustrated in Figure 6.3, our modifications of the typical execution stack include the installation of the RTLinux kernel on the host machine. On the host, we pri-

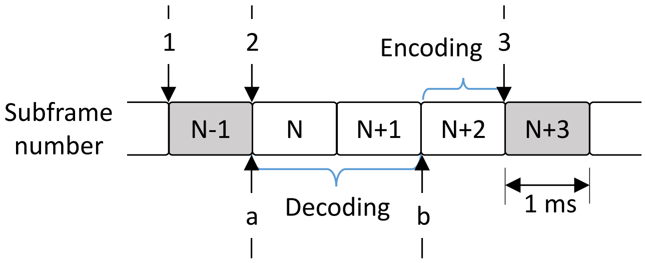


Figure 6.2: Processing orders in OAI.

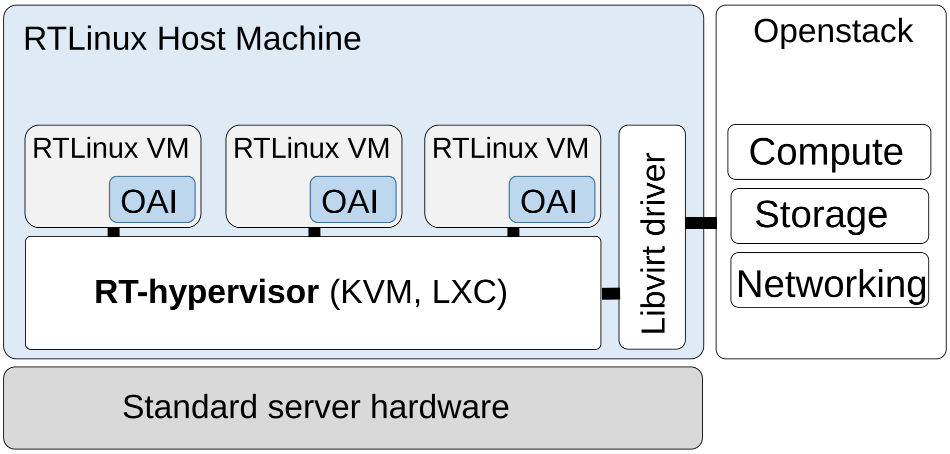


Figure 6.3: OpenStack management architecture.

oritize the Kernel-based Virtual Machine (KVM) hypervisor, which is one of the most popular hypervisors of type 2 in the OpenStack community (we refer to it as RT-hypervisor). We also installed and configured Linux Containers (LXC). The primary benefit of LXC over KVM is the high performance due to the native execution of all CPU instructions. The real-time prioritization of the KVM process is provided through the `chrt` Linux command, e.g., `chrt -p --rr 1 {pid}`. Real-time computing on the guest also requires the installation of the RTLinux kernel and the real-time prioritization of the OAI application in both KVM and LXC.

6.3 Evaluation

In this section, we present our evaluation of the architecture described in Sec. 6.2.2. We are interested in processing delays of the OAI application for downlink and uplink processing on VMs and physical (not virtualized) machines (c.f., Section 2.7.2, Figure 2.23). This procedure is important for understanding whether Cloud-RAN based on OpenStack, KVM or LXC, and Linux can provide satisfactory processing deadlines and could be used as an execution platform for RAN.

6.3.1 Experiments

The setup of our testbed is as follows. A dedicated Intel i5 GPP with configurable CPU frequencies between 1600 and 3300 MHz is used. The memory resources are always fixed to 2 GB RAM (on the physical and virtual testbeds). All the machines (hosts and guests) operate the Ubuntu Linux 12.04 distribution; the kernel version deployed is 3.12.

We are using OAI as a benchmark for profiling the processing time of the LTE PHY layer given different load scenarios configured through PRBs, MCS indices, and radio signal bandwidth. More specifically, we are using two benchmarking tools called “dlsim” and “ulsim” emulating (without any real transmission) the PDSCH and the PUSCH respectively. Both tools are designed to emulate the behavior of the eNB and UE PHY layers over a simulated wireless medium. The traffic load is considered as the maximum amount provided by the emulated system bandwidth and the wireless channel. The emulated data is randomly generated inside each benchmark. We operate all experiments using Additive White Gaussian Noise (AWGN) wireless channel, 16-bit log-likelihood ratios turbo decoder, and high (30 dB) signal to noise ratio (SNR). When the SNR value is high, the processing time variation of the turbo decoder is limited. Hence, we keep the focus on the processing time variation of different platforms and configurations.

The execution time of each signal processing module of downlink and uplink is calculated using timestamps at the beginning and the end of computing. OAI uses the *RDTSC* instruction implemented on all x86 and x64 processors as of the Pentium processors to get very precise timestamps. *RDTSC* counts the number of CPU clocks since a reset. Therefore, the execution time is proportional to the value returned by the following algorithm:

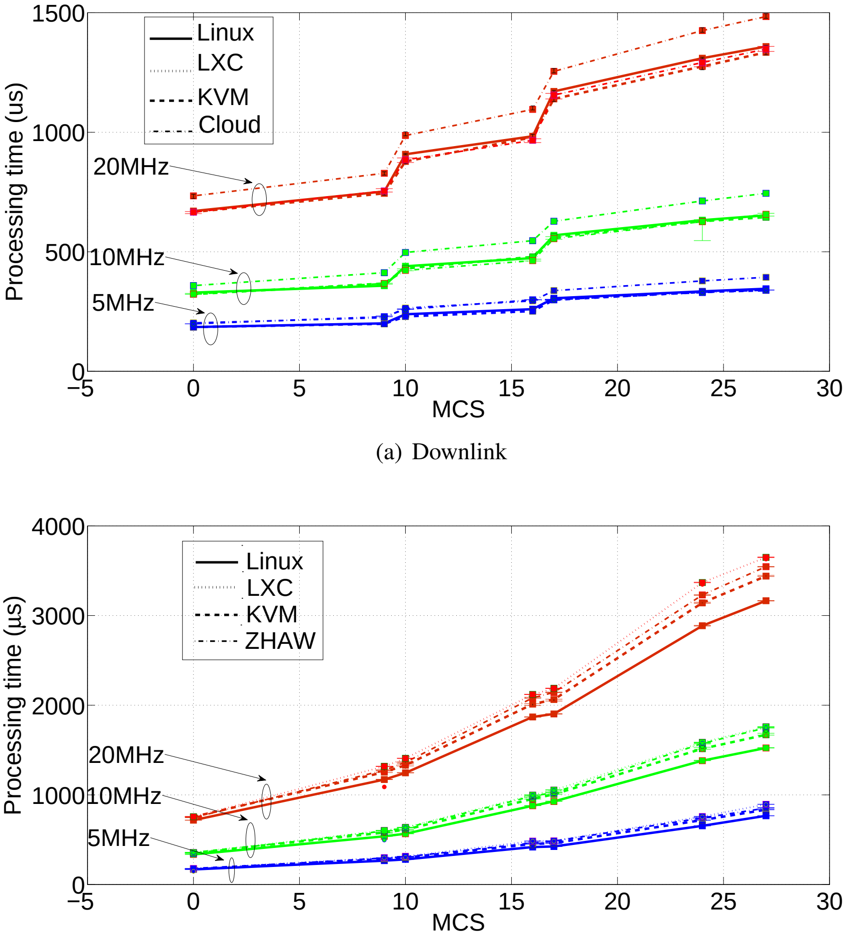
```
start = rdtsc();
compute();
stop = rdtsc();
diff = stop - start;
return diff;
```

To get the processing time T_{Process} in seconds, we have to divide `diff` by the CPU frequency. For statistical analysis, we first gather a large number of `diff` samples (i.e., 10000) to calculate the median, first quantile, third quantile, minimum, and maximum processing time for all subframes in uplink and downlink at the BBU side.

6.3.2 Results and Analysis

Processing Time

In this section, we study the decoding/encoding processing time by using the receiver/transmitter part of OAI ulsim/dlsim with fixed CPU frequency equal to 2.4



(a) Downlink

(b) Uplink

Figure 6.4: eNB processing time for transmitting packets.

GHz, as illustrated in Figs. 6.4(a), 6.4(b). In each figure, we plot the overall processing time for various modulation and coding schemes (MCS: 0, 9, 10, 16, 17, 24, and 27), radio signal bandwidth (5, 10 and 20 MHz), and machine environments: Dedicated Linux, KVM, LXC, and cloud (An Openstack-based private cloud [2]). The cloud environments are based on the KVM hypervisor without RT support. In our figures, the 1st and 3rd quantiles are denoted with short horizontal lines, which in most of the cases lie very close to each other; medians are depicted with filled squares. From our figures, we can draw the following conclusions. The decoding and encoding processing time for LTE subframes grow with the increase of MCS index for 25, 50, 100 PRB and 5, 10, 20 MHz bandwidth. On average, the decoding time is twice as long as the encoding time. Hence, the sequential organization of OAI including 2 ms for decoding and 1 ms for encoding is sound. Given that all the considered machines have the same RAM size and CPU frequency, we see that the median values for the cloud VM show lower performance (more processing time) than KVM or LXC VMs and Linux machines without virtualization; this is probably due to resource sharing and slightly different CPUs deployed in the cloud environment. Notice that in the case of OAI, signal processing is executed on a per subframe basis. The encoding process starts 2 ms after the decoding process. Therefore, the decoding process should not run for more than 2 ms while, in such a case, the encoder has to assemble a Nack message (even if the message were correctly received). This behavior increases retransmission rates and degrades the overall goodput.

Required CPU Frequency

It is important from the system design point of view to understand the minimum required CPU frequency that fully supports a given data rate. Figure 6.5 illustrates the processing time of an eNB (decoding SC-FDMA subframe and encoding OFDMA subframe) given different CPU frequencies (1.6, 2.0, 2.4, 2.8, and 3.3 GHz). Note that we consider the worst case scenario defined by the LTE standards, which stated that UE transmits a PUSCH subframe with MCS index equals to 27 (UE category 5) and the eNB transmits the Ack/Nack using a PDSCH channel. To perform experiments with different CPU frequencies, we used Linux *cpupower* tool to limit the available CPU clock [144]. In Figure 6.5, we observe the reciprocal behavior of the processing time against CPU frequency. We, therefore, fit a model of the processing time $\mathrm{T}_{\text{subframe}}$ valid for any Intel-based processor, which is expressed by the following formula:

$$T_{\text{subframe}}(x)\ [\text{us}] = \alpha/x,$$

$\alpha = 11740 \pm 26$ for uplink MCS = 27 or $\alpha = 8092 \pm 34$ for uplink MCS = 16 (for currently existing UE of category 4) and x is CPU frequency measured in GHz (note that the downlink MCS is always equal to 27). This formula allows us to estimate that cloud operators require VMs with at least 4 GHz CPUs to support the LTE-FDD PHY layer with maximum load.

6.4 Conclusions

Cloudification of LTE radio RAN is a fundamental element for network-based positioning and tracking application, where multiple neighboring eNB can exchange the spectrum allocation of target devices within the same computing frame. In this chapter, we have studied and analyzed several important aspects of the LTE radio access network cloudification. The operation of LTE eNB in a virtualized environment is important to access fine-grained radio measurements of multiple eNBs. To ensure the real-time operation of LTE eNB, we have evaluated OAI eNB implementation in different environments such as dedicated Linux, LXC, KVM, and KVM-based Clouds. Our findings are manyfold. First, we have benchmarked

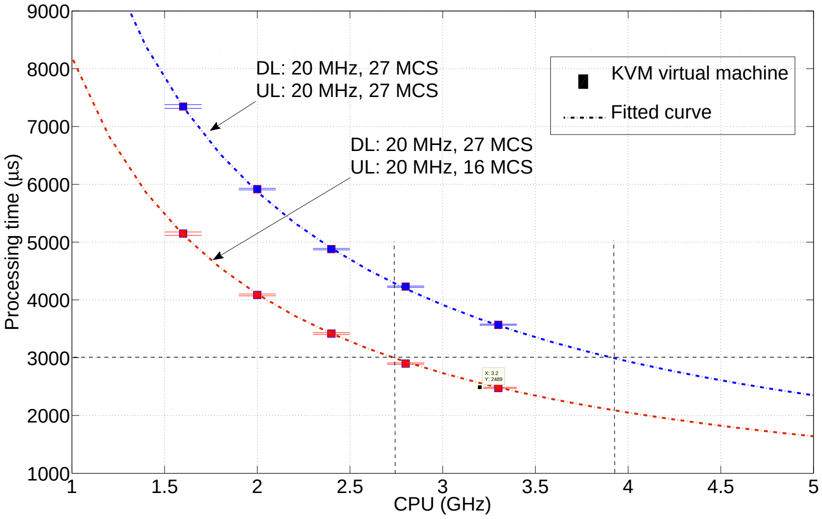


Figure 6.5: Processing time for transmitting (OFDMA) and receiving (SC-FDMA) packets at eNB given full PRB allocation at 20 MHz.

the encoding and decoding workload of the LTE FDD PHY layer subframes on GPP and cloud environments by using the OAI. Second, the reciprocal behavior of the OAI execution time on the Intel CPU family was illustrated. Therefore, the processing power required for any physical subframe processing can be estimated. Finally, the bottlenecks of the OAI cloud execution were identified, and new models of cloud execution were suggested.

Chapter 7

LTE-Based Localization Systems

7.1 Introduction

Real-time network-based positioning and tracking in LTE contains three main steps as shown in Figure 7.1: (i) operation of a LTE eNB in a cloud environment, (ii) simultaneous data acquisition from a set of eNBs, and (iii) a localization algorithm. We validated in Chapter 6 step (i): the possibility to run an eNB implementation in a cloud environment and estimated the minimum CPU requirements for real-time operation. However, for step (ii), neighboring eNBs should tune their RRHs to capture UE uplink messages (similar to CoMP uplink discussed in Section 2.4.5). The cloud environment is a key enabler to exchange in real-time the spectrum allocation of target UEs among neighboring eNBs [23, 22, 139]. However, this approach is faced with a set of challenges:

- Current implementation of OAI eNB does not support X2-interface. Up to our knowledge, there is no open-source implementation of a LTE eNB that support it. This will not affect our proposed positioning algorithms but the data acquisition approach (c.f., Section 7.3).

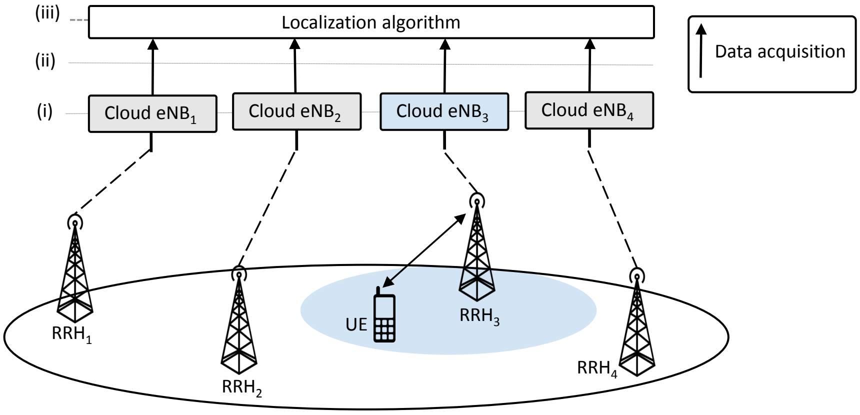


Figure 7.1: Network-based positioning in LTE.

- The power control algorithm inside the serving eNB ensures that UE messages are only decodable at its side (not other neighboring eNBs).

Hence, step (ii) is considered an open issue for future work and we focus our research on step (iii). For indoor environments, we propose RTPoT, a network-based positioning and tracking algorithm that operate at the edge of a LTE network. RTPoT provides a novel calibration technique to characterize the area of interest using the available infrastructure online and without interference with target devices (data acquisition and processing steps are done on the network side). Thus, RTPoT achieves its goal of accurate, fast, and environment agnostic positioning services in large-scale and heterogeneous deployments without any special hardware or fingerprinting.

The added advantages of positioning application deployed at the edge of the network (compared to conventional cellular approaches) are:

- The direct access to low-level signaling in real-time.
- Reduction of signaling overhead with the core of the network.

A key enabler of network-based positioning is cloud-RAN, which allows:

- Tight synchronization between eNBs instances running on the same processing frame.
- Data acquisition with lower latencies.

We evaluate the accuracy and response time of the RTPoT application on the top of an indoor cloudified LTE network and compare it with state-of-the-art solutions presented in the Microsoft competition [125]. Results have shown that RTPoT achieves highest positioning accuracy per deployment density compared to solutions leveraging existing infrastructure and without fingerprinting [22]. The contributions of this chapter are as follows:

- We propose RTPoT - a light-weight, scalable and accurate positioning and tracking system that exploits real-time radio measurements at the edge of radio networks. RTPoT relies on original ideas of filtering outliers and characterizing the radio transmission channels blindly, i.e., without any prior knowledge of transmitters' radio settings or indoor layout. RTPoT operates with zero-configuration and without any a prior knowledge of the deployment environment (c.f. Section 7.2).
- We built a cloudified LTE network with four eNBs connected to the same core network deployed in an indoor environment, and conduct over-the-air experiments with a static and moving UE to evaluate the performance of RTPoT under realistic settings. Results have shown the feasibility of an

accurate and low response time for positioning and tracking with RTPoT based on LTE (c.f. Sections 7.3 and 7.4).

7.2 Architecture

Recently, network-based positioning schemes have been actively researched. Many algorithms were developed to achieve network-based positioning with different accuracy, such as cell-ID with round trip time (RTT), TA or UTDoA [75]. It is shown that positioning techniques on the base of the network mainly appear to overcome many weak points of UE-based techniques:

- Minimize the impact on the UE manufacturer and supported protocols (UE agnostic).
- Take advantage of higher hardware reliability and accuracy for collecting radio measurements.
- Collect radio measurements in the correct direction; giving that downlink and uplink radio measurements are not always reciprocal [95].
- Allow access to fine-grained radio measurements, such as RSSI measurements at the subcarrier level.

In this section, we describe how our proposed solution RTPoT assures the quality of radio measurements and characterizes the area of interest precisely to achieve accurate positioning and tracking. The overall architecture of RTPoT as described in Figure 7.2 is composed from four main components as follows:

- The data acquisition component (1) is responsible to collect and correlate radio measurements from different eNBs. Note that estimating UE parameters, such as UE transmitted power, can only happen inside the serving eNB. Then, the estimating UE parameters will be shared with other eNBs for the correlating variables step.
- The outlier detection and mitigation component (2) aims to identify outliers in the correlated radio measurements and mitigate them before characterization of the channel with respect to the measurements. Furthermore, this component contains an aggregator, which reduces the rate of the filtered measurements.
- The channel characterization component (3) estimates the path loss exponent of the target environment using distance-power relationship. While power measurements are provided directly from component 2, the distance measurements can be estimated coarsely using a proximity-based algorithm.
- The localization component (4) uses the output measurements from component 2 and the channel parameters from component 3 to estimate accurately the UE location.

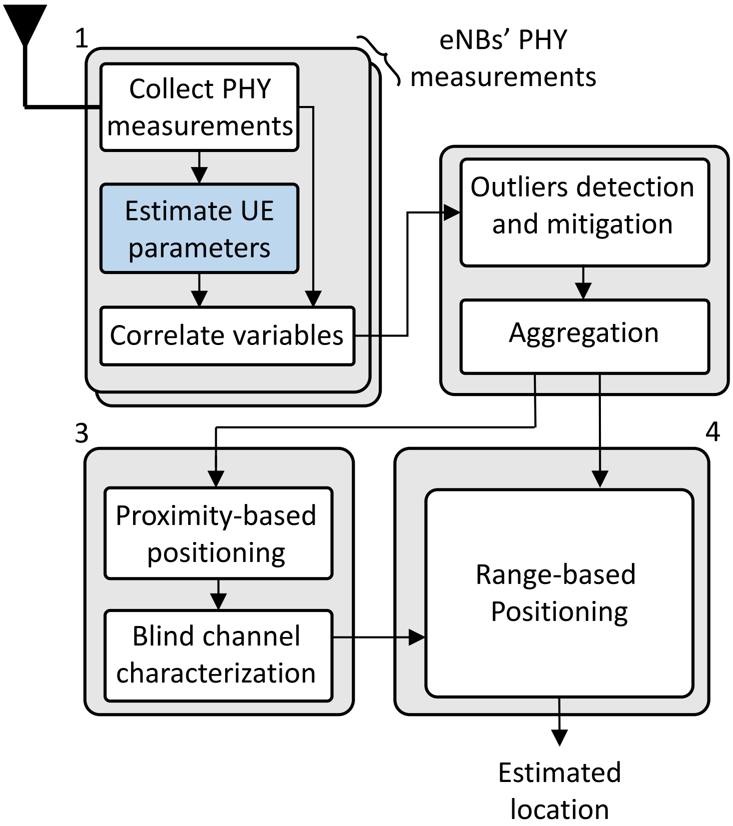


Figure 7.2: System architecture of RTPoT.

7.2.1 Algorithm

The proposed network-based positioning and tracking application is presented in Algorithm 1. The mobility mode (i.e. static or moving) and target positioning QoS (i.e. level of accuracy and response time) are the input parameters that depend on the application or service requirements. Then, the system initializes and adapts the algorithm operation accordingly. The main input of any algorithm is the acquisition time T, which has the requested accuracy, response time and mobility mode as inputs. T is determined based on statistical evaluations of each algorithm's response time and predicted accuracy. If the desired response time is short, the system outputs a coarse-grained estimation of users' locations without executing the whole algorithm, which in turn decreases the required processing time. The following subsections provide more details about each step of Algorithm 1.

RTPoT makes use of power measurements, such as RSRP and RSSI, as well as time measurements, such as the PRACH TA and the PUSCH TA updates, separately. Quality measurements such as RSRQ and CQI will be considered for extended work.

Algorithm 1 Positioning and Tracking

```
Inputs:
mobility mode // static or moving
positioning QoS // accuracy, response time
procedure RTPoT
    $T$ = get_time (mobility mode, QoS) // acquisition time
    while experiment is active do
        $X_n$ = radio_measurements_acquisition ($T$)
        $\widehat{X}_n$ = outlier_detection_mitigation ($X_n$)
        if mode is static then
            [$\alpha$, $\boldsymbol{P}$] = channel_characterization ($\widehat{X}_n$)
            $P$ = range_based_positioning ($\widehat{X}_n$, $\alpha$)
        else if mode is moving then
            $Y_n^t$ = track ($\widehat{X}_n$)
            [$\alpha^t$, $\boldsymbol{P^t}$] = channel_characterization ($Y_n^t$)
            $P^t$ = range_based_positioning ($Y_n^t$, $\alpha^t$)
            $\widehat{P}^t$ = track ($P^t$)
        end if
```

To ensure the quality of our radio measurements and remove outliers, we used our proposed outlier detection and mitigation algorithm described in Section 4.6.1. A summary of the proposed algorithm is presented in Algorithm 2. If the desired response time (c.f. Algorithm 1) is short or the number of simultaneously tracked UEs is large, the filtering step can be skipped, i.e., passing X_n directly to the positioning algorithm. However, this causes outlying measurements to be included in the next steps and degrades the localization accuracy.

7.2.2 Blind Channel Characterization

Blind channel characterization means to characterize radio channels without any prior knowledge of transmitters' radio setting or indoor layout. Using proximity-based algorithms, an initial position of the target UE can be obtained. Proximity algorithms, such as Combined Differential RSSI (CDRSS) [18], only use the decaying principle of RSSI values with increasing distance without enrolling any channel modeling to express exactly the amount of decaying. The target UE position $(x_{\text{est}}, y_{\text{est}})$ is a linear combination of at least three sensor nodes' positions (c.f. Equation 2.3). Sensor's weights w_i in CDRSS are calculated based on differential RSSI between sensor nodes and hence are transparent to the UE transmitted power and manufacturers [18]. Then, the estimated distance d_i (in log format) between UE and each sensor with corresponding RSSI measurements are paired, i.e., $Z_n = [X_n, d_n]$. Later, we select Z_i pairs that fit most to the linear model and exclude all outlying pairs caused by outlying RSSI measurements or by the proximity algorithm. Finally, the PLE is calculated as the slope of the nominated pairs. If the requested positioning accuracy is low, the channel characterization step can be skipped. The coarse-grained estimated position P can directly be used.

Algorithm 2 Outlier Filtering

```
input
Create $X_n$ of $T$ = 30 sec
procedure FILTER
    calculate $E_n(X_n)$
    for all RSSI$_i$ in $X_n$ do
        calculate $h_{ii}$ and $D_n^i(\widehat{\epsilon})$
        calculate S$_i$
        if S$_i > \varpi$ then
            Substitute(RSSI$_i$)
        else
            Model(RSSI$_i$)
        end if
    end for
    return $\widehat{X}_n$
procedure MODEL
    RSSI$_i \rightarrow X_{2M}$
    $g($RSSI$_i) = $ Pr$(x = $ RSSI$_i)$ $\forall\, x \in X_{2M}$
procedure SUBSTITUTE
    RSSI$_i \leftarrow$ RSSI$_{\text{filter}} = \underset{x \in X_{\text{normal}}}{\arg\max}\, g(x)$
```

In multivariate settings, outliers cannot always be detected by directly applying univariate outlier detection to each variable separately. Various methods for detecting multivariate outliers have been proposed, such as the MCD algorithm [76, 152]. Covariance **S** of non-outlying measurements is a measure of data spread between variables. For p = 2, **S** is expressed in Equation 7.1.

Algorithm 3 Blind Channel Charechterization

```
input
$\widehat{X}_n$
procedure ESTIMATE
    $d_n \leftarrow P_n$ = CDRSS ($\widehat{X}_n$)
    $Z_n$ = $[\widehat{X}_n, d_n]$
    $C$ = MCD ($Z_n$)
    for all $\boldsymbol{z_i}$ in $Z_n$ do
        $d(i)$ = Mahalanobis ($\boldsymbol{z_i}$, $C$)
        if $d(i) \leq \sqrt{\chi^2_{2,0.975}}$ then
            $S \leftarrow \boldsymbol{z_i}$
        end if
    end for
    [$\boldsymbol{\lambda}$, $\boldsymbol{V}$] = eigen_decomposition (S)
    $\alpha = -$max$(V)/$min$(V)$ | $max(\boldsymbol{\lambda})$
    return $\alpha$
```

$$\boldsymbol{S} = cov(X) = \frac{\sum\limits_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{(n-1)} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \tag{7.1}$$

$\overline{X}$ and $\overline{Y}$ are the mean value of X and Y variables. $\boldsymbol{S}$ is a symmetric matrix and can be decomposed into 2 orthogonal eigenvectors **V** and 2 eigenvalues $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda S} = \boldsymbol{\lambda V} \tag{7.2}$$

The MCD tolerance ellipse has one large axis called the major-axis and a smaller one called the minor-axis. The slope of the major axis, which represents (minus) the PLE α value, equals the ratio of the eigenvector of the largest eigenvalue. The slope of the minor axis equals ratio of the eigenvector of the second largest eigenvalue.

7.2.3 Positioning and Tracking Algorithms

Since RSSI measurements ideally are expected to be constant for a static UE, the filtering process uses the majority of the analysis to detect unexpected changes and treat them as outliers. In static scenarios, we considered ranged-based algorithms similar to our approach in Sections 2.9.5 and 3.4. However, RSSI changes in moving scenarios will be interpreted as outliers and will be filtered out. These outliers can degrade the tracking performance of a moving UE.

To obtain better positioning results, the proposed solution as expressed in Algorithm 4 is based on the Kalman filter (KF) algorithm. The proposed tracking solution starts by creating discrete samples X_n from RSSI measurements with constant periods of time T. The Huber's amplitude estimation has been also used here using X_n to provide an input parameter to the Kalman filter. KFs have a recursive operation and involves in two steps, a prediction step and a correction step [181]. The first step uses previous states to predict the current state. The second step uses the current measurement, such as RSSI or UE location to correct the predicted state. In the proposed solution, one-dimensional KF is applied on RSSI measurements first and later on the estimated UE positions. Finally, the UE speed is calculated using the final KF position output given the equation in Algorithm 4 Line 16. $d^{t_{i+1}} - d^{t_i}$ is the difference between two consecutive UE positions and $t_{i+1} - t_i$ is the time difference at which the corresponding positions were measured.

7.3 Experimental Setup

An indoor cloudified LTE network is built based on the OAI hardware and software platforms to validate the proposed concept in RTPoT [140]. Figure 7.3 illustrates the network setup that consists of one UE and 4 OAI eNBs connected to one OAI EPC and one OAI home subscriber server (HSS). eNBs are configured to operate at 5 MHz in band 7 with FDD mode and *one single antenna*.

Algorithm 4 UE Tracking

```
procedure Track
    while Tracking is active do
        Create $X_n^t$ of $T$ = 1 sec
        $\widehat{X}_n^t \leftarrow$ Algorithm 1 // $RSSI_{filter}$
        $Y_n^t$ = KalmanFilter($X_n^t$) // Track RSSI
        $d^t$ = Equation 2.4 $\leftarrow Y^t$ // substitute $Y^t$ in Equation 2.4 and store the result in $d^t$.
        $P^t$ = KalmanFilter($d^t$) // Track UE Position
        $V^t$ = SpeedEstimate($\boldsymbol{d}$)
procedure KalmanFilter
    Initialize the Kalman filter // RSSI or position
    for each measurement $\in I$ do // $I$ is the input
        $O^i$ = Prediction($P^{i-1}$)
        $P^i$ = Correction($I^i, O^i$)
    end for
procedure SpeedEstimate
    $V^{t_i} = \frac{d^{t_{i+1}} - d^{t_i}}{t_{i+1} - t_i}$
```

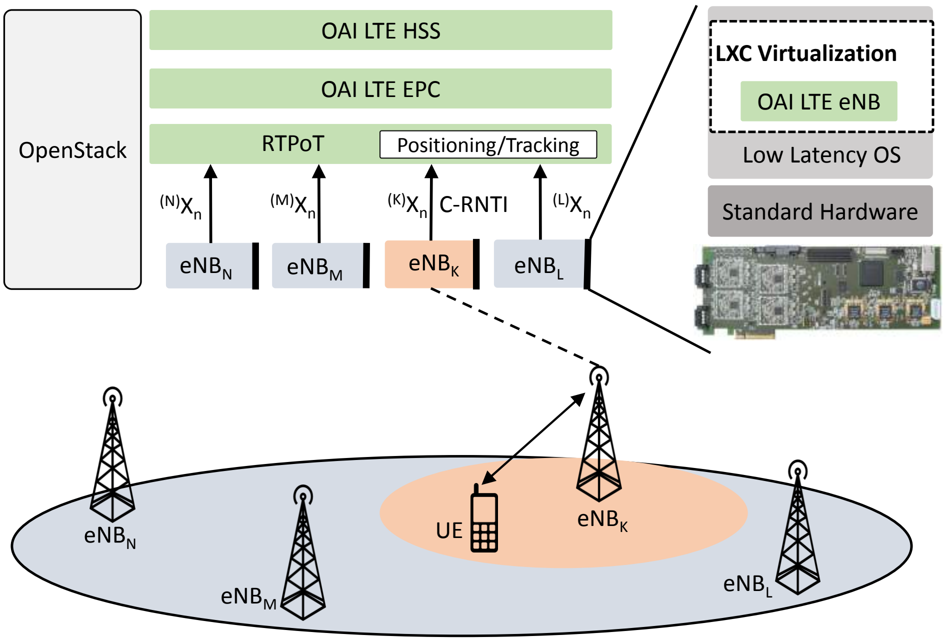


Figure 7.3: Experimental setup

LTE: As a continuation of our work in Chapter 6, we consider the OpenAir-Interface software implementation of LTE. To build a cloudified LTE network (as an initial step towards real-time network-based positioning), the OAI eNB, EPC, and HSS was ported into LXC environment (by Eurecom [72]) as the virtualization

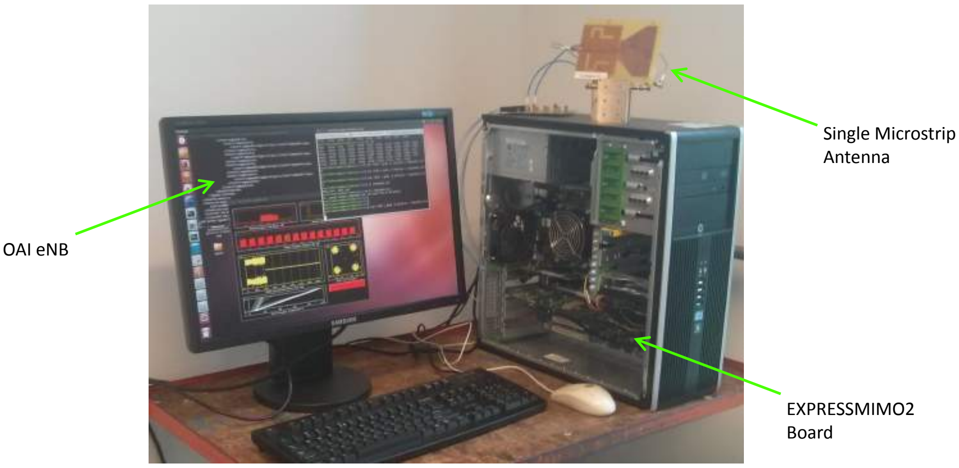


Figure 7.4: eNB setup with single Antenna

technology [24]. All the machines (hosts or guests) operate on Ubuntu 14.04 with the low-latency Linux kernel version 3.17 with x86-64 architecture. In the considered setup, each eNB is running on a separate PC interconnected with the EPC through the S1 interface. Furthermore, each eNB streams the radio measurement information to the RTPoT application located at the same PC running the EPC.

Radio unit: For real-world experimentation and validation, the default radio frequency frontend for OAI is the ExpressMIMO2 board. As shown in c.f. Figure 7.4, the ExpressMIMO2 board is connected to the PC running eNB through PCI Express (PCIe) serial interface *without* any fronthaul.

UE: For a user equipment, a laptop equipped with one USB LTE Dongle Huawei E398u-1 was used. To maintain the UE in an active mode, eNB is configured to periodically schedule an uplink transmission from the UE every 100 ms.

Online Measurements: We built several wrappers inside the PHY and MAC layers of OAI eNB implementation that collects a set of information on per subframe-basis and stream it to the RTPoT application for further processing. For power measurements, we collected wideband RSSI and RSSI per each active subcarrier. For time measurements, we collected PRACH TA and PUSCH TA updates. For quality measurements, we obtained wideband CQI and subband CQI (13 measurements for 5 MHz channel bandwidth). All aforementioned measurements were tagged with eNB ID, carrier component ID, LTE frame number, and LTE subframe number. Another wrapper was built inside the RRC of OAI eNB implementation that collects UE measurement reports (RSRP and RSRQ) every 100 ms.

The current implementation of OAI eNB neither supports the X2 interface nor soft handover between eNBs. To avoid the problem of overhearing UE's uplink measurements at different eNBs, we performed hard-handover by switching off one eNB and switching on another. The handover latency ranges from 30s to 1min. This approach does not affect the performance of RTPoT. However, it is limited to localize static UEs in 2D, but it will not be possible for tracking applications.

RTPoT: For the purpose of online data acquisition and processing, we developed RTPoT using Matlab. Matlab includes a rich set of statistical tools, which enables us to experiment various positioning and tracking approaches and optimize RTPoT in near real-time processing. RTPoT uses an ongoing data-handling routine, based on Matlab timer, which collects and merges different streams over a local network in a synchronized container and prepare them for joint processing.

7.4 Measurement Results

To validate the proposed solutions of online positioning and tracking, we conducted several experiments using a real cloud-based LTE network over a floor as shown in Figure 7.9. All experiments were conducted on the floor of the mobile communications group, Eurecom, France.

7.4.1 Channel Characterization

A closer look at the power distribution among different subcarriers versus distance is illustrated in Figure 7.5. In a NLOS environment, we measured RSSI of each allocated subcarrier at an eNB with respect to distance. The decaying behavior of RSSI with distance is observed. However, we see fluctuations in RSSI between subcarriers at the same distance. Moreover, deep fading points, up to -30 dB lower than average RSSI measurements, are happening more often at further distances. As a compromising solution between estimators' efficiency and computational overhead, we implemented the median estimator (instead of mean, which is very sensitive to outliers) to estimate the RSSI of a subframe. All further experiments are using the median estimator of active subcarriers to report RSSI measurements at eNB RRH. However, the UE makes use of the mean estimator to calculate RSRP measurements over the allocated bandwidth. Recall that the LTE scheduler handles allocating N subcarriers for active UEs' uplink transmission every TTI. In Figure 7.5, we only selected subframes with 36 active subcarriers for illustration purposes.

Nevertheless, using the log-normal shadowing model expressed by Equation 2.4, it is hard to characterize radio channels using only RSSI measurements. This challenge is because of power control algorithms operating at the eNB to sustain a

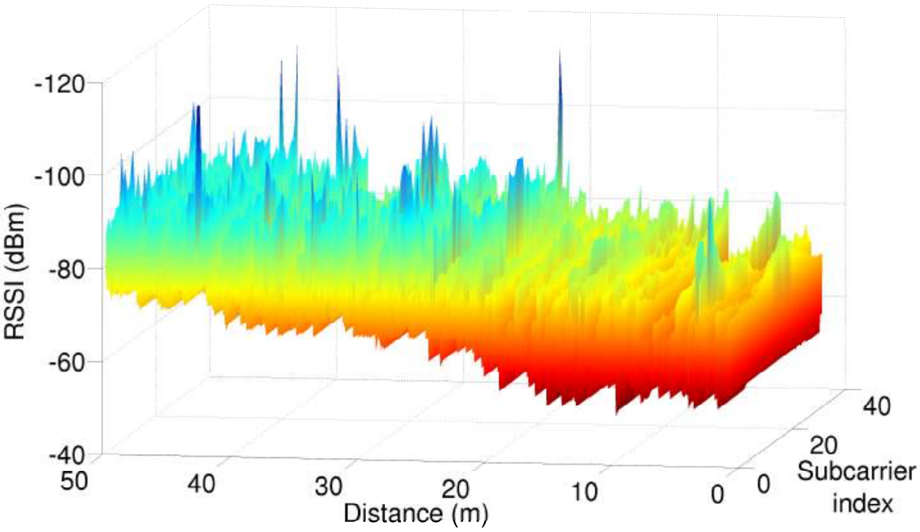


Figure 7.5: Fading in subcarriers.

consistently received signal strength at the eNB RRH and/or at the UE terminal. Hence, to obtain a correct mapping between received power and its corresponding distance, we correlate UE estimated P_{Tx} and received RSSI measurements inside the serving eNB. Similarly, we correlate eNB transmitted power with UE received power.

Power-based approach: Results illustrated in Figure 7.6-a and 7.6-b show the correlation between UE transmitted power and eNB received power with respect to distance in a NLOS scenario. The power control algorithm can adjust the UE transmitted power within the UE manufacturer specifications. Up to a distance of around 25m, the UE increases its transmitted power until it reaches the maximum transmit power to satisfy the required received power at eNB. As the UE gets farther from eNB; the UE transmitted power stays constant while the eNB received power starts decreasing. Given a predefined PLE measured in an offline phase, we estimate the UE position using Equation 2.4 as illustrated in Figure 7.6-c. Note that the predefined PLE value is used only in this experiment to study the distance power relationship. All further localization experiments rely on our proposed algorithm to estimate PLE from online measurements.

Time-based approach: As mentioned in Section 2.4.1, the PRACH timestamp is a promising radio measure for positioning in LTE due to its high resolution. However, the LTE PRACH TA resolution does not meet the demands for indoor or dense urban area deployments, where the distance between two eNBs can then be less than one LTE TA value (78.12m). Hence, the measured TA will be constant regardless of the actual location of the UE. Hence, the results presented here con-

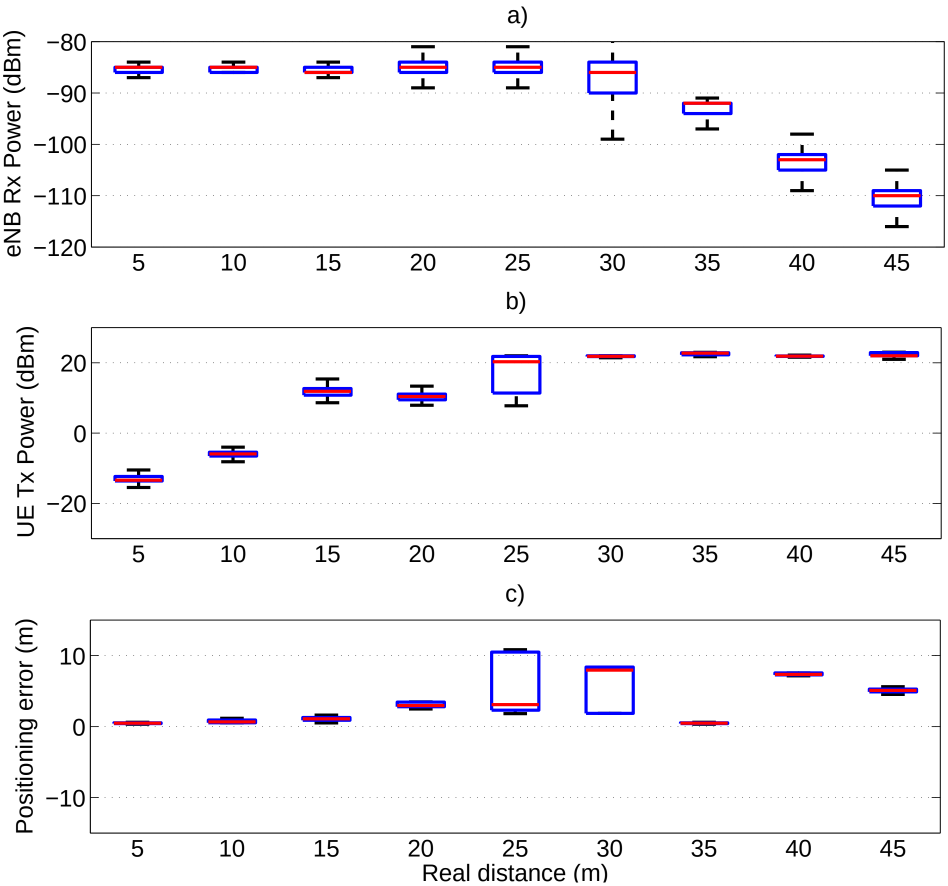


Figure 7.6: NLOS power characterization.

sider only the PUSCH TA for UE positioning. Results illustrated in Figure 7.7-a and 7.7-b show the PRACH TA and the PUSCH TA update with respect to distance in a NLOS environment. The PUSCH TA update shows a definite increase with the distance. The base value of both TA measurements is due to calibration settings that have to be considered inside OAI. Figure 7.7-c shows the corresponding positioning error for PUSCH TA update giving that 1 TA at 5 MHz = $4T_s$. It can be seen from Figure 7.7 that the indoor time-based algorithm provides a good positioning estimate, however, less accurate than power-based algorithms. Time-based positioning approaches are not so good for indoor environments due to harsh multipath environments as power-based approaches[79]. To further improve the positioning accuracy, hybrid-metric positioning algorithms, which use time and power measurements simultaneously, are considered for extended work. Hence, the following experiments will only consider power measurements.

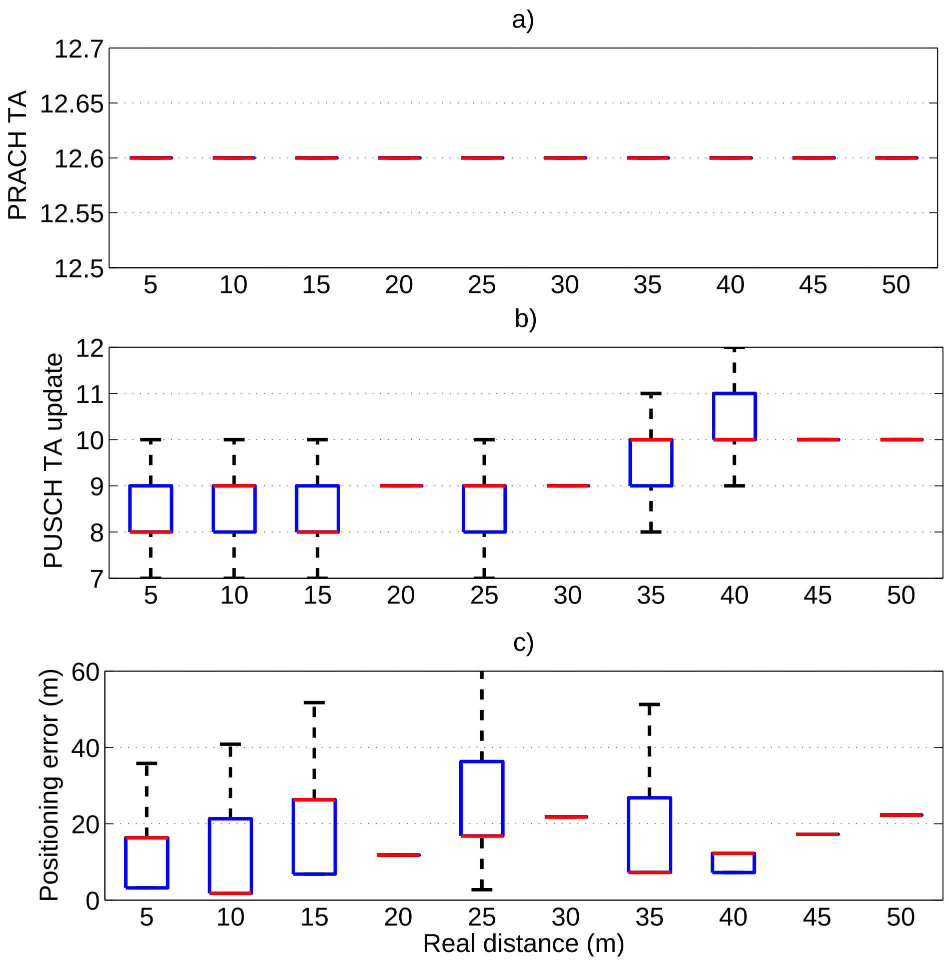


Figure 7.7: NLOS timing advance measurements.

7.4.2 Static Positioning

We conducted experiments using 4 static eNBs and 1 UE at 7 different locations. Two performance parameters were considered for stream processing: data acquisition time and RTPOT processing time. In Algorithm 2, we acquire data into sets of 30 sec. The UE was considered as static during the acquisition time, whereas each experiment spans a duration of 5 min. The filtered dataset will then be divided into subsets of 1 sec duration. The median value of each subset will be sent separately to the proximity positioning algorithm to get an initial estimate of the UE position. All paired distance and RSSI points (120 pairs = 1 UE * 4 eNBs * 30 sec) will be used in Algorithm 3 for robust PLE estimation as shown in Figure 7.8. At each UE location and over each subset measurements, we estimated the PLE based on two approaches: (i) the linear curve fitting described in [65] and (ii) the MCD approach described in Algorithm 3. In Figure 7.8, the result of (i) is the linear regression line and the result of (ii) is the MCD estimation. Recall that the MCD tolerance ellipse and its major access (the MCD estimation) is calculated based on our discussion in Section 7.2.2. In Figure 7.8, measurements outside the tolerance ellipse are considered as outliers. The resulting PLE values will be used in Equation 2.4 for final UE estimated position. The final positioning results are expressed in Figures 7.9 and 7.10. Even though various metrics have been proposed in the literature (i.e., average location error, RMSE, 95th percentile and so on.), we believe there are many more to be considered, such as the deployment density of sensors and the acquisition duration. We propose the positioning efficiency, defined as the absolute positioning error times deployment density of sensors, as a benchmark to compare the performance of RTPoT with other solutions in literature [125]. We define the deployment density as the number of sensor nodes deployed per 100 square meters (m^2) of the evaluation area. The evaluation area is considered bounded by sensors at the edges, i.e., in our setup, the trapezoid area connecting the vertices sensors is 451 m^2, which leads to 0.89 sensors/100 m^2.

Figure 7.10 shows that a static UE has a variation in its estimated position up to 2 meters in locations 1, 6 and 7 (defined by the boxplot upper and lower whiskers). It is interesting to note that all these points tend to be located at edges of the evaluation area, where it is most challenging for the proximity algorithm to estimate accurately the UE position [18]. Depending on the positioning application and the required acquisition time, additional post-processing algorithms can be used at highly variant points to improve the positioning median and reduce the positioning variation, e.g., MCD algorithm with x and y coordinates as input variables. Table 7.1 contains the positioning efficiency of our proposed solution when compared with the best four systems of [125]-Table 2. RTPoT does not use any additional positioning hardware and without any training phase, hence, we compare RTPoT

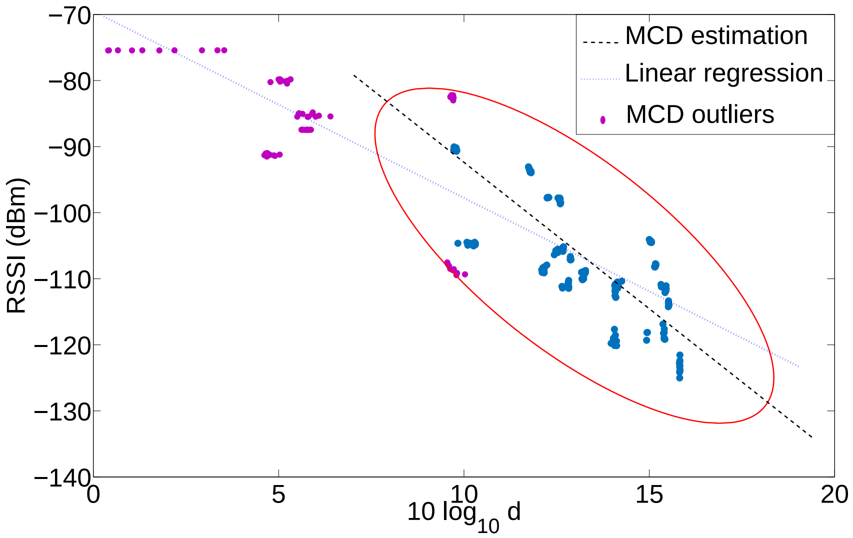


Figure 7.8: Robust estimation of PLE using MCD.

with infrastructure free and modified rooms solutions [125]. Our experimental setup was bounded to the number of eNBs and their location. The comparison in Table 7.1 considers linear relationship between the number of ANs and the positioning error. RTPoT has the highest positioning efficiency by just leveraging existing LTE radio measurements without any additional deployment overhead.

Figure 7.11 shows the performance of the proposed outlier detection and mitigation procedure expressed in Algorithm 2. It is clear that Algorithm 2 using the Huber estimator enhances the positioning performance better than using other estimators, such as the mean and the MLE estimators. This performance is due to better statistical characteristics of the Huber estimator in terms of robustness to outliers and efficiency compared to the mean and the median. The filtering threshold ϖ was set to 2 because of the achieved high performance after extensive measurements covering a broad range of ϖ.

To validate the proposed solution of channel modeling, we compared the proposed solution using the MCD algorithm for PLE estimation with the following.

- Constant PLE = 2 is considered as a baseline approach [128]. It considers indoor environments, such as free-space environments, without obstructions.
- Linear curve fitting approach [65], which estimate the best line fit that minimizes the mean square error of all distance-RSSI pairs (120 pairs).
- Proximity-based algorithm. More specifically, we used the CDRSS algorithm exploited in Algorithm 3.

Results as shown in Figure 7.12 reveal that RTPoT using Algorithm 3 obtains the least positioning error among different 7 locations with 2.7m compared to 4.3m for the linear curve fitting approach, 5.6m for constant PLE and 6.6m for the proximity-based solution. Our results can be easily exported by UE-based measurements, i.e., RSRP. RTPoT using RSRP in Algorithm 3 obtains 3.3m positioning error. These findings mean that the proposed algorithm provides a more

Table 7.1: Positioning performance comparison.

System	Density (sensor/100m^2)	Positioning performance error (m)	efficiency (sensor/100m)
RTPoT	**4 / 4.5**	**2.7**	**2.4**
Klepal et al.	10 / 2.2	1.67	7.5
Laoudias et al.	10 / 2.2	1.92	8.7
Zou et al.	10 / 2.2	2.69	12.2
Ferraz et al.	10 / 2.2	2.91	13.2

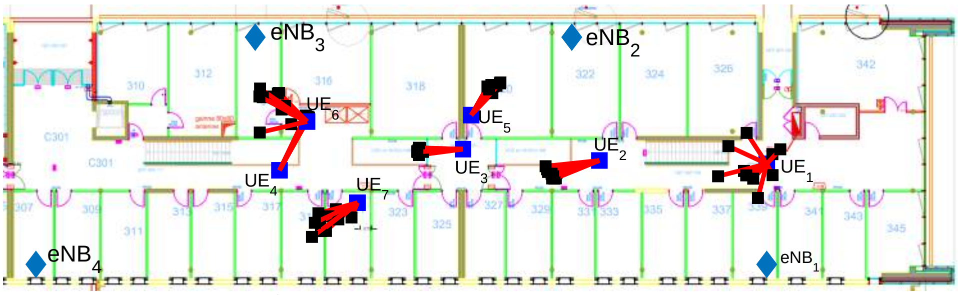


Figure 7.9: Positioning experiment setup.

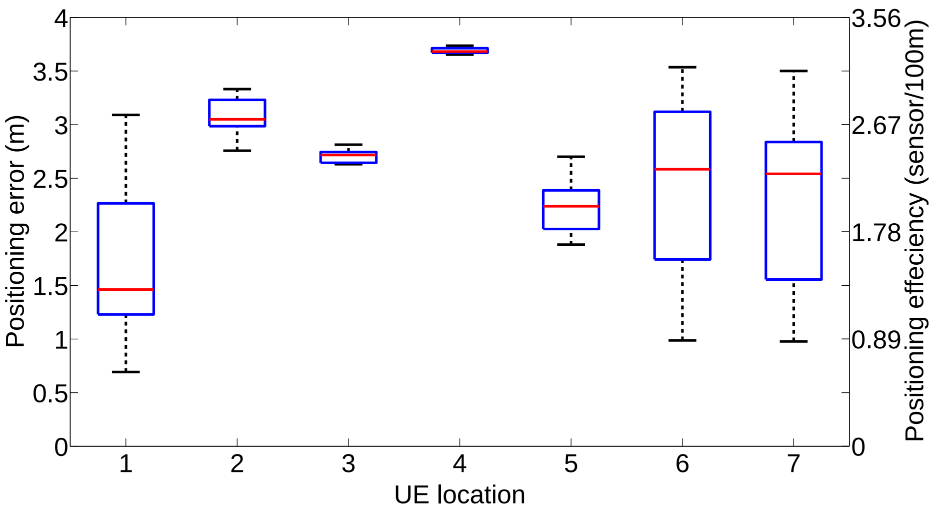


Figure 7.10: Positioning experiment result with 30 sec aggregation.

accurate estimation of PLE, which improves positioning performance, especially for network-based measurements. However, we do not see much gain from the linear curve fitting solution compared to the proximity-based solution. The lack of gain is because the linear curve fitting solution considers all distance-power pairs for calculation. Hence, if the proximity-based solution diverges, the linear curve fitting solution will also diverge. Finally, we benchmarked the processing time of the proposed solution built in Matlab. Results illustrated in Figure 7.13 show the time distribution between different entities to process one dataset of 30 sec. Most processing time is consumed inside the filtering module (1.2 sec) and initial configuration and intermediate setups (0.8 sec). Channel modeling and positioning require only 0.4 sec. Multiple techniques are available to speed up the simulation, such as deploying Matlab code as executable (called MEX-files) [131]. While a Matlab code is interpreted during runtime, MEX-files are first compiled into the computer's native language, which yields better performance.

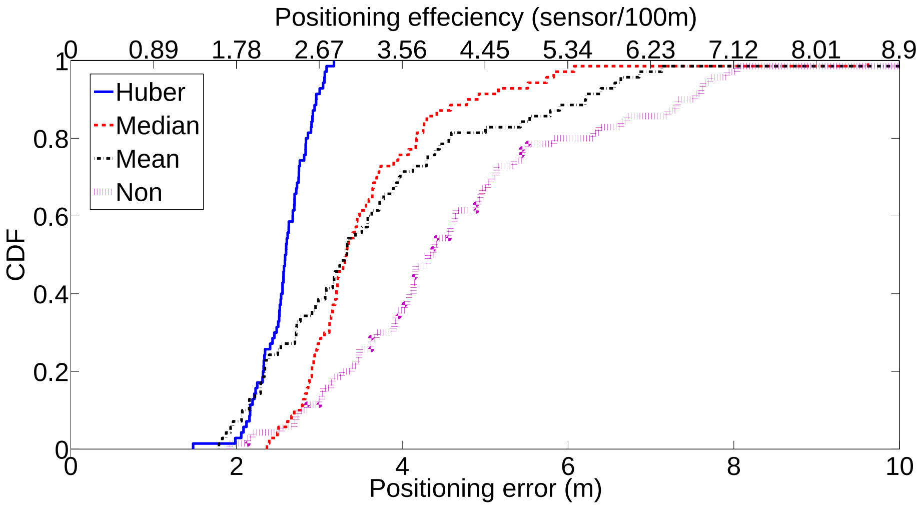


Figure 7.11: Prosed outlier mitigation algorithm using different estimators.

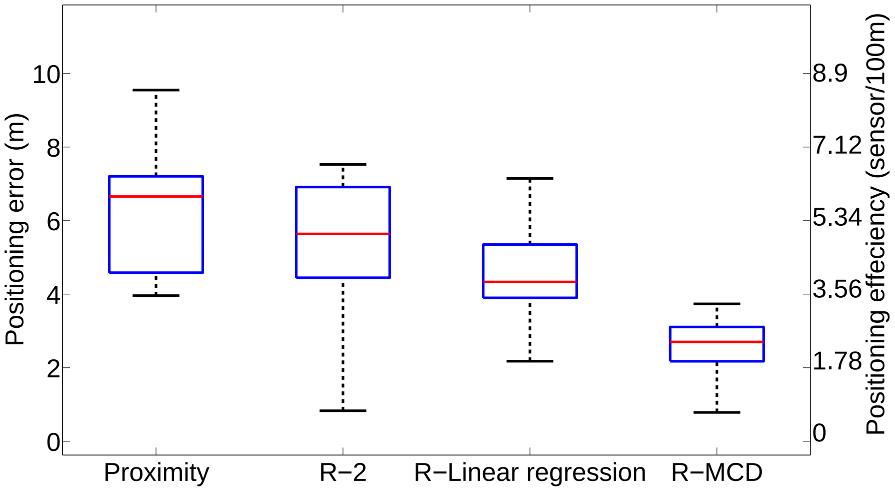


Figure 7.12: Positioning experiment setup.

7.4.3 Mobile Tracking

Tracking UE devices has the high importance of application and business models as static UEs. We conducted several walking experiment using one eNB and one UE to validate the proposed solution in Section 7.2.3. The UE moved in a straight line away from the eNB in different experimental settings: LOS and NLOS, relatively slow (0.8 m/s) and relatively fast (1.6 m/s) movements. The results illustrated in Figures 7.14 are examples of relatively slow walking (0.8 m/s) in NLOS

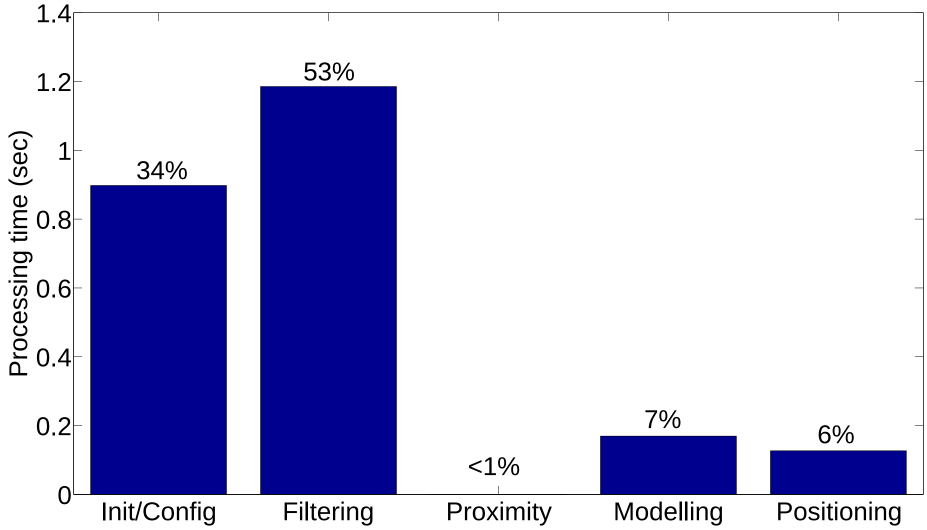


Figure 7.13: Matlab Processing time of total 2.4 sec.

conditions. We make use of RSRP measurements reported inside the RRC to exhibit that RTPoT works with the various type of measurements. Similar results were obtained using network-based measurements, i.e., RSSI correlated with estimated UE P_{Tx} measurements. In Figure 7.14-a, we see RSRP measurements with respect to distance, given that the eNB has a constant transmission power of 0 dBm. The general trend observed in Figure 7.14 is that RSRP measurements decay with the distance. Fast fading and slow fading between distance 40-80m are clearly seen in Figure 7.14-a. First, we apply Algorithm 4 using the collected RSRP measurements. In Figure 7.14-a, the KF output, the purple line, smooths RSRP measurements from most fast fading. However, looking at the slow fading region, the KF did not overcome it completely, but smoothing it. This shortage is because of the initializing parameters of the KF, which optimizes the performance of smoothing level, the number of measurements, and so on. Figure 7.14-b shows the estimated UE positioning (dashed black line) using the RSRP KF output with 7.2m positioning error. The purple line represents the KF positioning output, which is considered as our final estimation of the UE position with 6.2m average positioning error. Moreover, for mobility prediction applications and services, we estimate the UE moving speed. Similar to our positioning approach, we compared the estimated velocity after applying KF on RSRP measurements only and after applying KF on RSRP measurements and positioning estimates. The latter approach achieved higher accuracy for speed estimation with 0.2 m/s average speed error and less variation compared to the former approach with 1.3 m/s average speed error. As mentioned in Section 7.2.3, the computational efficiency of Kalman filter is high. Our Matlab implementation of the Kalman filter introduced an additional latency of 0.4 sec for each 30 sec tracking.

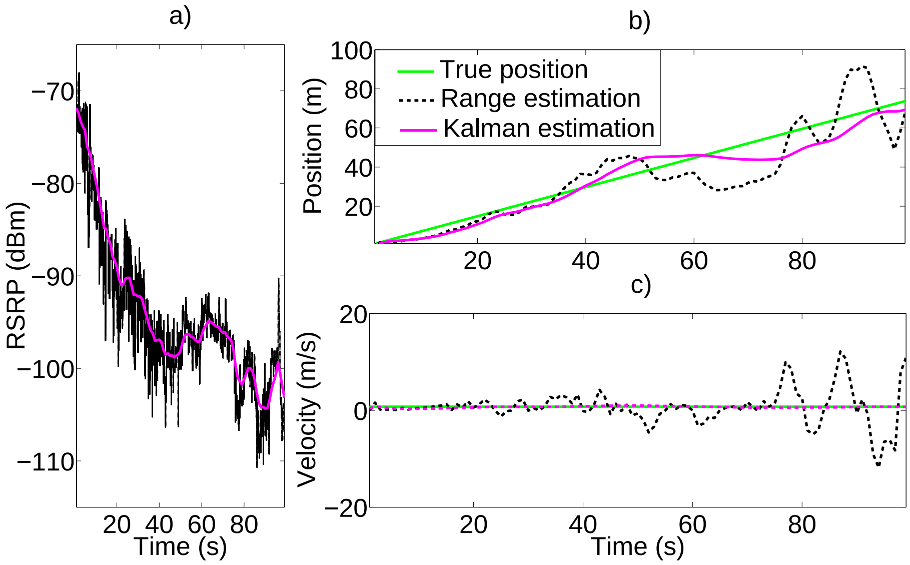


Figure 7.14: Tracking experiment using double Kalman filters.

7.5 Discussion and Remarks

We presented RTPoT as a network-based positioning and tracking solution compliant with LTE/LTE-A. The evaluation showed promising results for a network-based positioning and its applicability to an UE-assisted positioning and environment agnostic scenarios. Going forward, we would like to elaborate further in the following directions:

Towards pervasive RTPoT: Mobile Edge Computing is complementary to and supportive of both software-defined networking (SDN) and network function virtualization (NFV) concepts, which play an important role for configuring, managing, and optimizing network resources. An added benefit of the RTPoT solution is that it acts as a passive MEC application at the network edge, and hence, it operates with any cellular network architecture. As long as the required real-time measurement information is available, RTPoT operates transparently with multiple radio access technologies. Hence, deployment, optimization, and further updates are minimized to a software upgrade without replacing any installed equipment or adding location-measurement-unites to the distributed RRHs. Thus, RTPoT can meet the cost demands for any service provider. Moreover, RTPoT does not require any additional signaling over the Air interface or updates in the network protocol stack. Thus, the deployment overhead will be minimum.

Positioning accuracy: RTPoT makes use of radio measurements collected at the UE or eNB side. eNB measurements are potentially higher quality and with higher granularity than UE measurements (c.f. Section 7.2). However, the high number of measurements comes at the expense of RTPoT positioning latency. The quality of eNB radio measurements can be further improved by operating the eNB at higher channel bandwidth, which reduces the impact of multipath fading [79]. RTPoT, as well as any other positioning algorithm, can be improved by increasing the deployment density of the development area. However, a realistic density is required to achieve a good compromise between the deployment cost and QoS.

Multi-user Scenario: An excellent use case for cloudified RAN is to enable direct access to DCI allocations between BBUs of nearby geographical RRHs (assume BBUs of nearby eNBs are operating within the same computing infrastructure). To be specific, the high bandwidth connection between the host running the virtual switch (or shared memory) and VMs running the C-RAN is much faster than any physical X2-interface. This connection is illustrated in Figure 6.1 and acts as X2-like interface, as we now call it [23]. The synchronous nature of cloudified RAN brings some gains for LTE X2-like interface, such as reduced signaling overhead over X2-interface and instantaneous access to DCI information of different BBUs. One use-case envisioned by the X2-like interface is simultaneous multi-user positioning through various inter-cell interference coordination techniques [145]. However, neither the basic X2-interface nor the X2-like interface is currently implemented in the OAI eNB implementation. The applicability of these scenarios is left open for future work.

Coordinated multipoint: When an UE is moving towards the edge of the cell, CoMP benefits from multiple receptions to combine UE radio signals coherently or noncoherently with various eNBs to create a consistent user experience across the entire serving area in downlink and uplink. CoMP requires that all UEs' transmissions in a cell arrive at the eNB within the cyclic prefix of an SC-FDMA symbol. In the case of extended cyclic prefix (= 16.67µ), the maximum inter-eNB distance can be up to 5 km. RTPoT and CoMP can play together a complementary role. On the one side, CoMP provides RTPoT with UEs' radio measurements at different eNBs. On the other side, RTPoT allocates and steers radio resources based on UEs positioning and mobility prediction information. Similar to the Multi-user scenario, Coordinated multipoint applications require very fast implementation of X2/X2-like interfaces.

7.6 Conclusions

Many location-based services and applications require a comparable positioning performance in different environments. In areas where GPS fails to provide an accurate positioning and tracking performance, several solutions have been proposed in the literature that heavily rely on the use of specialized hardware or fingerprinting to increase the positioning accuracy. The former are usually expensive and inefficient for medium to large scale deployments, and the latter is time-space dependent and requires massive processing making it not applicable for real-time scenarios. To obtain a cost-efficient and accurate positioning solution, RTPoT contains a novel approach to improve the quality of radio measurements and to estimate and model radio transmission channels blindly while being transparent towards end-users. By removing the environment dependency, RTPoT supports consistent and ubiquitous positioning services over the large-scale deployment of cellular networks. RTPoT is applied to LTE/LTE-A as the best enabler to tighten all location services together. In terms of accuracy and processing time, RTPoT proved to be highly accurate with up to 2.7m average positioning error at 0.89 sensors/100 m^2 and computationally-efficient with 2.4 seconds as a processing latency.

Chapter 8

Experiences and Lessons Learned

Based on our experience from the past four years running research on indoor localization, we have gathered a set of lessons learned and guidelines, which could allow better understanding of the indoor localization challenge. This research study allowed us to observe closely and evaluate multiple localization systems deploying various technologies in different environments. Even though we did not cover every single challenge in localization, we believe that the fundamental basics of the problem is clear.

8.1 Performance Measures

Propagation of indoor radio signals is affected by the environmental layout, such as the number of walls, floors, furniture density and mobility. Expressing irregular environments with regular propagation models will not give accurate match between theoretical and practical results. Models that rely on the exact knowledge of the environment, the specification of target devices or bring a customized formula for a particular unique environment are not attractive for wide deployments. To overcome these challenges, we proposed the following approaches:

- For passive localization systems without any collaboration with network operators or end users, we proposed a set of proximity-based localization algorithms, which work agnostic to the environment layout, target devices' radio settings and show reliable performance in the range of 3m in different indoor environments.

- To overcome the lack of knowledge about the indoor layout, we proposed robust algorithms to filter out contaminated signals by the indoor environment and improve the quality of radio measurements before being fed to the localization algorithm. Evaluation results show an improvement up to 75% at some locations with lower-quality of radio measurements.

- To improve further the localization accuracy for devices with multiple active radio interfaces, we proposed probabilistic algorithms that combine radio measurements before or after the localization algorithms. Experimental results show an improvement in the localization error down to 1.7m.

- For active localization systems, such as LTE network-based system, we proposed novel iterative algorithms to model the indoor environment using online measurements from target devices. The proposed solution shows a reliable performance of 2.7m with a deployment density of 0.89 eNB/100m^2.

We discuss here three main metrics to evaluate our proposed solutions and define a set of limiting factors in our target scenarios:

Response Times: real-time operations are quite important especially in dynamic environments, such as in hospitals. The response time from the user perspective includes latencies from data acquisition and localization process. In case of passive WiFi-based solutions and if the target device is in active mode, e.g., browsing a video file, data acquisition can be as small as a fraction of a second. However, for passive GSM-based solutions, data acquisition might be a limiting factor. In this system, we rely on users activities on communication scenarios that contain a message identity. Hybrid passive solutions that combine multiple radio technologies or multiple localization algorithms tend to perform better than non-hybrid solutions. However, the complexity of the system and subsequently added additional latency for data acquisitions. In case of active LTE network-based localization, RTPoT operates in two modes: coarse and accurate estimation modes. The coarse estimation mode operates when the system is based on proximity algorithms. In this mode, data acquisition and localization processing latencies are very short (a fraction of a second). However, the accurate estimation mode requires online training and data acquisition up to one minute for static users and one sec for mobile users.

Accuracy: due to the irregular nature of indoor environments and the lack of an accurate and unique model, which represents all indoor layout at one, localization algorithms have a trade-off between accuracy and response time. The more radio measurements we collect and the more sophisticated algorithms we operate, the better accuracy we get. Filtering algorithms that can improve the localization accuracy rely mainly on collecting more radio measurements to detect unexpected behavior in the environment and eliminate its impact on the localization output. In case of active WiFi devices or connected LTE UEs, performing filtering might introduce an additional delay. However, due to the high packet rate in both systems, the overall latency is much lower than in case of GSM-based solutions with low packet rate.

Deployment and calibration costs: unlike other proposed localization systems, our research targets realistic and cost-efficient scenarios that require a minimum amount of deployment and calibration costs. Our proposed solutions work with zero configuration and calibration. The only inputs to the localization module are the coordinates of anchor nodes and RSSI measurements.

8.2 System Costs

Most indoor localization solutions tend to deploy the target system within the same building or even the same floor. Most efficient localization systems are those which use available infrastructure such as WiFi APs and home eNBs. Such solutions can be considered as free because they leverage available infrastructure information. While WiFi passive sensors can be as cheap as 50$, current GSM solutions are in the range of 20 times more expensive than WiFi ones. Moreover, SDR-based solutions require a dedicated desktop machine connected to each GSM-sensor for signal processing and filtering. Hence, WiFi passive solutions are expected to be more cost efficient. Nevertheless, GSM systems provide more valuable information to third party service providers, such as country of origin of target devices.

8.3 Application Scenarios

Though passive localization systems support independent and hidden operations, which is attracted by many service providers, they are not perfect candidates for accurate localization performance. Systems that rely on signal overhearing have limited information due to privacy of users and encryption. In most realistic scenarios, the system waits for users to be active to localize them. Hence, devices that are powered but idle are not localizable in the passive approach. In a special case, GSM-based passive localization systems have to operate typically with a low number of messages. This is because the number of unencrypted uplink messages with a MD identity is limited. Such messages can only be obtained while generating a call, receiving a call, or changing the location of the MD. However, in case of WiFi, every transmitted packet in uplink has the identity (MAC address) of the target device. Hence, for passive tracking applications, WiFi-based systems are more close to the real-time operation than GSM-based ones. For active systems, such LTE network-based positioning and tracking, every transmitted uplink packet is tagged with the transmitter identity (C-RNTI). Hence, positioning and tracking applications make use of every transmitted packet.

Chapter 9

Conclusions and Outlook

The growing numbers of location-based services are strongly tied to the increased number of radio devices. Localization services in indoor environments can not only produce commercial benefit, such as localizing consumers in a shopping mall, but also improve the life quality for people, such as in old people's homes, and save people's lives in emergency circumstances. An efficient, accurate, and reliable indoor localization system requires a proper understanding of indoor radio characteristics and their influence on radio signals.

In Chapter 3, we conducted several experiments to understand parameters that affect indoor radio metrics and the implications for indoor localization systems. Our main conclusions are: (i) signal strength varies less between sensors of the same type than between MDs from different manufacturers; (ii) multipath seems to have dominant effect on signal strength; (iii) radio signals experience distinct propagation conditions in various directions; and (iv) the choice of evaluation metric depends on the time granularity of localization, i.e., mean values are convenient for real-time localization. For WiFi-based systems, data acquisition at the network-side shows different localization performance than it at the user-side. This is because wireless channels are not reciprocal even in indoor LOS conditions. The constant path loss exponent for indoor LOS scenarios is just an approximation that might degrade the overall localization performance.

In Chapter 4, we studied GSM-based localization systems. To overcome the problem of signal capturing, we studied the problem of a passive uplink GSM receiver from a theoretical and realization perspectives. We proposed a new synchronization implementation for GSM symbols and busts, which relies only on features of uplink frames transmitted by GSM MDs. The proposed solution can synchronize to multiple MDs simultaneously by only overhearing their uplink traffic. The proposed passive receiver was implemented using SDR Platform. Its performance evaluation showed an average of 98.8% success rate of downlink message recovery and close to 70% of uplink messages. We also investigated the impact of received signal strength on the receiver's performance. The results indicate a

RSSI threshold, below which successful message decoding is not possible. To overcome the problem of capturing multiple GSM devices on different operating frequencies, we proposed a novel architecture for a wideband GSM receiver. To split the captured band efficiently, we used the polyphase filterbank channelizer. Our baseline implementation is a CPU-based GR implementation. We did several AVX2 optimizations to improve the performance by up to 30 % using recent CPU instructions, namely fused multiply-add operations, compared to the AVX implementation. Moreover, we implemented a GPU-based channelizer, which performs 3.2-fold faster with respect to the PFB processing time when compared to the original GR implementation on CPU only. Now the signal is distributed and ready to be processed. Then, we define a set of challenges facing the wideband passive GSM receiver from theoretical and real implementation perspectives. Given that these constraints are natural in an uncontrolled environment, a correct operation of the proposed wideband receiver should not violate them. The difference in power between the strongest channel and the weakest channel should not be larger than 50 dBm. Otherwise, channels with high received power at the receiver side will contaminate channels with low received power. If the number of channelized streams are larger that what a machine can process, a buffer overflow will occur inside the UHD and the wideband signal will be dropped. Hence, we implemented a distributed system over an IP network to servers. The passive wideband receiver operates in a controlled environment.

To complete our localization system, we proposed several proximity-based localization algorithms, namely CDRSS and WCC. The first algorithm builds upon the centroid concept while the second algorithm is based on the circumcenter concept. Both algorithms explore information on received signal strength and particularly the difference in readings from the sensors to derive a location estimate. The second algorithm allowed us to derive location estimates of targets outside the geometry of the deployment. To validate the effectiveness of our proposed algorithms, a set of measurements were conducted in a real indoor environment and an actual GSM MD as a target. The performance results showed that with zero network configurations, both algorithms outperformed the linear weighted centroid algorithm. Moreover, we observed the impact on localization accuracy by dynamic indoor environments. To account for the complicated radio propagation and its consequences on localization accuracy in dense indoor environments, we proposed to extend outlier mitigation algorithms with a novel filtering algorithm. The approach aims to increase the quality of the radio parameters before they are fed into the localization algorithm. The experimental results indicate that the proposed algorithms can significantly decrease the absolute localization error mean and deviation using the minimum number of anchor nodes, i.e., three ANs for a 2-dimensional space.

To further improve the localization performance of a passive system, we studied in Chapter 5 our proposed hybrid-network solution using WiFi and GSM radio signals. Our proposed solution relies on online propagandistic approach to weight contributing signals. The weighting process was made before and after the localization algorithm. Due to the deployment flexibility of our hybrid SNs, we have the choice for colocated and distributed antenna setups.

To reduce the response time of an accurate system that operates passively, we focused our research on active localization systems. Namely, network-based LTE localization. Network-based localization requires the exchange of information in real-time between different eNBs. To solve this problem, a software-based implementation of eNBs running on a virtualized environment (cloud) is proposed. For large deployment density and ease of update, it is important to run our LTE sensors on commodity hardware following the SDR approach. In Chapter 6, we evaluated the minimum CPU requirements to operate a LTE eNB in a cloud environment. Using sufficient processing resources on machines operating our eNBs, we moved forward to perform localization. In Chapter 7, we proposed RTPoT, a network-based localization algorithm for LTE. RTPoT contains a novel approach to improve the quality of radio measurements and to estimate and model radio transmission channels blindly while being transparent towards end-users. By removing the environment dependency, RTPoT supports consistent and ubiquitous positioning services over the large-scale deployment of cellular networks. RTPoT is applied to LTE/LTE-A as the best enabler to tighten all location services together.

Given the limitations faced by a passive localization system (especially cellular-based), in terms of encryption and low packet rate, our future work will focus only on active localization systems. With the high adoption rate of LTE devices, such as smartphones and tablets, our future research will be focused on the following domains:

- Real-time network based positioning and tracking using cloud-based implementation of LTE eNB. Exchanging information between different eNBs can only be possible through the X2-interface. Given the fact that current implementation of OAI does not support X2-interface or even handover, our near future work is focused to implement the X2-like interface in a cloud environment to allow several mobile-edge-computing services, such as CoMP and localization.

- Given the fact that LTE radio signals are available indoor and outdoor, our future work will include also the validation of our proposed RTPoT in outdoor environments and a study on the impact of lower density of eNBs.

- Taking into consideration the Multi-Input Multi-Output (MIMO) antenna in LTE (supported by the ExpressMIMO2 board) and massive-MIMO potential in 5G networks, angle-of-arrival (AoA) methods become more and more as a native supporter for eNBs. Our future work will include hybrid-algorithm techniques for localization and tracking using a single eNB. Our proposal is based on accessing additional fine-grained radio parameters inside eNB PHY, such as timing advance, wideband-/subband RSSI, and reporting parameters for the radio resource control layer, such as RSRP and RSRQ. With the available TA resolution, a range of 72m around the defined AoA can be estimated. RSSI and RSRP can also be combined to enhance further the estimated location.

Author main contributions

- **I. Alyafawi**, N. Nikaein, T. Braun. RTPoT: Real-Time Positioning and Tracking for Mobile Edge Computing. To be submitted, 2015.
- **I. Alyafawi**, S. Kiener, T. Braun. Hybrid Indoor Localization Using Multiple Radio Interfaces. To be submitted, 2015:
  - Designed and conducted experiments, analyzed data, and wrote the paper.
- **I. Alyafawi**, A. Durand, T. Braun. High-Performance Wideband SDR Channelizers. To be submitted, 2015:
  - Designed experiments, integrated tools with GSM receiver, established and updated the paper.
- **I. Alyafawi**, T. Braun, D. Dimitrova. Robust Indoor Localization System using Narrowband Radio Signals. PIMRC, 2015.
- **I. Alyafawi**, N. Nikaein, T. Braun. Towards Real-Time Network-Based Positioning in LTE, KuVS Talk on Localization, 2015.
- **I. Alyafawi**, E. Schiller, T. Braun, D. Dimitrova, A. Gomes, N. Nikaein. Critical Issues of Centralized and Cloudified LTE-FDD Radio Access Networks, IEEE International Conference on Communications (ICC), 2015:
  - Conducted experiments, analyzed data, and wrote the paper.
- **I. Alyafawi**, D. Dimitrova, T. Braun. SDR-based Passive Indoor Localization System for GSM. ACM SIGCOMM Software Radio Implementation Forum (SRIF), 2014
- **I. Alyafawi**, D. Dimitrova, T. Braun. Real-Time Passive Capturing of the GSM Radio. IEEE International Conference on Communications (ICC), 2014.
- D. Dimitrova, **I. Alyafawi**, T. Braun. Experimental comparison of Bluetooth and WiFi signal propagation for indoor localization. The 10th International Conference on Wired/Wireless Internet Communications, 2012.

  - Conducted experiments and analyzed data.

- **I. Alyafawi**. Towards Self-Learning Radio-Based Localization Systems. IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM PhD Forum), 2012.

List of Acronyms

ADC Analog to Digital Converter

AF Attenuation Factor

AN Anchor Node

AOA Angle of Arrival

AP Access Point

AVX Advanced Vector Extensions

AWC Adaptive Weighted Centroid

AWGN Additive White Gaussian Noise

BBU Base-Band Unit

BCCH Broadcast Control Channel

BSIC Base Station Identity Code

BTS Base Transceiver Station

CDF Cumulative Distribution Function

CDMA Code Division Multiple Access

CDRSS Combined Differential RSSI

CQI Channel Quality Indicator

CoMP Coordinated Multipoint

DAC Digital to Analog Converter

DCI Downlink Control Information

DDC Digital Down Converter

DFL Device-Free Localization

DSP Digital Signal Processor

eNB evolved Node B

EPC Evolved Packet Core

FB Frequency correction Burst

FCCH Frequency Correction Channel

FDD Frequency Division Duplex

FDD Frequency Division Duplex

FDMA Frequency-Division Multiple Access

FFT Fast Fourier Transform

FIR Finite Impulse Response

FMA Fused Multiply-Add

FPGA Field-Programmable Gate Array

FPU Floating-Point Unit

FS Free Space

GbE Gigabit Ethernet

GPP General-Purpose Processors

GPS Global Positioning System

GPU Graphic Processing Unit

GR GNURadio

GSM Global System for Mobile Communications

GUI Graphical User Interface

HARQ Hybrid Automatic Repeat Request

HSS Home Subscriber Server

IMSI International Mobile Subscriber Identity

ISM Industrial, Scientific and Medical

KVM Kernel-based Virtual Machine

LNA Low Noise

LOS Line of Sight

LPF Low Pass Filter

LPF Low Pass Filter

LTE Long Term Evolution

LWC Linear Weighted Centroid

LXC Linux Containers

MAC Medium Access Control layer

MCS Modulation and Coding Scheme

MD Mobile Device

MiTM Man-in-the-Middle

MLE Maximum-Likelihood Estimator

MME Mobility Management Entity

MWF Multi-Wall-and-Floor

NB Normal Burst

NLOS Non Line of Sight

OAI OpenAirInterface

OFDM Orthogonal Frequency Division Multiplex

OFDMA Orthogonal Frequency-Division Multiple Access

PA Power Amplifier

PDCCH Physical Downlink Control Channel

PDF Probability Density Function

PDSCH Physical Downlink Shared Channel

PFB Polyphase Filterbank

PGW Packet Data Network (PDN) Gateway

PHY Physical layer

PL Path Loss

PLE Path Loss Exponent

PMF Probability Mass Function

PRB Physical Resource Block

PUCCH Physical Uplink Control Channel

PUSCH Physical Uplink Shared Channel

QoS Quality of Service

Q-Q quantile-quantile

RAN Radio Access Network

RF Radio Frequency

RFID Radio Frequency Identification)

RFN Running Frame Number

RI Radio Interface

RR Response Rate

RRC Radio Resource Control layer

RRH Remote Radio Head

RSRP Reference Signal Received Power

RSRQ Reference Signal Received Quality

RSSI Received Signal Strength Indicator

RTPoT Real-Time Positioning and Tracking

SB Synchronization Burst

SC-FDMA Single-Carrier Frequency-Division Multiple Access

SCH Synchronization Channel

SDR Software Defined Radio

SGW Serving Gateway

SIM Subscriber Identity Module

SIMD Single Instruction Multiple Data

SN Sensor Node

SNIR Signal-to-Noise-and-Interference Ratio

SNR Signal-to-Noise Ratio

SQNR Signal-to-Quantized-Noise Ratio

SS Single-Slope

TA Timing Advance

TDMA Time Division Multiple Access

TDOA Time Difference of Arrival

TMSI Temporary Mobile Subscriber Identity

TOA Time of Arrival

TS Time Slot

TSC Training Sequence Code

UE User Equipment

UHD USRP Hardware Driver

USRP Universal Software Radio Peripheral

UTDoA Uplink Time Difference of Arrival

UWB Ultra Wide Band

VOLK Vector-Optimized Library of Kernels

VPN Virtual Private Network

WARP Wireless Open Access Research Platform

WC Weighted Centroid

WCC Weighted Circumcenter

WLAN Wireless Local Area Network

Bibliography

[1] "Amari LTE 100, Software LTE base station on PC," *[Online]. Available: http://www.amarisoft.com/.*

[2] "ZHAW Openstack Testbed:," *[Online]. Available: http://blog.zhaw.ch/icclab/.*

[3] "Intel c++ compiler intrinsics reference," *Intel Corperation*, 2006.

[4] E. Abril, "Hybrid rss-rtt localization scheme for indoor wireless networks," 2010.

[5] ActiveMQ, "[online]: http://activemq.apache.org/."

[6] V. Adhinarayanan and W. C. Feng, "Wideband channelization for software-defined radio via mobile graphics processors," *International Conference on Parallel and Distributed Systems (ICPADS)*, 2013.

[7] V. Adhinarayanan, T. Koehn, K. Kepa, W. chun Feng, and P. Athanas, "On the performance and energy efficiency of fpgas and gpus for polyphase channelization," *International Conference on ReConFigurable Computing and FPGAs*, 2014.

[8] I. K. Adusei, K. Kyamakya, and K. Jobmann, "Mobile positioning technologies in cellular networks: an evaluation of their performance metrics," *MILCOM*, vol. 2, pp. 1239–1244, 2002.

[9] S. Ahmadi, *LTE-Advanced: A Practical Systems Approach to Understanding 3GPP LTE Releases 10 and 11 Radio Access Technologies*. Academic Press, 2013.

[10] I. Ahmed, S. Orfali, T. Khattab, and A. Mohamed, "Characterization of the indoor-outdoor radio propagation channel at 2.4 ghz," in *GCC Conference and Exhibition (GCC), 2011 IEEE*, feb. 2011, pp. 605 –608.

[11] I. Ahriz, Y. Oussar, B. Denby, and G. Dreyfus, "Full-band gsm fingerprints for indoor localization using a machine learning approach," *International Journal of Navigation and Observation*, 2010.

[12] S. Ahson and M. Ilyas, “Location-based services handbook: Applications, technologies, and security,” *Taylor and Francis (CRC Press)*, 2010.

[13] Airprobe, “[online]: https://svn.berlin.ccc.de/projects/airprobe/.”

[14] R. Akl, D. Tummala, and X. Li, “Indoor propagation modeling at 2.4 ghz for ieee 802.11 networks,” in *Sixth IASTED International Multi-Conference on Wireless and Optical Communications*. IASTED/ACTA Press, 2006.

[15] R. Akl and D. Tummala, “Indoor propagation modeling at 2.4 ghz for ieee 802.11 networks,” *Wireless Networks and emerging technologies*, 2006.

[16] N. Alsindi, “Indoor cooperative localization for ultra wideband wireless sensor networks,” Ph.D. dissertation, WORCESTER POLYTECHNIC INSTITUTE, 2008.

[17] I. Alyafawi, T. Braun, and D. C. Dimitrova, “Robust indoor localization system using narrowband radio signals,” *PMIRC (submitted)*, 2015.

[18] I. Alyafawi, D. Dimitrova, and T. Braun, “Sdr-based passive indoor localization system for gsm,” *ACM SIGCOMM Software Radio Implementation Forum (SRIF)*, 2014.

[19] I. Alyafawi, D. C. Dimitrova, and T. Braun, “Real-time passive capturing of the gsm radio,” *ICC*, 2014.

[20] I. Alyafawi, A. Durand, and T. Braun, “High-performance wideband sdr channelizers,” *Mobicom Workshop: Mobiarch (submitted)*, 2015.

[21] I. Alyafawi, S. Kiener, and T. Braun, “Hybrid indoor localization using multiple radio interfaces,” *To be submitted*, 2015.

[22] I. Alyafawi, N. Nikaein, and T. Braun., “Rtpot: Real-time positioning and tracking for mobile edge computing.” *To be submitted*, 2015.

[23] I. Alyafawi, N. Nikaein, and T. Braun, “Towards real-time network-based positioning in lte,” *KuVS Talk on Localization*, 2015.

[24] I. Alyafawi, E. Schiller, T. Braun, D. Dimitrova, A. Gomes, and N. Nikaein, “Critical issues of centralized and cloudified lte-fdd radio access networks,” *ICC*, 2015.

[25] Amazon, “[online]: http://aws.amazon.com/ec2/.”

[26] K. Amiri, Y. Sun, P. Murphy, C. Hunter, J. Cavallaro, and A. Sabharwal, “Warp, a unified wireless network testbed for education and research,” *IEEE International Conference Microelectronic Systems Education*, 2007.

[27] K. Arya, P. Gupta, P.Kalra, and P. Mitra, “Image registration using robust m-estimators,” *Pattern Recognition Letters*, pp. 1957– 1968, 2007.

[28] Asterisk, "[online]: http://www.asterisk.org/."

[29] M. Awan, P. Koch, C. Dick, and F. Harris, "Fpga implementation analysis of polyphase channelizer performing sample rate change required for both matched filtering and channel frequency spacing," *Asilomar Conference on Signals, Systems, and Computers*, 2010.

[30] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," *In Proc. IEEE Infocom*, pp. 775 – 784, 2000.

[31] C. Beder and M. Klepal, "Fingerprinting based localisation revisited," *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[32] R. Behnke and D. Timmermann, "Awcl: Adaptive weighted centroid localization as an efficient improvement of coarse grained localization," *Proceedings of the 5th Workshop on Positioning, Navigation and Communication*, 2008.

[33] I. Ben-Gal, *Outlier Detection.* Kluwer Academic Publishers, 2005.

[34] G. Berardinelli, L. R. de Temino, S. Frattasi, M. Rahman, and P. Mogensen, "Ofdma vs. sc-fdma: Performance comparison in local area imt-a scenarios," *IEEE Wireless Communications*, 2008.

[35] C. C. Bissell and D. A. Chapman, *Digital signal transmission.* Cambridge University Press, 1992.

[36] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted centroid localization in zigbee-based sensor networks," *Intelligent Signal Processing 2007, IEEE International Symposium*, pp. 1 – 6, October, 2007.

[37] M. Bouet and A. L. Dos Santos, "Rfid tags: Positioning principles and localization techniques," in *Wireless Days, 2008. WD'08. 1st IFIP.* IEEE, 2008, pp. 1–5.

[38] CDS, "[online]: http://cds.unibe.ch/research/software.html."

[39] S. Chan and G. Sohn, "Indoor localization using wi-fi based fingerprinting and trilateration techiques for lbs applications," in *Proceedings of the 7th International Conference on 3D Geoinformation*, 2012.

[40] Channelizer, "[online]: http://cds.unibe.ch/research/software.html."

[41] S. Chen, J. Liu, and L. Xue, "A robust non-line-of-sight error mitigation method in mobile position location," *Advances in Neural Networks – ISNN*, pp. 162–170, 2009.

[42] Y.-C. Chen and J.-C. Juang, "Outlier-detection-based indoor localization system for wireless sensor networks," *International Journal of Navigation and Observation*, 2012.

[43] J. Cherukuri, "Comparative study of stochastic indoor propagation models," The University of North Carolina at Charlotte, Tech. Rep., 2004.

[44] D. J. Cichon and T. Kurner, *Propagation Prediction Models*, 1995, ch. 4, pp. 115–208.

[45] CISCO, "[online]: www.cisco.com."

[46] M. Ciurana, F. Barceló-Arroyo, and S. Cugno, "A robust to multi-path ranging technique over IEEE 802.11 networks," *Wireless Networks*, vol. 16, pp. 943–953, 2010.

[47] L. Cong and W. Zhuang, "Nonline-of-sight error mitigation in mobile location," *IEEE Transactions on Wireless Communications*, p. 560–573, 2005.

[48] X. Costa-Pérez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio Access Network Virtualization for Future Mobile Carrier Networks," *IEEE Communications Magazine*, 2013.

[49] K. Curran, E. Furey, T. Lunney, J. Santos, D. Woods, and A. Caughey, "An evaluation of indoor location determination technologies," *Journal of Location Based Services*, 2011.

[50] E. Dahlman, S. Parkvall, and J. Sköld, *4G LTE/LTE-Advanced for Mobile Broadband*. Academic Press, UK, 2011.

[51] F. Darbari, R. Stewart, I. MaGregor, G. Whyte, and I. Thayne, "Channel estimation for short range wireless sensor network," *EURASIP*, 2005.

[52] J. Daunizeau, K. Friston, and S. Kiebel, "Variational bayesian identification and prediction of stochastic nonlinear dynamic causal models," *Physica D: Nonlinear Phenomena*, 2009.

[53] G. de la Roche, A. Alayon-Glazunov, and B. Allen, *LTE-Advanced and Next Generation Wireless Networks: Channel Modelling and Propagation*. Wiley, 2012.

[54] G. Deak, K. Curran, and J. Condell, "A survey of active and passive indoor localization systems," *Computer Communication*, pp. 1939 – 1954, 2012.

[55] DecaWave, "[online]: www.decawave.com."

[56] J. A. Deddens and M. R. Petersen, "Approaches for estimating prevalence ratios," *Occupational and environmental medicine*, pp. 501 – 506, 2008.

[57] C. del Mundo, V. Adhinarayanan, and W. C. Feng, "Accelerating fast fourier transform for wideband channelization," *ICC*, 2013.

[58] DFRC, "[online]: http://www.dfrc.ch/."

[59] D. C. Dimitrova, U. Bürgi, G. M. Dias, T. Braun, and T. Staub, "Inquiry-based bluetooth parameters for indoor localisation - an experimental study," *ERCIM*, vol. 5, 2012.

[60] D. Dimitrova, I. Alyafawi, and T. Braun, "Experimental comparison of bluetooth and wifi signal propagation for indoor localisation," *Proceeding of 10th International Conference on Wired/Wireless Internet Communications (WWIC)*, 2012.

[61] L. Doherty, K. Pister, and L. Ghaoui, "Convex position estimation in wireless sensor networks," *Proceedings of IEEE INFOCOM*, April, 2001.

[62] N. R. Draper and H. Smith, *Applied Regression Analysis*. Wiley, 1998.

[63] A. Durand, "[online]: http://cds.unibe.ch/research/pub_files/internal/du15.pdf," 2015.

[64] J. Eberspaecher, H.-J. Vogel, C. Bettstetter, and C. Hartmann, *GSM – Architecture, Protocols and Services*, 3rd ed. WILEY, 2009.

[65] V. Erceg, L. Greenstein, S. Tjandra, S. Parkoff, A. Gupta, B. Kulic, A. Julius, and R. Bianchi, "An empirically based path loss model for wireless channels in suburban environments," *IEEE Journal on Selected Areas in Communications*, 1999.

[66] ETSI, *Digital cellular telecommunications system (Phase 2+); Channel coding (GSM 05.03)*, 3gpp Std.

[67] ETSI, *Digital cellular telecommunications system (Phase 2+); Data Link (DL) layer . General aspects (GSM 04.05 version 8.0.1 Release 1999)*, ETSI Std.

[68] ETSI, *Recommendation GSM 05.10 : "Radio Sub-system Synchronization"*, ETSI Std.

[69] ETSI, "Digital cellular telecommunications system (phase 2+); mobile radio interface layer 3 specification (gsm 04.08)," 1995.

[70] ETSI, *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (3GPP TS 36.213 version 12.4.0 Release 12)*, ETSI Std., 2015.

[71] Ettus, "[online]: http://www.ettus.com/," *Ettus Research, A National Instruments Company*.

[72] Eurecom, "Graduate School and Research Center in Communication Systems," *[Online]. Available: http://www.eurecom.fr/en.*

[73] EURECOM, "Open air interface," http://www.openairinterface.org/, 2014.

[74] FFTW, "[online]: www.fftw.org."

[75] J. Figueiras and S. Frattasi, *Mobile Positioning and Tracking: From Conventional to Cooperative Techniques*. wiley, 2010.

[76] P. Filzmoser, R. Maronna, and M.Werner, "Outlier identification in high dimensions," *Computational Statistics and Data Analysis*, 2008.

[77] A. Gangal, T. Kayikcioglu, M. Sezer, and H. Kaya, "Estimation of the parameters of multipath channels," *Microsoft Research Gate*, 2001.

[78] P. Gemperline, *Practical Guide To Chemometrics, Second Edition*. CRC Press, 2006.

[79] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation Techniques: Principles and Applications*. Springer, 2013.

[80] S. Gezici, T. Zhi, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70 – 84, 2005.

[81] GNURadio, "[online]: https://gnuradio.org."

[82] A. Goswami, L. E. Ortiz, and S. R. Das, "Wigem : A learning-based approach for indoor localization," *CoNEXT*, 2011.

[83] GRAS, "[online]: https://github.com/guruofquality/gras/wiki."

[84] Y. Gwon and et al., "Robust indoor location estimation of stationary and mobile users," 2004.

[85] B. Haberland, F. Derakhshan, H. Grob-Lipski, R. Klotsche, W. Rehm, P. Schefczik, and M. Soellner, "Radio Base Stations in the Cloud," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 129–152, 2013. [Online]. Available: http://dx.doi.org/10.1002/bltj.21596

[86] A. Haeberlen, E. Flannery, A. Ladd, A. Rudys, D. Wallach, and L. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proc. of 10th annual international conference on Mobile computing and networking*, ser. MobiCom '04. ACM, 2004, pp. 70–84.

[87] F. J. Harris, *Multirate Signal Processing for Communication Systems*. Prentice Hall PTR, 2004.

[88] H. Hashemi, “The indoor radio propagation channel,” *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943 –968, 1993.

[89] R. E. Hattachi, J. Erfanian, and B. Daly, *NGMN 5G Initiative White Paper*. NGMN, 2015.

[90] P. Havinga *et al.*, “On the calibration and performance of rss-based localization methods,” in *Internet of Things (IOT), 2010*. IEEE, 2010, pp. 1–8.

[91] S. Hay and R. Harle, “Bluetooth tracking without discoverability,” in *Proc. of 4th International Symposium on Location and Context Awareness*, ser. LoCA ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 120–137.

[92] T. He, C. Huang, B. Blum, J. S. T., and Abdelzaher, “Range-free localization schemes for large scale sensor networks,” *MobiCom ‘03. Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, 2003.

[93] P. Hintjens, *ZeroMQ Messaging for Many Applications*. O’Reilly Media, 2013.

[94] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial Intelligence Review*, pp. 85 – 126, 2004.

[95] H. Holma and A. Toskala, Eds., *LTE for UMTS - OFDMA and SC-FDMA Based Radio Access*. John Wiley & Sons, Inc., 2009.

[96] HP, “Wi-fi and bluetooth interference issues,” HP, Tech. Rep., 2002.

[97] https://www.bluetooth.org/Technical/Specifications/adopted.htm.

[98] P. Huber and E. Ronchetti, *Robust Statistics*. John Wiley & Sons, Inc., 2009.

[99] M. Hubert and M. Debruyne, “Breakdown value,” *Wiley Interdisciplinary Reviews: Computational Statistics*, p. 296–302, 2009.

[100] M. Hubert and M. Debruyne, “Minimum covariance determinant,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, pp. 36–43, 2010.

[101] iJOIN, “iJOIN: an FP7 STREP project co-funded by the European Commission under the ICT theme,” *[Online]. Available: http://www.ict-ijoin.eu/.*

[102] C. M. R. Institute, “C-RAN White Paper: The Road Towards Green RAN,” *[Online]. Available: http://labs.chinamobile.com/cran*, 2013.

[103] T. Jain and T. Agrawal, “The haswell microarchitecture - 4th generation processor,” *IJCSIT*, 2013.

[104] P. Jensfelt, "Approaches to mobile robot localisation in indoor environments," Ph.D. dissertation, Kungul Tekniska Högskolan, 2001.

[105] D. Jiang, Q. Wang, J. Liu, G. Liu, and C. Cui, "Uplink coordinated multipoint reception for lte-advanced systems," *Wireless Communications, Networking and Mobile Computing*, 2009.

[106] C. B. C. H. Joerg Eberspaecher, Hans-Joerg Voegel, *GSM - Architecture, Protocols and Services*. Wiley, 2009.

[107] M. H. Kabir and R. Kohno, "A hybrid toa-fingerprinting based localization of mobile nodes using uwb signaling for non line-of-sight conditions," *Sensors*, vol. 12, no. 8, pp. 11 187–11 204, 2012.

[108] S. Kay, "Fundamentals of statistical signal processing, volume i: Estimation theory," *Prentice Hall PTR, New Jersy*, 1993.

[109] S. C. Kim and S. S. Bhattacharyya, "Implementation of a high-throughput low-latency polyphase channelizer on gpus," *EURASIP Journal on Advances in Signal Processing*, 2014.

[110] M. B. Kjaergaard, "A taxonomy for radio location fingerprinting," *Lecture Notes in Computer Science, 4718*, pp. 139 – 156, 2007.

[111] A. Kostrzew, "Development of a man in the middle attack on the gsm um-interface," *Technische Universität Berlin*, 2011.

[112] A. Kotanen, M. Hannikainen, H. Leppakoski, and T. Hamalainen, "Experiments on local positioning with bluetooth," in *International Conference on Information Technology: Coding and Computing [Computers and Communications]*, 2003, pp. 297 – 303.

[113] N. Kothari, B. Kannan, and M. B. Dias, "Robust indoor localization on a commercial smart phone," *Procedia Computer Science*, vol. 10, p. 1114–1120, 2012.

[114] A. Kumar, V. Kumar, and V. Kapoor, "Range free localization schemes for wireless sensor networks," *Recent Researches in Software Engineering, Parallel and Distributed Systems*, 2011.

[115] J. Kwon, B. Dundar, and P. Varaiya, "Hybrid algorithm for indoor positioning using wireless lan," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 7. IEEE, 2004, pp. 4625–4629.

[116] Y. H. Kwon, S. G. Choi, and J. K. Choi, "Movement detection mechanism using differential rssi in mobile mpls network," *8th International Conference Advanced Communication Technology*, pp. 594 – 598, 2006.

[117] M. Laaraiedh, L. Yu, S. Avrillon, and B. Uguen, "Comparison of hybrid localization schemes using rssi, toa, and tdoa," in *Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless), 11th European*. VDE, 2011, pp. 1–5.

[118] R. Larsen and M. Marx, *An Introduction to Mathematical Statistics and Its Applications*. Prentice Hall, 2011.

[119] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in lte-advanced: Deployment scenarios and operational challenges," *LTE-Advanced and 4G Wireless Communications*, vol. 50, 2, pp. 148 – 155, 2012.

[120] X. Lin, "Enhanced accuracy gps navigation using the interacting multiple model estimator," *Aerospace Conference*, vol. 4, pp. 1911–1923, 2001.

[121] H. Linde, "On aspects of indoor localization," Ph.D. dissertation, Universit¨at Dortmund, 2006.

[122] I. Lita, I. B. Cioc, and D. A. Visan, "A new approach of automobile localization system using gps and gsm/gprs transmission," *Electronics Technology*, pp. 115–119, 2006.

[123] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions On Ssystem, Man, and Cybernetics*, vol. 37(6), 2007.

[124] C. Lomont, "Introduction to intel advanced vector extensions," *Intel Corperation*, 2011.

[125] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, S. Sen, and V. Handziski, "Microsoft indoor localization competition: Experiences and lessons learned," *ACM SIGMOBILE*, 2014.

[126] V. K. Madisetti, *The Digital Signal Processing Handbook, Second Edition*. CRC Press, 2009.

[127] A. Mahtab Hossain, H. Nguyen Van, Y. Jin, and W. Soh, "Indoor localization using multiple wireless technologies," in *Proc. of Mobile Adhoc and Sensor Systems, MASS 2007.*, 2007, pp. 1–8.

[128] G. Mao, B. Anderson, and B. Fidan, "Online calibration of path loss exponent in wireless sensor networks," *GLOBECOM*, 2006.

[129] V. Marojevic, X. Reves, and A. Gelonch, "Computing resource management for sdr platforms," *PIMRC*, 2005.

[130] I. Martin-Escalona and F. Barcelo-Arroyo, "A new time-based algorithm for positioning mobile terminals in wireless networks," in *Journal on Advances in Signal Processing*. EURASIP, 2008.

[131] Matlab, "[online]: http://ch.mathworks.com/company/newsletters/articles/accelerating-matlab-algorithms-and-applications.html."

[132] S. Mazuelas, A. Bahillo, and R. M., "Robust indoor positioning provided by real-time rssi values in unmodified wlan networks," *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING*, vol. 3, pp. 821–831, 2009.

[133] MCN, "Mobile Cloud Networking project (FP7-ICT-318109)," *[Online]. Available: http://www.mobile-cloud-networking.eu.*, 2014.

[134] A. Melikov, *Cellular Networks - Positioning, Performance Analysis, Reliability*. InTech, Chapters published, 2011.

[135] B. Mondal, E. Visotsky, T. A. Thomas, X. Wang, and A. Ghosh, "Performance of downlink comp in lte under practical constraints," *PIMRC*, 2012.

[136] A. Munshi, "The opencl specification, version 1.1," *Khronos OpenCL Working Group*, 2010.

[137] J. Narzullaev, A. Narzullaev, and Y. Park, "Fast and cost-efficient signal strength prediction based wi-fi positioning," *International Conference on Indoor Positioning and Indoor Navigation*, 2011.

[138] NGMN, "Suggestions on Potential Solutions to C-RAN by NGMN Alliance," The Next Generation Mobile Networks (NGMN) Alliance, Tech. Rep., Jan. 2013. [Online]. Available: http://www.ngmn.org/uploads/media/NGMN_CRAN_Suggestions_on_Potential_Solutions_to_CRAN.pdf

[139] N. Nikaein, R. Favraud, E. Schiller, I. Alyafawi, Z. Zhao, and T. Braun, "Network store: Exploring slicing in future 5g networks," *Mobicom Workshop: Mobiarch (submitted)*, 2015.

[140] N. Nikaein, R. Knopp, F. Kaltenberger, L. Gauthier, C. Bonnet, D. Nussbaum, and R. Ghaddab, "OpenAirInterface 4G: an open LTE network in a PC," *International Conference on Mobile Computing and Networking*, 2014.

[141] OpenBTS, "[online]: http://wush.net/trac/rangepublic."

[142] Oracle, "Oracle Cloud, Enterprise-Grade Cloud Solutions: SaaS, PaaS, and IaaS," *Available Online: [https://cloud.oracle.com/home]*, 2014.

[143] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, "Accurate gsm indoor localization," *Proceeding of UBICOMP*, pp. 141 – 158, 2005.

[144] V. Pallipadi and A. Starikovskiy, "The ondemand governor: past, present and future," *Proceedings of Linux Symposium*, 2006.

[145] M. Patel, J. Joubert, J. R. Ramos, N. Sprecher, S. Abeta, and A. Neal, "Mobile-edge computing," *ETSI*, 2014.

[146] N. Patwari and P. Agrawal, "Calibration and measurement of signal strength for sensor localization," *Localization Algorithms and Strategies for Wireless Sensor Networks*, pp. 122 – 145, 2009.

[147] N. Patwari and J. Wilson, "Rf sensor networks for device-free localization: Measurements, models, and algorithms," *Proceedings of the IEEE*, 2010.

[148] A. Paul and E. A. Wan, "Rssi-based indoor localization and tracking using sigma-point kalman smoothers," *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING*, vol. 3, pp. 860–873, 2009.

[149] N. Paul-Ugochukwu, Q. Bao, O. Okelana, and A. Zhushi., "Enhancements to the radar user location and tracking system," *Microsoft Research*, vol. 2, p. 775–784, 2000.

[150] H. M. Paul Zarchan, *Fundamentals of Kalman Filtering:: A Practical Approach.* American Institute of Aeronautics and Astronautics, 2005.

[151] R. K. Pearson, *Mining Imperfect Data: Dealing with Contamination and Incomplete Records.* ProSanos Corporation, 2005.

[152] D. Pena and F. Prieto, "Multivariate outlier detection and robust covariance matrix estimation," *Technometrics*, 2001.

[153] C. Perez-Vega, J. L. Garcia, and J. M. L. Higuera, "A simple and efficient model for indoor path-loss prediction," *Measurement Science and Technology*, vol. 8, 1997.

[154] N. Pirzadaa, M. Y. Nayana, F. Subhanc, M. F. Hassana, and M. A. Khan, "Device-free localization technique for indoor detection and tracking of human body: A survey," *2nd International Conference on Innovation, Management and Technology Research*, 2014.

[155] W. Plishker, G. F. Zaki, S. S. Bhattacharyya, C. Clancy, and J. Kuykendall, "Applying graphics processor acceleration in a software defined radio prototyping environment," *International Symposium on Rapid System Prototyping*, 2011.

[156] RabbitMQ, "[online]: https://www.rabbitmq.com/."

[157] RadarGNURadio, "[online]: https://github.com/kit-cel/gr-radar."

[158] T. S. Rappaport, *Wireless Communications: Principles and Practice, 2nd edition*. Prentice Hall PTR, 2001.

[159] T. W. Rondeau, V. T. Shelburne, and V. T. O'Shea, "Designing anaylsis and synthesis filterbanks in gnu radio," *Karlsruhe Workshop on Software Radios*, 2014.

[160] R. Rose, C. Meier, S. Zorn, A. Goetz, , and R. Weigel, "A gsm-network for mobile phone localization in disaster scenarios," *Microwave Conference (GeMIC)*, pp. 1–4, 2011.

[161] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, NY, USA, 1987.

[162] SAIL, "Scalable and Adaptive Internet Solutions (SAIL). EU FP7 Official Website," *[Online]. Available:http://www.sail-project.eu.*, 2012.

[163] C. Sapumohotti, M. Y. Alias, and S. W. Tan, "Effects of multipath propagation and measurement noise in ieee 802.11g wlan beacon for indoor localization," *Progress In Electromagnetics Research Symposium Proceedings*, p. 5, 2012.

[164] R. Schooler, "Transforming networks with nfv & sdn," *Intel Architecture Group*, 2013.

[165] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice*. Wiley, 2011.

[166] A. Shareef, Y. Zhu, and M. Musavi, "Localization using neural networks in wireless sensor networks," *Mobileware*, vol. 1, pp. 1–7, 2008.

[167] M. Shoemake, "Wi-fi (ieee 802.11b) and bluetooth coexistence issues and solutions for the 2.4 ghz ism band," *Texas Instruments*, 2001.

[168] C. Suliman, C. Cruceru, and F. Moldoveanu, "Mobile robot position estimation using the kalman filter," *Scientific Bulletin of the Petru Maior University of Tirgu Mures*, vol. 6, 2009.

[169] C. M. Takenga and K. Kyamakya, "Location fingerprinting in gsm network and impact of data pre-processing," *World Mobile Congress*, 2006.

[170] P. Tarrio, A. M. Bernardos, and J. R. Casar, "Weighted least squares techniques for improved received signal strength based localization," *Sensors*, 2011.

[171] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik., "Indoor localization without infrastructure using the acoustic background spectrum," *MobiSys*, 2011.

[172] L. Telefonaktiebolaget and I. Siomina, “Ensuring positioning quality-of-service for lte positioning,” *Google Patents*, 2013.

[173] Z. B. Timea Bagosi, “Indoor localization by wifi,” *Intelligent Computer Communication and Processing (ICCP)*, pp. 449–452, 2011.

[174] TrackingSatellite, “[online]: https://gnuradio.org/redmine/projects/gnuradio /wiki/globalpositioningsystem.”

[175] R. Tsuchiyama, T. Nakamura, T. Iizuka, A. Asahara, J. Son, and S. Miki, *The OpenCL Programming Book*. Fixstars, 2010.

[176] A. S. Ugal, “Hard Real Time Linux using Xenomai on Intel Multi-Core Processors,” *Intel Corporation*, 2009.

[177] K. van der Veldt, R. van Nieuwpoort, V. A. L., and C. Jesshope, “A polyphase filter for gpus and multi-core processors,” *Workshop on High-Performance Computing for Astronomy*, 2012.

[178] VOLK, “[online]: https://gnuradio.org/redmine/projects/gnuradio/wiki/volk.”

[179] Y. Wang, B. Zhao, and Z. Jiang, “Rssi-based smooth localization for indoor environment,” *The Scientific World Journal*, 2014.

[180] X. Wanga, O. Bischoffa, R. Laura, and S. Paula, “Localization in wireless ad-hoc sensor networks using multilateration with rssi for logistic applications,” *Procedia Chemistry*, 2009.

[181] C.-D. Wann, *Kalman Filtering for NLOS Mitigation and Target Tracking in Indoor Wireless Environment*. InTech, 2010, ch. 16, pp. 309–326.

[182] X. Wei, L. Wang, and J. Wan, “A new localization technique based on network tdoa information,” in *ITS Telecommunications Proceedings, 2006 6th International Conference on*. IEEE, 2006, pp. 127–130.

[183] H. Wen, P. K. Tiwary, and T. Le-Ngoc, *Wireless Virtualization*. Springer, 2013.

[184] A. Wesselsa, X. Wangb, R. Laurb, and W. Langa, “Dynamic indoor localization using multilateration with rssi in wireless sensor networks for transport logistics,” *Procedia Engineering*, 2010.

[185] N. Wilt, *The CUDA Handbook: A Comprehensive Guide to GPU Programming*. Addison-Wesley Professional, 2012.

[186] J. Xiong and K. Jamieson., “Arraytrack: A fine-grained indoor location system,” *NSDI*, 2013.

[187] C. Yang, Y. Chang, Y. Chen, and C. Chu, "A self-adaptable indoor localization scheme for wireless sensor networks," *International Journal of Software Engineering and Knowledge Engineering*, vol. 21, p. 33–54, 2011.

[188] M. Youssef, M. Mah, and A. Agrawala, "Challenges: device-free passive localization for wireless environments," *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking. New York, NY, USA: ACM*, p. 222–229, 2007.

[189] P. Zandbergen, "Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning," *Transactions in GIS*, 2009.

[190] R. Zekavat and R. M. Buehrer, *Handbook of position location: Theory, Practice and Advances*. Wiley and IEEE press, 2012.

[191] G. Zhang, S. Krishnan, F. Chin, and C. Ko, "UWB multicell indoor localization experiment system with adaptive TDOA combination," in *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, 2008, pp. 1–5.

[192] V. Y. Zhang, A.-S. Wong, K. T. Woo, and R. W. Ouyang, "Hybrid toa/aoa-based mobile localization with and without tracking in cdma cellular networks," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[193] Z. Zhang and Z. Rao, "A weighted compensation of coordinate error localization algorithm based on rssi," *Journal of Chemical and Pharmaceutical Research*, 2014.

[194] S. Zorn, R. Rose, A. Goetz, and R. Weigel, "A novel technique for mobile phone localization for search and rescue applications," *Indoor Positioning and Indoor Navigation*, vol. 1-4, 2010.